



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

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

(11)



1-0029457

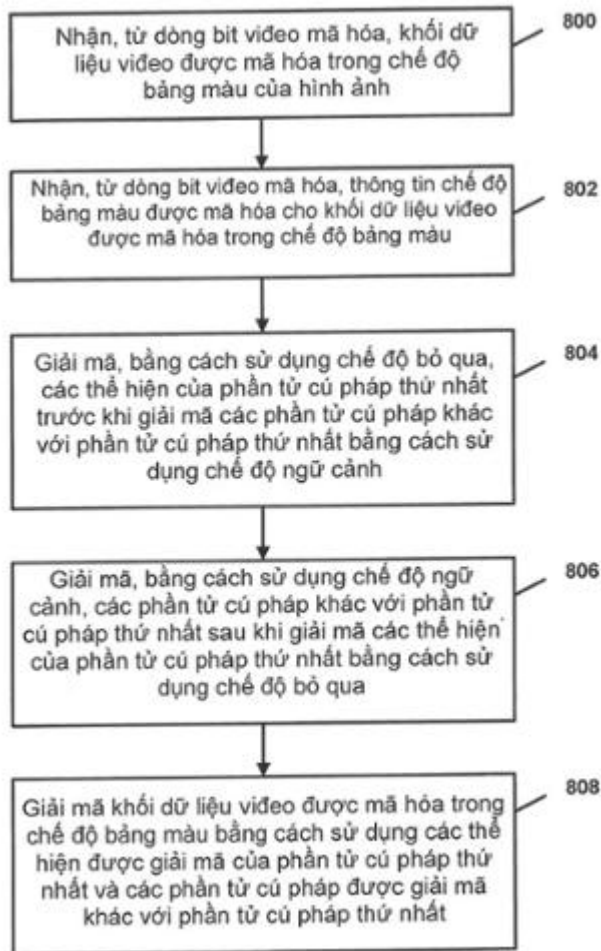
(51)⁷ H04N 19/70; H04N 19/42; H04N
19/593; H04N 19/13; H04N 19/463

(13) B

-
- (21) 1-2017-02711 (22) 29/01/2016
(86) PCT/US2016/015663 29/01/2016 (87) WO 2016/123488 A1 04/08/2016
(30) 62/110,302 30/01/2015 US; 15/009,477 28/01/2016 US
(45) 25/09/2021 402 (43) 25/10/2017 355A
(73) QUALCOMM INCORPORATED (US)
Attn: International IP Administration, 5775 Morehouse Drive, San Diego, CA 92121-1714, United States of America
(72) KARCZEWICZ, Marta (US); PU, Wei (CN); JOSHI, Rajan Laxman (US);
SEREGIN, Vadim (RU).
(74) Công ty TNHH Quốc tế D&N (D&N INTERNATIONAL CO.,LTD.)
-

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ GIẢI MÃ VÀ MÃ HÓA DỮ LIỆU VIDEO VÀ
VẬT GHI BẮT BIẾN ĐỘC ĐƯỢC BẰNG MÁY TÍNH

(57) Sáng chế đề xuất các phương pháp và thiết bị giải mã và mã hóa dữ liệu video. Theo một ví dụ, phương pháp giải mã dữ liệu video có thể bao gồm bước nhận khôi dữ liệu video được mã hóa trong chế độ bảng màu của hình ảnh. Phương pháp này có thể bao gồm bước nhận thông tin chế độ bảng màu được mã hóa cho khôi dữ liệu video được mã hóa trong chế độ bảng màu. Thông tin chế độ bảng màu được mã hóa có thể bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất. Phương pháp này có thể bao gồm bước giải mã, bằng cách sử dụng chế độ bỏ qua, các thể hiện của phần tử cú pháp thứ nhất trước khi giải mã các phần tử cú pháp khác với phần tử cú pháp thứ nhất bằng cách sử dụng chế độ ngữ cảnh. Phương pháp này có thể bao gồm bước giải mã, bằng cách sử dụng chế độ ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất sau khi giải mã các thể hiện của phần tử cú pháp thứ nhất bằng cách sử dụng chế độ bỏ qua. Ngoài ra, vật ghi bắt biến độc được bằng máy tính cũng được bộc lộ.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến việc mã hóa và giải mã nội dung, và cụ thể hơn đến việc mã hóa và giải mã nội dung theo chế độ mã hóa dựa trên bảng màu.

Tình trạng kỹ thuật của sáng chế

Các tính năng video số có thể được đưa vào áp dụng trong rất nhiều thiết bị bao gồm máy thu hình kỹ thuật số, hệ thống truyền hình trực tiếp kỹ thuật số, hệ thống truyền hình không dây, thiết bị hỗ trợ số cá nhân (personal digital assistant - PDA), máy tính xách tay hoặc máy tính để bàn, máy tính bảng, thiết bị đọc sách điện tử, máy ảnh số, thiết bị ghi số, thiết bị phát phương tiện số, thiết bị chơi trò chơi điện tử, bàn giao tiếp trò chơi điện tử, điện thoại di động hoặc điện thoại vô tuyến vệ tinh, được gọi là “máy điện thoại thông minh”, thiết bị hội nghị truyền hình, thiết bị truyền dòng dữ liệu video, và các thiết bị tương tự khác. Các thiết bị video kỹ thuật số thực hiện các kỹ thuật nén video, chẳng hạn như các kỹ thuật được mô tả trong các chuẩn được quy định bởi MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Phần 10, Mã hóa dữ liệu video nâng cao (Advanced Video Coding - AVC), Chuẩn mã hóa dữ liệu video hiệu suất cao (High Efficiency Video Coding - HEVC), và các dạng mở rộng của các chuẩn này. Các thiết bị video có thể truyền, nhận, mã hóa, giải mã, và/hoặc lưu trữ thông tin video số hiệu quả hơn bằng cách thực hiện các kỹ thuật nén video như vậy.

Các kỹ thuật nén dữ liệu video thực hiện kỹ thuật dự báo không gian (dự báo nội hình ảnh) và/hoặc dự báo thời gian (dự báo liên hình ảnh) để giảm bớt hoặc loại bỏ dữ liệu dư thừa có trong các chuỗi dữ liệu video. Đối với mã hóa dữ liệu video theo khối, lát video (tức là, khung video hoặc một phần khung video) có thể được chia thành các khối video. Các khối video trong lát mã hóa dự báo nội cấu trúc (I) của hình ảnh được mã hóa bằng cách áp dụng kỹ thuật dự báo không gian dựa vào các mẫu chuẩn ở các khối liền kề trong cùng một hình ảnh. Các khối video trong lát mã hóa dự báo liên cấu trúc (P hoặc B) của hình ảnh có thể áp dụng kỹ thuật dự báo không gian dựa vào các mẫu chuẩn trong các khối lân cận trong cùng một hình ảnh hoặc áp dụng kỹ thuật dự báo thời gian dựa vào mẫu chuẩn trong các hình ảnh chuẩn khác. Hình ảnh có

thể được gọi là khung, và hình ảnh chuẩn có thể được gọi là khung chuẩn.

Dự báo không gian hoặc thời gian tạo ra khối dự báo cho khối cần được mã hóa. Dữ liệu dư là giá trị chênh lệch điểm ảnh giữa khối gốc cần mã hóa và khối dự báo. Khối mã hóa dự báo liên cấu trúc được mã hóa theo vector chuyển động trở đến một khối mẫu chuẩn tạo nên khối dự báo, và dữ liệu dư chỉ báo giá trị chênh lệch giữa khối mã hóa và khối dự báo. Khối mã hóa nội cấu trúc được mã hóa theo chế độ mã hóa nội cấu trúc và dữ liệu dư. Để nén thêm, dữ liệu dư có thể được biến đổi từ miền điểm ảnh sang miền biến đổi, tạo ra các hệ số biến đổi dư mà sau đó các hệ thống này có thể được lượng tử hóa. Các hệ số được lượng tử hóa, ban đầu được sắp xếp thành mảng hai chiều, có thể được quét để tạo ra vector một chiều của các hệ số, và kỹ thuật mã hóa entropy có thể được áp dụng để đạt hiệu quả nén cao hơn.

Nội dung, chẳng hạn như ảnh, có thể được mã hóa và giải mã bằng cách sử dụng chế độ bảng màu. Nhìn chung, chế độ bảng màu là kỹ thuật liên quan đến việc sử dụng bảng màu để biểu diễn nội dung. Nội dung có thể được mã hóa sao cho nội dung được biểu diễn bởi giá trị ánh xạ chỉ số mà bao gồm các giá trị tương ứng với bảng màu. Giá trị ánh xạ chỉ số có thể được giải mã để tái tạo nội dung.

Bản chất kỹ thuật của sáng chế

Các kỹ thuật theo sáng chế đề cập đến việc mã hóa nội dung dựa trên bảng màu. Ví dụ, trong kỹ thuật mã hóa nội dung dựa trên bảng màu, bộ mã hóa nội dung (ví dụ, bộ mã hóa nội dung như bộ mã hóa dữ liệu video hoặc bộ giải mã dữ liệu video) có thể tạo ra “bảng màu” dưới dạng một bảng các màu sắc để biểu diễn dữ liệu video của một vùng cụ thể (ví dụ, một khối cho trước). Kỹ thuật mã hóa nội dung dựa trên bảng màu có thể, ví dụ, đặc biệt hữu dụng cho việc mã hóa các vùng dữ liệu video có số lượng màu tương đối ít. Thay vì mã hóa các giá trị điểm ảnh thực (hoặc dữ liệu dư của chúng), bộ mã hóa nội dung có thể mã hóa các chỉ số bảng màu (ví dụ, các giá trị chỉ số) của một hoặc nhiều điểm ảnh mà liên quan các điểm ảnh với các mục nhập trong bảng màu biểu diễn màu sắc của các điểm ảnh. Các kỹ thuật được mô tả theo sáng chế có thể bao gồm các kỹ thuật dùng cho các kết hợp khác nhau của một hoặc nhiều kỹ thuật báo hiệu các chế độ mã hóa dựa trên bảng màu, truyền các bảng màu, suy ra các bảng màu, suy ra giá trị của các phần tử cú pháp không được truyền, truyền các giá trị ánh xạ mã hóa dựa trên bảng màu và các phần tử cú pháp khác, dự báo các mục nhập

bảng màu, mã hóa các loạt chỉ số bảng màu, mã hóa entropy thông tin bảng màu, và nhiều kỹ thuật mã hóa bảng màu khác.

Theo một ví dụ, sáng chế đề xuất phương pháp giải mã dữ liệu video bao gồm bước nhận, từ dòng bit video mã hóa, khối dữ liệu video mã hóa trong chế độ bảng màu của hình ảnh; nhận, từ dòng bit video mã hóa, thông tin chế độ bảng màu được mã hóa cho khối dữ liệu video mã hóa trong chế độ bảng màu, trong đó thông tin chế độ bảng màu được mã hóa bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất; giải mã, bằng cách sử dụng chế độ bỏ qua, các thể hiện của phần tử cú pháp thứ nhất trước khi giải mã các phần tử cú pháp khác với phần tử cú pháp thứ nhất bằng cách sử dụng chế độ ngữ cảnh; giải mã, bằng cách sử dụng chế độ ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất sau khi giải mã các thể hiện của phần tử cú pháp thứ nhất bằng cách sử dụng chế độ bỏ qua; và giải mã khối dữ liệu video được mã hóa trong chế độ bảng màu bằng cách sử dụng các thể hiện được giải mã của phần tử cú pháp thứ nhất và các phần tử cú pháp được giải mã khác với phần tử cú pháp thứ nhất.

Theo ví dụ khác, sáng chế đề xuất thiết bị giải mã dữ liệu video bao gồm bộ nhớ được tạo cấu hình để lưu trữ dữ liệu video; và bộ giải mã dữ liệu video truyền thông với bộ nhớ được tạo cấu hình để: nhận, từ dòng bit video mã hóa, khối dữ liệu video được mã hóa trong chế độ bảng màu của hình ảnh; nhận, từ dòng bit video mã hóa, thông tin chế độ bảng màu được mã hóa cho khối dữ liệu video được mã hóa trong chế độ bảng màu, trong đó thông tin chế độ bảng màu được mã hóa bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất; giải mã, bằng cách sử dụng chế độ bỏ qua, các thể hiện của phần tử cú pháp thứ nhất trước khi giải mã các phần tử cú pháp khác với phần tử cú pháp thứ nhất bằng cách sử dụng chế độ ngữ cảnh; giải mã, bằng cách sử dụng chế độ ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất sau khi giải mã các thể hiện của phần tử cú pháp thứ nhất bằng cách sử dụng chế độ bỏ qua; và giải mã khối dữ liệu video được mã hóa trong chế độ bảng màu bằng cách sử dụng các thể hiện được giải mã của phần tử cú pháp thứ nhất và các phần tử cú pháp được giải mã khác với phần tử cú pháp thứ nhất.

Theo ví dụ khác, sáng chế đề xuất vật ghi bất biến đọc được bằng máy tính có các lệnh được lưu trữ trên đó mà, khi được thực thi, khiến cho một hoặc nhiều bộ xử lý

nhận, từ dòng bit video mã hóa, khối dữ liệu video được mã hóa trong chế độ bảng màu của hình ảnh; nhận, từ dòng bit video mã hóa, thông tin chế độ bảng màu được mã hóa cho khối dữ liệu video được mã hóa trong chế độ bảng màu, trong đó thông tin chế độ bảng màu được mã hóa bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất; giải mã, bằng cách sử dụng chế độ bỏ qua, các thể hiện của phần tử cú pháp thứ nhất trước khi giải mã các phần tử cú pháp khác với phần tử cú pháp thứ nhất bằng cách sử dụng chế độ ngữ cảnh; giải mã, bằng cách sử dụng chế độ ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất sau khi giải mã các thể hiện của phần tử cú pháp thứ nhất bằng cách sử dụng chế độ bỏ qua; và giải mã khối dữ liệu video được mã hóa trong chế độ bảng màu bằng cách sử dụng các thể hiện được giải mã của phần tử cú pháp thứ nhất và các phần tử cú pháp được giải mã khác với phần tử cú pháp thứ nhất.

Theo ví dụ khác, sáng chế đề xuất phương pháp mã hóa dữ liệu video bao gồm bước xác định rằng khối dữ liệu video cần mã hóa trong chế độ bảng màu; mã hóa khối dữ liệu video bằng cách sử dụng chế độ bảng màu trong dòng bit mã hóa, trong đó bước mã hóa khối dữ liệu video bằng cách sử dụng chế độ bảng màu bao gồm các bước: tạo ra thông tin chế độ bảng màu cho khối dữ liệu video, trong đó thông tin chế độ bảng màu bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất; mã hóa, bằng cách sử dụng chế độ bỏ qua, các thể hiện của phần tử cú pháp thứ nhất trong dòng bit mã hóa trước khi mã hóa các phần tử cú pháp khác với phần tử cú pháp thứ nhất trong dòng bit mã hóa bằng cách sử dụng chế độ ngữ cảnh; và mã hóa, bằng cách sử dụng chế độ ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất trong dòng bit mã hóa sau khi mã hóa các thể hiện của phần tử cú pháp thứ nhất bằng cách sử dụng chế độ bỏ qua trong dòng bit mã hóa.

Theo ví dụ khác, sáng chế đề xuất thiết bị mã hóa dữ liệu video, thiết bị này bao gồm bộ nhớ được tạo cấu hình để lưu trữ dữ liệu video; và bộ mã hóa dữ liệu video truyền thông với bộ nhớ, bộ mã hóa dữ liệu video được tạo cấu hình để: xác định rằng khối dữ liệu video lưu trữ trong bộ nhớ cần được mã hóa trong chế độ bảng màu; mã hóa khối dữ liệu video bằng cách sử dụng chế độ bảng màu trong dòng bit mã hóa, trong đó bộ mã hóa dữ liệu video được tạo cấu hình để mã hóa khối dữ liệu video bằng cách sử dụng chế độ bảng màu bao gồm bộ mã hóa dữ liệu video được tạo cấu hình để:

tạo ra thông tin chế độ bảng màu cho khối dữ liệu video, trong đó thông tin chế độ bảng màu bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất; mã hóa, bằng cách sử dụng chế độ bỏ qua, các thể hiện của phần tử cú pháp thứ nhất trong dòng bit mã hóa trước khi mã hóa các phần tử cú pháp khác với phần tử cú pháp thứ nhất trong dòng bit mã hóa bằng cách sử dụng chế độ ngữ cảnh; và mã hóa, bằng cách sử dụng chế độ ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất trong dòng bit mã hóa sau khi mã hóa các thể hiện của phần tử cú pháp thứ nhất bằng cách sử dụng chế độ bỏ qua trong dòng bit mã hóa.

Một hoặc nhiều phương án làm ví dụ sẽ được mô tả chi tiết dưới đây có dựa vào hình vẽ kèm theo và phần mô tả chi tiết sáng chế. Các dấu hiệu, mục đích và ưu điểm khác của sáng chế sẽ được hiểu rõ hơn khi xem phần mô tả chi tiết sáng chế và hình vẽ, cùng với các điểm yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ khối thể hiện hệ thống mã hóa dữ liệu video làm ví dụ mà có thể sử dụng các kỹ thuật được mô tả trong bản mô tả này.

Fig.2 là sơ đồ khối thể hiện bộ mã hóa dữ liệu video làm ví dụ mà có thể thực hiện các kỹ thuật được mô tả trong bản mô tả này.

Fig.3 là sơ đồ khối thể hiện bộ giải mã dữ liệu video làm ví dụ mà có thể thực hiện các kỹ thuật được mô tả trong bản mô tả này.

Fig.4 là sơ đồ khái niệm thể hiện ví dụ về việc xác định các mục nhập bảng màu để mã hóa dữ liệu video dựa trên bảng màu, theo các kỹ thuật trong bản mô tả này.

Fig.5 là sơ đồ khái niệm thể hiện ví dụ về việc xác định các chỉ số theo bảng màu cho khối điểm ảnh, theo các kỹ thuật trong bản mô tả này.

Fig.6 là sơ đồ khái niệm thể hiện ví dụ về việc xác định độ dài loạt copyabove lớn nhất, giả sử là trình tự quét mảnh, theo các kỹ thuật của sáng chế.

Fig.7 là bảng thể hiện các thay đổi đối với trình tự mã hóa của các phần tử cú pháp trong chế độ bảng màu.

Fig.8 là lưu đồ thể hiện quy trình giải mã dữ liệu video làm ví dụ theo các kỹ thuật mã hóa dữ liệu video dựa trên bảng màu của sáng chế.

Fig.9 là lưu đồ thể hiện quy trình mã hóa dữ liệu video làm ví dụ theo các kỹ thuật mã hóa dữ liệu video dựa trên bảng màu của sáng chế.

Mô tả chi tiết sáng chế

Các khía cạnh của sáng chế đề cập đến các kỹ thuật mã hóa nội dung (ví dụ, mã hóa dữ liệu video). Cụ thể, sáng chế mô tả các kỹ thuật mã hóa dữ liệu nội dung dựa trên bảng màu (ví dụ, dữ liệu video) và các kỹ thuật mã hóa số học nhị phân thích nghi theo ngữ cảnh (context-based adaptive binary arithmetic coding - CABAC) đối với thông tin mã hóa theo bảng màu. Theo các ví dụ của sáng chế, các kỹ thuật của sáng chế có thể hướng đến các quá trình dự báo hoặc mã hóa khối trong chế độ bảng màu để nâng cao hiệu suất mã hóa và/hoặc làm giảm độ phức tạp mã hóa-giải mã, như được mô tả chi tiết hơn trong phần dưới đây. Ví dụ, sáng chế mô tả các kỹ thuật liên quan đến việc nhóm chỉ số bảng màu (chẳng hạn như nhóm chỉ số bảng màu nâng cao).

Trong quy trình CABAC, ví dụ, như được mô tả trong tài liệu của D. Marpe, H. Schwarz, và T. Wiegand, “Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard”, trong tạp chí *IEEE Trans. Cir. & Sys. Video Tech.*, tập 13, số 7, tháng 7/2003, có hai chế độ: (1) chế độ bỏ qua và (2) chế độ ngữ cảnh. Trong chế độ bỏ qua, không có quá trình cập nhật ngữ cảnh. Do đó, chế độ bỏ qua có thể đạt được năng suất dữ liệu cao hơn so với chế độ dựa trên ngữ cảnh bằng cách khai thác phần cứng hoặc cấu trúc song song mức ISA. Ưu điểm này của chế độ bỏ qua trở nên lớn hơn khi số lượng bin trong chế độ bỏ qua có thể được xử lý cùng tăng lên.

Theo thiết kế mã hóa trong chế độ bảng màu hiện nay, như được mô tả trong tài liệu của R. Joshi và J. Xu, “High efficient video coding (HEVC) screen content coding: Draft 2”, JCTVC-S1005, trong phương pháp mã hóa nội dung màn hình, các phần tử cú pháp **palette_index_idc** và **palette_escape_val** được mã hóa trong chế độ bỏ qua CABAC, và được đan xen với các phần tử cú pháp khác chẳng hạn như **palette_run_msb_id_plus1** mà được mã hóa trong chế độ ngữ cảnh CABAC. Sáng chế mô tả các kỹ thuật nhóm các phần tử cú pháp mã hóa trong chế độ bỏ qua với nhau. Như được sử dụng ở đây, cụm từ “được mã hóa trong chế độ bỏ qua” và “được mã hóa trong chế độ ngữ cảnh” có thể thay thế bởi cụm từ “được mã hóa bỏ qua” và “được mã hóa ngữ cảnh”.

Như được sử dụng ở đây, các ví dụ của thuật ngữ “nội dung” có thể được thay đổi thành thuật ngữ “video”, và các ví dụ của thuật ngữ “video” có thể được thay đổi

thành thuật ngữ “nội dung”. Điều này luôn đúng bất kể các thuật ngữ “nội dung” hay “video” đang được dùng với vai trò là tính từ, danh từ hay loại từ khác. Ví dụ, khi đề cập đến “bộ mã hóa nội dung” là cũng để chỉ “bộ mã hóa dữ liệu video”, và khi đề cập đến “bộ mã hóa dữ liệu video” là cũng để chỉ “bộ mã hóa nội dung”. Tương tự, khi đề cập đến “nội dung” là cũng để chỉ “dữ liệu video”, và khi đề cập đến “dữ liệu video” là cũng để chỉ “nội dung”.

Khi được sử dụng ở đây, “nội dung” có nghĩa là chỉ loại nội dung bất kỳ. Ví dụ, “nội dung” dùng để chỉ dữ liệu video, nội dung màn hình, ảnh, nội dung đồ họa bất kỳ, nội dung hiển thị được bất kỳ, hoặc dữ liệu bất kỳ tương ứng với (ví dụ, dữ liệu video, dữ liệu nội dung màn hình, dữ liệu ảnh, dữ liệu nội dung đồ họa, dữ liệu nội dung hiển thị được, và tương tự).

Như được sử dụng ở đây, thuật ngữ “dữ liệu video” có thể chỉ nội dung màn hình, nội dung có thể di động, nhiều ảnh mà có thể được biểu diễn trong chuỗi, hoặc dữ liệu bất kỳ tương ứng với (ví dụ, dữ liệu nội dung màn hình, dữ liệu nội dung di động, dữ liệu video, dữ liệu ảnh, và dữ liệu tương tự).

Như được sử dụng ở đây, thuật ngữ “ảnh” có thể chỉ một ảnh, một hoặc nhiều ảnh, một hoặc nhiều ảnh trong số nhiều ảnh tương ứng với dữ liệu video, một hoặc nhiều ảnh trong số nhiều ảnh không tương ứng với dữ liệu video, nhiều ảnh tương ứng với dữ liệu video (ví dụ, tất cả ảnh tương ứng với dữ liệu video hoặc ít hơn tất cả ảnh tương ứng với dữ liệu video), phần phụ của một ảnh, nhiều phần phụ của một ảnh, nhiều phần phụ tương ứng với nhiều ảnh, một hoặc nhiều hình đồ họa gốc, dữ liệu ảnh, dữ liệu đồ họa, và tương tự.

Trong kỹ thuật mã hóa dữ liệu video thông thường, ảnh được giả định là có sắc màu liên tục và mượt trong không gian. Dựa vào những giả định này, nhiều công cụ đã được phát triển chẳng hạn như biến đổi theo khối, lọc, và các công cụ mã hóa khác, và các công cụ như vậy đã cho thấy hiệu suất tốt đối với các video có nội dung tự nhiên. Tuy nhiên, trong các ứng dụng như điều khiển màn hình từ xa, màn hình làm việc cộng tác và không dây, nội dung màn hình được tạo ra bởi máy tính có thể là nội dung chính cần được nén. Loại nội dung màn hình này thường có sắc màu rời rạc, các đường gấp khúc và các đường bao đối tượng có độ tương phản cao. Việc giả định về sắc màu liên tục và độ mượt có thể không còn áp dụng được nữa, và do đó, các kỹ thuật mã hóa dữ liệu video thông thường có thể không có hiệu quả khi nén nội dung

(ví dụ, nội dung màn hình).

Theo một ví dụ về mã hóa dữ liệu video dựa trên bảng màu, bộ mã hóa dữ liệu video có thể mã hóa khối dữ liệu video bằng cách xác định bảng màu cho khối đó (chẳng hạn, mã hóa bảng màu một cách rõ ràng, dự báo bảng màu hoặc kết hợp cả hai), định vị mục nhập trong bảng màu để biểu diễn (các) giá trị của một hoặc nhiều điểm ảnh, và mã hóa cả bảng màu và khối với các giá trị chỉ số biểu thị mục nhập trong bảng màu được sử dụng để biểu diễn các giá trị điểm ảnh của khối này. Theo một số ví dụ, bộ mã hóa dữ liệu video có thể báo hiệu bảng màu và/hoặc các giá trị chỉ số trong dòng bit mã hóa. Sau đó, bộ giải mã dữ liệu video có thể thu được, từ dòng bit mã hóa, bảng màu cho khối, cũng như các giá trị chỉ số cho các điểm ảnh riêng lẻ của khối này. Bộ giải mã dữ liệu video có thể liên kết các giá trị chỉ số của các điểm ảnh với các mục nhập của bảng màu để tái tạo các giá trị điểm ảnh của khối.

Ví dụ, vùng dữ liệu video cụ thể có thể được giả định có số lượng màu sắc tương đối nhỏ. Bộ mã hóa dữ liệu video (ví dụ, bộ mã hóa dữ liệu video hoặc bộ giải mã dữ liệu video) có thể mã hóa (ví dụ, mã hóa hoặc giải mã) cái gọi là “bảng màu” để biểu diễn dữ liệu video của vùng cụ thể này. Bảng màu có thể được biểu thị dưới dạng chỉ số (ví dụ, bảng) của các màu sắc hoặc các giá trị điểm ảnh biểu diễn dữ liệu video của vùng cụ thể (ví dụ, khối cho trước). Bộ mã hóa dữ liệu video có thể mã hóa chỉ số, chỉ số này liên kết một hoặc nhiều giá trị điểm ảnh với giá trị thích hợp trong bảng màu. Mỗi điểm ảnh có thể được kết hợp với một mục nhập trong bảng màu để biểu diễn màu sắc của điểm ảnh. Ví dụ, bảng màu có thể bao gồm các giá trị điểm ảnh trội nhất trong khối cho trước. Trong một số trường hợp, các giá trị điểm ảnh trội nhất này có thể bao gồm một hoặc nhiều giá trị điểm ảnh xuất hiện thường xuyên nhất trong khối. Ngoài ra, trong một số trường hợp, bộ mã hóa dữ liệu video có thể áp dụng giá trị ngưỡng để xác định xem giá trị điểm ảnh có cần được đưa vào như là một trong số các giá trị điểm ảnh trội nhất của khối hay không. Theo các khía cạnh khác nhau của kỹ thuật mã hóa dựa trên bảng màu, bộ mã hóa dữ liệu video có thể mã hóa các giá trị chỉ số biểu thị một hoặc nhiều giá trị điểm ảnh của khối hiện thời, thay vì mã hóa các giá trị điểm ảnh thực hoặc các giá trị dư của chúng cho khối dữ liệu video hiện thời. Trong ngữ cảnh mã hóa dựa trên bảng màu, các giá trị chỉ số biểu thị các mục nhập tương ứng trong bảng màu mà được dùng để biểu diễn các giá trị điểm ảnh riêng lẻ của khối hiện thời. Phần mô tả ở trên dự định cung cấp sự mô tả khái quát về kỹ thuật mã

hóa dữ liệu video dựa trên bảng màu.

Kỹ thuật mã hóa dựa trên bảng màu có thể đặc biệt thích hợp đối với mã hóa nội dung được tạo ra từ màn hình hoặc nội dung khác khi một hoặc nhiều công cụ mã hóa truyền thống không đem lại hiệu quả. Các kỹ thuật mã hóa dữ liệu video dựa trên bảng màu có thể được sử dụng cùng với một hoặc nhiều kỹ thuật mã hóa khác, chẳng hạn như các kỹ thuật mã hóa dự báo liên cấu trúc hoặc nội cấu trúc. Ví dụ, như được mô tả chi tiết hơn dưới đây, bộ mã hóa hoặc bộ giải mã, hoặc bộ mã hóa-giải mã (encoder-decoder - codec) kết hợp, có thể được tạo cấu hình để thực hiện mã hóa dự báo liên cấu trúc và nội cấu trúc, cũng như mã hóa dựa trên bảng màu.

Theo một số ví dụ, các kỹ thuật mã hóa dựa trên bảng màu có thể được tạo cấu hình để sử dụng với một hoặc nhiều chuẩn mã hóa dữ liệu video. Ví dụ, mã hóa dữ liệu video hiệu suất cao (High Efficiency Video Coding - HEVC) là chuẩn mã hóa dữ liệu video mới được phát triển bởi Nhóm hợp tác chung về mã hóa dữ liệu video (Joint Collaboration Team on Video Coding - JCT-VC) thuộc Nhóm chuyên gia mã hóa dữ liệu video (Video Coding Experts Group - VCEG) của tổ chức ITU-T và Nhóm chuyên gia mã hóa hình ảnh động (Motion Picture Experts Group - MPEG) của tổ chức ISO/IEC. Tài liệu chuẩn HEVC cuối cùng được xuất bản là “ITU-T H.265, SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services – Coding of moving video - High efficiency video coding”, Ban tiêu chuẩn hóa viễn thông thuộc Hiệp hội viễn thông quốc tế (International Telecommunication Union - ITU), 4/2013.

Để mã hóa nội dung được tạo ra trên màn hình hiệu quả hơn, JCT-VC đang phát triển phần mở rộng theo chuẩn HEVC, được gọi là chuẩn mã hóa nội dung màn hình (Screen Content Coding - SCC) HEVC. Bản dự thảo làm việc mới đây của chuẩn HEVC SCC, được gọi là “Dự thảo HEVC SCC 2” hoặc “WD2”, được mô tả trong tài liệu JCTVC-S1005, của R. Joshi và J. Xu, “HEVC screen content coding draft text 2”, Nhóm hợp tác chung về mã hóa dữ liệu video (JCT-VC) của tổ chức ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 19: Strasbourg, FR, 17-24/10/2014.

Đối với khung HEVC, ví dụ, các kỹ thuật mã hóa dựa trên bảng màu có thể được tạo cấu hình để được sử dụng làm chế độ đơn vị mã hóa (coding unit - CU). Theo các ví dụ khác, các kỹ thuật mã hóa dựa trên bảng màu có thể được tạo cấu hình để

được sử dụng làm chế độ đơn vị dự báo (prediction unit - PU) trong khung HEVC. Do đó, tất cả các quy trình được bộc lộ sau đây được mô tả trong ngữ cảnh chế độ CU có thể, ngoài ra hoặc theo cách khác, áp dụng cho PU. Tuy nhiên, các ví dụ dựa trên HEVC này không nên được coi là giới hạn hoặc hạn chế của các kỹ thuật mã hóa dựa trên bảng màu được mô tả ở đây, vì các kỹ thuật đó có thể được áp dụng để làm việc độc lập hoặc là một phần của các hệ thống/chuẩn khác hiện có hoặc sắp được phát triển. Trong các trường hợp này, đơn vị để mã hóa bảng màu có thể là các khối vuông, các khối hình chữ nhật hoặc thậm chí là các vùng không phải hình chữ nhật.

Trong một số ví dụ, bảng màu có thể được suy ra cho một hoặc nhiều CU, PU, hoặc vùng dữ liệu bất kỳ (ví dụ, khối dữ liệu bất kỳ). Ví dụ, bảng màu có thể bao gồm (và có thể gồm có) các giá trị điểm ảnh trội nhất trong CU hiện thời, trong đó CU là vùng dữ liệu đối với ví dụ cụ thể này. Kích thước và các phần tử của bảng màu được truyền trước tiên từ bộ mã hóa dữ liệu video đến bộ giải mã dữ liệu video. Kích thước và/hoặc các phần tử của bảng màu có thể được mã hóa trực tiếp hoặc được mã hóa dự báo bằng cách sử dụng kích thước và/hoặc các phần tử của bảng màu trong các CU lân cận (ví dụ CU được mã hóa ở bên trên và/hoặc bên trái). Sau đó, các giá trị điểm ảnh trong CU được mã hóa dựa trên bảng màu theo trình tự quét nhất định. Đối với mỗi vị trí điểm ảnh trong CU, cờ (ví dụ, `palette_flag` hoặc `escape_flag`) có thể được truyền trước tiên để chỉ báo liệu giá trị điểm ảnh có được bao gồm trong bảng màu hay không. Đối với các giá trị điểm ảnh mà ánh xạ đến một mục nhập trong bảng màu, chỉ số bảng màu liên quan đến mục nhập đó được báo hiệu cho vị trí điểm ảnh cho trước trong CU. Thay vì gửi cờ (ví dụ, `palette_flag` hoặc `escape_flag`), đối với các giá trị điểm ảnh không tồn tại trong bảng màu, chỉ số đặc biệt có thể được gán cho điểm ảnh và giá trị điểm ảnh thực (có thể là dạng lượng tử hóa) có thể được truyền cho vị trí điểm ảnh cho trước trong CU. Các điểm ảnh này được gọi là “các điểm ảnh thoát”. Điểm ảnh thoát có thể được mã hóa bằng cách sử dụng phương pháp mã hóa entropy hiện có bất kỳ chẳng hạn như mã hóa độ dài cố định, mã hóa đơn phân, v.v.. Trong một số ví dụ, một hoặc nhiều kỹ thuật được mô tả ở đây có thể sử dụng cờ chẳng hạn như `palette_flag` hoặc `escape_flag`. Trong các ví dụ khác, một hoặc nhiều kỹ thuật được mô tả ở đây có thể không sử dụng cờ chẳng hạn như `palette_flag` hoặc `escape_flag`.

Các mẫu trong khối dữ liệu video có thể được xử lý (ví dụ, quét) bằng cách sử

dụng trình tự quét mảnh ngang hoặc trình tự quét khác. Ví dụ, bộ mã hóa dữ liệu video có thể chuyển đổi khối các chỉ số bảng màu hai chiều thành mảng một chiều bằng cách quét các chỉ số bảng màu sử dụng trình tự quét mảnh ngang. Tương tự, bộ giải mã dữ liệu video có thể tái tạo khối gồm các chỉ số bảng màu bằng cách sử dụng trình tự quét mảnh ngang. Theo đó, sáng chế có thể đề cập đến mẫu trước đó là mẫu đứng trước mẫu hiện đang được mã hóa trong khối theo trình tự quét. Cần phải hiểu rằng các trình tự quét ngoài trình tự quét mảnh ngang, chẳng hạn như trình tự quét mảnh dọc, cũng có thể áp dụng được. Ví dụ trên đây, cũng như các ví dụ khác được đưa ra trong bản mô tả này, dự định cung cấp sự mô tả chung về mã hóa dữ liệu video dựa trên bảng màu.

Fig.1 là sơ đồ khối thể hiện hệ thống mã hóa dữ liệu video 10 làm ví dụ mà có thể sử dụng các kỹ thuật trong bản mô tả này. Như được sử dụng ở đây, thuật ngữ “bộ mã hóa dữ liệu video” chỉ chung cả bộ mã hóa dữ liệu video lẫn bộ giải mã dữ liệu video. Trong bản mô tả này, các thuật ngữ “mã hóa dữ liệu video” hoặc “mã hóa” có thể chỉ chung việc mã hóa dữ liệu video hoặc giải mã dữ liệu video. Bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 của hệ thống mã hóa dữ liệu video 10 thể hiện các ví dụ về các thiết bị có thể được tạo cấu hình để thực hiện các kỹ thuật mã hóa dữ liệu video dựa trên bảng màu và mã hóa entropy (ví dụ, CABAC) theo các ví dụ khác nhau được mô tả trong bản mô tả này. Ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để mã hóa chọn lọc các khối dữ liệu video khác nhau, chẳng hạn như các CU hoặc PU trong mã hóa HEVC, bằng cách sử dụng kỹ thuật mã hóa dựa trên bảng màu hoặc mã hóa không dựa trên bảng màu. Các chế độ mã hóa không dựa trên bảng màu có thể được dùng để chỉ các chế độ mã hóa dự báo liên cấu trúc theo thời gian hoặc các chế độ mã hóa dự báo nội cấu trúc theo không gian khác nhau, chẳng hạn như các chế độ mã hóa khác nhau được quy định theo chuẩn HEVC.

Như được thể hiện trên Fig.1, hệ thống mã hóa dữ liệu video 10 bao gồm thiết bị nguồn 12 và thiết bị đích 14. Thiết bị nguồn 12 tạo ra dữ liệu video được mã hóa. Do đó, thiết bị nguồn 12 có thể được gọi là thiết bị mã hóa dữ liệu video hoặc cơ cấu mã hóa dữ liệu video. Thiết bị đích 14 có thể giải mã dữ liệu video mã hóa được tạo ra bởi thiết bị nguồn 12. Do đó, thiết bị đích 14 có thể được gọi là thiết bị giải mã dữ liệu video hoặc cơ cấu giải mã dữ liệu video. Thiết bị nguồn 12 và thiết bị đích 14 này có

thể là các ví dụ của các thiết bị mã hóa dữ liệu video hoặc cơ cấu mã hóa dữ liệu video.

Thiết bị nguồn 12 và thiết bị đích 14 có thể bao gồm một loạt các thiết bị, bao gồm máy tính để bàn, các thiết bị tính toán di động, các máy tính notebook (chẳng hạn, máy tính xách tay), các máy tính bảng, hộp giải mã tín hiệu, các máy điện thoại cầm tay chẳng hạn như điện thoại “thông minh”, ti vi, máy ảnh, các thiết bị hiển thị, các thiết bị phát phương tiện kỹ thuật số, các bàn giao tiếp chơi trò chơi video, máy tính trong xe ô tô, hoặc các thiết bị tương tự.

Thiết bị đích 14 có thể nhận dữ liệu video đã mã hóa từ thiết bị nguồn 12 qua kênh 16. Kênh 16 có thể bao gồm một hoặc nhiều phương tiện hoặc thiết bị có khả năng di chuyển dữ liệu video đã mã hóa từ thiết bị nguồn 12 đến thiết bị đích 14. Theo một ví dụ, kênh 16 có thể bao gồm một hoặc nhiều phương tiện truyền thông cho phép thiết bị nguồn 12 truyền dữ liệu video đã mã hóa trực tiếp đến thiết bị đích 14 theo thời gian thực. Trong ví dụ này, thiết bị nguồn 12 có thể điều biến dữ liệu video đã mã hóa theo chuẩn truyền thông, chẳng hạn như giao thức truyền thông không dây, và có thể truyền dữ liệu video đã điều biến đến thiết bị đích 14. Một hoặc nhiều phương tiện truyền thông có thể bao gồm các phương tiện truyền thông không dây và/hoặc có dây, chẳng hạn như phổ tần số vô tuyến (RF - radio frequency) hoặc một hoặc nhiều đường truyền vật lý. Một hoặc nhiều phương tiện truyền thông này có thể tạo thành một phần của mạng dựa trên gói, chẳng hạn như mạng cục bộ, mạng diện rộng, hoặc mạng toàn cầu (chẳng hạn, mạng Internet). Một hoặc nhiều phương tiện truyền thông có thể bao gồm các bộ định tuyến, các bộ chuyển mạch, các trạm cơ sở, hoặc thiết bị khác hỗ trợ truyền thông từ thiết bị nguồn 12 đến thiết bị đích 14.

Theo ví dụ khác, kênh 16 có thể bao gồm vật ghi lưu trữ để lưu trữ dữ liệu video đã mã hóa được tạo ra bởi thiết bị nguồn 12. Trong ví dụ này, thiết bị đích 14 có thể truy cập vật ghi lưu trữ qua, ví dụ, truy cập đĩa hoặc truy cập thẻ. Vật ghi lưu trữ này có thể bao gồm nhiều vật ghi lưu trữ dữ liệu được truy cập cục bộ chẳng hạn như đĩa Blu-ray, DVD, CD-ROM, bộ nhớ nhanh, hoặc các vật ghi lưu trữ kỹ thuật số phù hợp khác để lưu trữ dữ liệu video đã mã hóa.

Trong một ví dụ khác, kênh 16 có thể bao gồm máy chủ tập tin hoặc thiết bị lưu trữ trung gian khác lưu trữ dữ liệu video đã mã hóa được tạo ra bởi thiết bị nguồn 12. Trong ví dụ này, thiết bị đích 14 có thể truy cập dữ liệu video đã mã hóa được lưu trữ

ở máy chủ tập tin hoặc thiết bị lưu trữ trung gian khác qua tạo luồng hoặc tải về. Máy chủ tập tin có thể là kiểu máy chủ có khả năng lưu trữ dữ liệu video đã mã hóa và truyền dữ liệu video đã mã hóa này đến thiết bị đích 14. Các máy chủ tập tin làm ví dụ bao gồm máy chủ web (chẳng hạn, cho trang web), máy chủ giao thức truyền tập tin (file transfer protocol - FTP), thiết bị lưu trữ gắn mạng (network attached storage - NAS), và các ổ đĩa cục bộ.

Thiết bị đích 14 có thể truy cập dữ liệu video đã mã hóa qua kết nối dữ liệu chuẩn, chẳng hạn như kết nối Internet. Ví dụ về các kiểu kết nối dữ liệu có thể bao gồm các kênh không dây (ví dụ, kết nối Wi-Fi), kết nối có dây (ví dụ, DSL, môđem cáp, v.v.), hoặc dạng kết hợp của chúng mà thích hợp để truy cập dữ liệu video đã mã hóa được lưu trữ trên máy chủ tập tin. Cuộc truyền dữ liệu video đã mã hóa từ máy chủ tập tin có thể là cuộc truyền theo luồng, truyền tải về hoặc dạng kết hợp của cả hai.

Thiết bị nguồn 12 và thiết bị đích 14 có thể được tạo cấu hình để thực hiện mã hóa dựa trên bảng màu và mã hóa entropy (ví dụ, CABAC) theo sáng chế. Tuy nhiên, các kỹ thuật của bản mô tả này để mã hóa dựa trên bảng màu hoặc CABAC không bị giới hạn ở các ứng dụng hoặc các thiết lập không dây. Các kỹ thuật này có thể được áp dụng cho mã hóa dữ liệu video để hỗ trợ nhiều ứng dụng đa phương tiện, chẳng hạn như phát sóng truyền hình qua không gian, truyền hình cáp, truyền hình vệ tinh, truyền video theo luồng, chẳng hạn qua mạng Internet, mã hóa dữ liệu video để lưu trữ vào vật ghi lưu trữ dữ liệu, giải mã dữ liệu video lưu trên vật ghi lưu trữ dữ liệu, hoặc các ứng dụng khác. Theo một số ví dụ, hệ thống mã hóa dữ liệu video 10 có thể được tạo cấu hình để hỗ trợ truyền video một chiều hoặc hai chiều để hỗ trợ các ứng dụng như tạo luồng video, phát lại video, phát rộng video, và/hoặc điện thoại video.

Hệ thống mã hóa dữ liệu video 10 được minh họa trên Fig.1 chỉ là ví dụ và các kỹ thuật của bản mô tả này có thể áp dụng cho các thiết lập mã hóa dữ liệu video (ví dụ, mã hóa dữ liệu video hoặc giải mã dữ liệu video) mà không nhất thiết bao gồm cuộc truyền dữ liệu bất kỳ giữa các thiết bị mã hóa và giải mã. Trong các ví dụ khác, dữ liệu được truy hồi từ bộ nhớ cục bộ, được tạo luồng trên mạng, hoặc tương tự. Thiết bị mã hóa dữ liệu video có thể mã hóa và lưu trữ dữ liệu vào bộ nhớ và/hoặc thiết bị giải mã dữ liệu video có thể truy hồi và giải mã dữ liệu từ bộ nhớ. Trong nhiều ví dụ, việc mã hóa và giải mã được thực hiện bởi các thiết bị không truyền thông với nhau mà chỉ mã hóa dữ liệu vào bộ nhớ và/hoặc truy hồi và giải mã dữ liệu từ bộ nhớ.

Trong ví dụ trên Fig.1, thiết bị nguồn 12 bao gồm nguồn dữ liệu video 18, bộ mã hóa dữ liệu video 20, và giao diện xuất 22. Trong một số ví dụ, giao diện xuất 22 có thể bao gồm bộ điều biến/bộ giải điều biến (modem - modulator/demodulator) và/hoặc bộ truyền. Nguồn dữ liệu video 18 có thể bao gồm thiết bị ghi video, ví dụ máy quay video, bộ lưu trữ video chứa các dữ liệu video đã được ghi trước đó, giao diện nạp video để nhận dữ liệu video từ nhà cung cấp nội dung video, và/hoặc hệ thống đồ họa máy tính để tạo ra dữ liệu video, hoặc tổ hợp của các nguồn dữ liệu video này.

Bộ mã hóa dữ liệu video 20 có thể mã hóa dữ liệu video từ nguồn dữ liệu video 18. Theo một số ví dụ, thiết bị nguồn 12 truyền trực tiếp dữ liệu video đã mã hóa đến thiết bị đích 14 qua giao diện xuất 22. Trong các ví dụ khác, dữ liệu video đã mã hóa cũng có thể được lưu trữ trên vật ghi lưu trữ hoặc máy chủ tập tin để truy cập sau bằng thiết bị đích 14 để giải mã và/hoặc phát lại.

Trong ví dụ trên Fig.1, thiết bị đích 14 bao gồm giao diện nhập 28, bộ giải mã dữ liệu video 30, và thiết bị hiển thị 32. Trong một số ví dụ, giao diện nhập 28 bao gồm bộ thu và/hoặc môđem. Giao diện nhập 28 có thể nhận được dữ liệu video đã mã hóa qua kênh 16. Thiết bị hiển thị 32 có thể được tích hợp vào hoặc có thể ở bên ngoài thiết bị đích 14. Nói chung, thiết bị hiển thị 32 hiển thị dữ liệu video đã giải mã. Thiết bị hiển thị 32 có thể bao gồm nhiều thiết bị hiển thị, chẳng hạn như màn hình tinh thể lỏng (liquid crystal display - LCD), màn hình plasma, màn hình điốt phát quang hữu cơ (organic light emitting diode - OLED), hoặc loại thiết bị hiển thị khác.

Nói chung, bản mô tả này có thể đề cập đến bộ mã hóa dữ liệu video 20 “báo hiệu” hoặc “truyền” thông tin nhất định đến một thiết bị khác, chẳng hạn như bộ giải mã dữ liệu video 30. Thuật ngữ “báo hiệu” hoặc “truyền” nói chung có thể được dùng để chỉ việc truyền các phần tử cú pháp và/hoặc dữ liệu khác được dùng để giải mã dữ liệu video nén. Việc truyền thông này có thể xảy ra theo thời gian thực hoặc gần thực. Theo cách khác, việc truyền thông này có thể xảy ra trong một khoảng thời gian, như có thể xảy ra khi lưu trữ các phần tử cú pháp vào vật ghi đọc được bằng máy tính trong dòng bit mã hóa tại thời điểm mã hóa, mà sau đó có thể được truy hồi bởi thiết bị giải mã tại thời điểm bất kỳ sau khi lưu trữ vào vật ghi này. Do đó, trong khi bộ giải mã dữ liệu video 30 có thể được gọi là “nhận” một số thông tin nhất định, việc nhận thông tin này không nhất thiết diễn ra theo thời gian thực hoặc gần thực và có thể được truy hồi

từ vật ghi tại thời điểm bất kỳ sau khi lưu trữ.

Mỗi bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được thực hiện dưới dạng mạch mã hóa bất kỳ trong số rất nhiều mạch mã hóa thích hợp, chẳng hạn như một hoặc nhiều bộ vi xử lý, bộ xử lý tín hiệu số (DSP - digital signal processor), mạch tích hợp chuyên dụng (ASIC - application-specific integrated circuit), mảng cổng lập trình được dạng trường (FPGA - field-programmable gate array), logic rời rạc, phần cứng hoặc tổ hợp bất kỳ của chúng. Nếu các kỹ thuật được thực hiện từng phần trong phần mềm, thì thiết bị có thể lưu trữ các lệnh cho phần mềm trong vật ghi bất biến đọc được bằng máy tính thích hợp và có thể thực thi các lệnh này trong phần cứng bằng cách sử dụng một hoặc nhiều bộ xử lý để thực hiện các kỹ thuật theo sáng chế. Phần bất kỳ trong số các phần nêu trên (bao gồm phần cứng, phần mềm, tổ hợp của phần cứng và phần mềm, v.v.) có thể được coi là một hoặc nhiều bộ xử lý. Mỗi bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được bao gồm trong một hoặc nhiều bộ mã hóa hoặc bộ giải mã, mỗi bộ phận này có thể được tích hợp làm một phần của bộ mã hóa/giải mã (encoder/decoder - CODEC) kết hợp trong thiết bị tương ứng.

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 hoạt động theo một chuẩn nén video, chẳng hạn như chuẩn HEVC nêu trên, và được mô tả trong chuẩn HEVC. Ngoài chuẩn HEVC cơ bản, hiện đang có nhiều nỗ lực để tạo ra mã hóa dữ liệu video có thể thay đổi, mã hóa dữ liệu video đa khung hình, và các phần mở rộng mã hóa 3D cho HEVC. Ngoài ra, các chế độ mã hóa dựa trên bảng màu, chẳng hạn, như được mô tả trong bản mô tả này, có thể được cung cấp cho phần mở rộng của chuẩn HEVC. Trong một số ví dụ, các kỹ thuật được mô tả trong bản mô tả này để mã hóa dựa trên bảng màu có thể được áp dụng cho các bộ mã hóa và các bộ giải mã được tạo cấu hình để hoạt động theo các chuẩn mã hóa dữ liệu video khác. Do đó, việc áp dụng chế độ mã hóa dựa trên bảng màu để mã hóa các đơn vị mã hóa (coding unit – CU) hoặc các đơn vị dự báo (prediction unit - PU) trong bộ mã hóa-giải mã HEVC được mô tả nhằm mục đích minh họa.

Trong HEVC và các chuẩn mã hóa dữ liệu video khác, chuỗi video thường bao gồm một loạt các hình ảnh. Các hình ảnh còn được gọi là “các khung”. Hình ảnh có thể bao gồm ba mảng mẫu, được ký hiệu là S_L , S_{Cb} và S_{Cr} . S_L là mảng ((tức là, khối) các mẫu độ chói hai chiều. S_{Cb} là mảng các mẫu màu Cb hai chiều. S_{Cr} là mảng các

mẫu màu Cr hai chiều. Ở đây các mẫu màu cũng có thể được gọi là các mẫu "sắc độ". Trong các trường hợp khác, hình ảnh có thể là đơn sắc và có thể chỉ bao gồm mảng mẫu độ chói.

Để tạo ra dạng biểu diễn mã hóa của hình ảnh, bộ mã hóa dữ liệu video 20 có thể tạo ra tập hợp đơn vị cây mã hóa (CTU - coding tree unit). Mỗi CTU có thể là khối cây mã hóa gồm các mẫu độ chói, hai khối cây mã hóa tương ứng gồm các mẫu màu, và các cấu trúc cú pháp được sử dụng để mã hóa các mẫu của các khối cây mã hóa. Khối cây mã hóa có thể là khối các mẫu NxN. CTU cũng có thể được gọi là “khối cây” hoặc “đơn vị mã hóa lớn nhất” (LCU - largest coding unit). Nói chung các CTU của HEVC có thể giống với các khối macro của các chuẩn khác, như H.264/AVC. Tuy nhiên, CTU không nhất thiết chỉ giới hạn ở kích thước cụ thể và có thể bao gồm một hoặc nhiều đơn vị mã hóa (CU - coding unit). Lát có thể bao gồm một số nguyên các CTU được sắp xếp liên tiếp theo trình tự quét mảnh. Lát đã mã hóa có thể bao gồm phần đầu lát và dữ liệu lát. Phần đầu lát của lát có thể là cấu trúc cú pháp mà bao gồm các phần tử cú pháp cung cấp thông tin về lát này. Dữ liệu lát có thể bao gồm các CTU được mã hóa của lát.

Bản mô tả này có thể sử dụng thuật ngữ “đơn vị dữ liệu video” hoặc “khối dữ liệu video” hoặc “khối” để chỉ một hoặc nhiều khối mẫu và các cấu trúc cú pháp được sử dụng để mã hóa các mẫu của một hoặc nhiều khối mẫu. Ví dụ về các kiểu đơn vị hoặc khối dữ liệu video có thể bao gồm CTU, CU, PU, các đơn vị biến đổi (transform unit - TU), các khối macro, các phần chia khối macro, v.v.. Trong một số trường hợp, phần mô tả các PU có thể được dùng thay thế cho phần mô tả các khối macro hoặc các phần chia khối macro.

Để tạo ra CTU mã hóa, bộ mã hóa dữ liệu video 20 có thể thực hiện một cách đệ quy sự phân chia cây tứ phân trên các khối cây mã hóa của CTU để chia khối cây mã hóa này thành các khối mã hóa, do đó có tên là “các đơn vị cây mã hóa”. Khối mã hóa là khối mẫu NxN. CU có thể là khối mã hóa gồm các mẫu độ chói và hai khối mã hóa tương ứng gồm các mẫu màu của hình ảnh mà có mảng mẫu độ chói, mảng mẫu Cb và mảng mẫu Cr, và các cấu trúc cú pháp được sử dụng để mã hóa các mẫu của các khối mã hóa. Bộ mã hóa dữ liệu video 20 có thể chia khối mã hóa của CU thành một hoặc nhiều khối dự báo. Khối dự báo có thể là khối mẫu hình chữ nhật (chẳng hạn, hình vuông hoặc không vuông) mà áp dụng cùng một kỹ thuật dự báo trên đó. Đơn vị

dự báo (prediction unit - PU) của CU có thể là khối dự báo gồm các mẫu độ chói, hai khối dự báo tương ứng gồm các mẫu màu của hình ảnh, và các cấu trúc cú pháp được sử dụng để dự báo các mẫu khối dự báo. Bộ mã hóa dữ liệu video 20 có thể tạo ra các khối dự báo độ chói, Cb, và Cr cho các khối dự báo độ chói, Cb, và Cr của mỗi PU của CU.

Bộ mã hóa dữ liệu video 20 có thể sử dụng dự báo nội cấu trúc hoặc dự báo liên cấu trúc để tạo ra các khối dự báo cho PU. Nếu bộ mã hóa dữ liệu video 20 sử dụng dự báo nội cấu trúc để tạo ra khối dự báo của PU, thì bộ mã hóa dữ liệu video 20 có thể tạo ra các khối dự báo của PU dựa trên các mẫu được giải mã của hình ảnh liên quan đến PU.

Nếu bộ mã hóa dữ liệu video 20 sử dụng dự báo liên cấu trúc để tạo ra các khối dự báo của PU, thì bộ mã hóa dữ liệu video 20 có thể tạo ra các khối dự báo của PU dựa trên các mẫu được giải mã của một hoặc nhiều hình ảnh không phải hình ảnh liên quan đến PU. Bộ mã hóa dữ liệu video 20 có thể sử dụng dự báo đơn hoặc dự báo đôi để tạo ra các khối dự báo của PU. Khi bộ mã hóa dữ liệu video 20 sử dụng dự báo đơn để tạo ra các khối dự báo cho PU, thì PU này có thể có một vector chuyển động (motion vector - MV). Khi bộ mã hóa dữ liệu video 20 sử dụng dự báo đôi để tạo ra các khối dự báo cho PU, PU này có thể có hai MV.

Sau khi bộ mã hóa dữ liệu video 20 tạo ra các khối dự báo (chẳng hạn, các khối dự báo độ chói, Cb và Cr) cho một hoặc nhiều PU của CU, bộ mã hóa dữ liệu video 20 có thể tạo ra các khối dư cho CU này. Mỗi mẫu trong khối dư của CU có thể biểu thị chênh lệch giữa mẫu trong khối dự báo của PU của CU và mẫu tương ứng trong khối mã hóa của CU. Ví dụ, bộ mã hóa dữ liệu video 20 có thể tạo ra khối dư độ chói cho CU. Mỗi mẫu trong khối dư độ chói của CU chỉ báo chênh lệch giữa mẫu độ chói trong một trong số các khối độ chói dự báo của CU và mẫu tương ứng trong khối mã hóa độ chói gốc của CU. Ngoài ra, bộ mã hóa dữ liệu video 20 có thể tạo ra khối dư Cb cho CU. Mỗi mẫu trong khối dư Cb của CU có thể chỉ báo chênh lệch giữa mẫu Cb trong một trong số các khối Cb dự báo của CU và mẫu tương ứng trong khối mã hóa Cb gốc của CU. Bộ mã hóa dữ liệu video 20 còn có thể tạo ra khối dư Cr cho CU. Mỗi mẫu trong khối dư Cr của CU chỉ báo chênh lệch giữa mẫu Cr trong một trong số các khối Cr dự báo của CU và mẫu tương ứng trong khối mã hóa Cr gốc của CU.

Ngoài ra, bộ mã hóa dữ liệu video 20 có thể sử dụng phép phân chia cây tứ

phân để phân tích các khối dư (chẳng hạn, các khối dư độ chói, Cb và Cr) của CU thành một hoặc nhiều khối biến đổi (chẳng hạn, các khối biến đổi độ chói, Cb và Cr). Khối biến đổi có thể là khối mẫu hình chữ nhật mà áp dụng cùng một phép biến đổi trên đó. Đơn vị biến đổi (transform unit - TU) của CU có thể là khối biến đổi gồm các mẫu độ chói, hai khối biến đổi tương ứng gồm các mẫu màu, và các cấu trúc cú pháp được sử dụng để biến đổi các mẫu khối biến đổi này. Do vậy, mỗi TU của CU có thể được kết hợp với khối biến đổi độ chói, khối biến đổi Cb, và khối biến đổi Cr. Khối biến đổi độ chói gắn với TU có thể là khối con của khối dư độ chói của CU. Khối biến đổi Cb có thể là khối con của khối dư Cb của CU. Khối biến đổi Cr có thể là khối con của khối dư Cr của CU.

Bộ mã hóa dữ liệu video 20 có thể áp dụng một hoặc nhiều phép biến đổi cho khối biến đổi để tạo ra khối hệ số cho TU. Khối hệ số có thể là mảng các hệ số biến đổi hai chiều. Hệ số biến đổi có thể là đại lượng vô hướng. Ví dụ, bộ mã hóa dữ liệu video 20 có thể áp dụng một hoặc nhiều phép biến đổi cho khối biến đổi độ chói của TU để tạo ra khối hệ số độ chói cho TU này. Bộ mã hóa dữ liệu video 20 có thể áp dụng một hoặc nhiều phép biến đổi cho khối biến đổi Cb của TU để tạo ra khối hệ số Cb cho TU. Bộ mã hóa dữ liệu video 20 có thể áp dụng một hoặc nhiều phép biến đổi cho khối biến đổi Cr của TU để tạo ra khối hệ số Cr cho TU.

Sau khi tạo ra khối hệ số (ví dụ, khối hệ số độ chói, khối hệ số Cb hoặc khối hệ số Cr), bộ mã hóa dữ liệu video 20 có thể lượng tử hóa khối hệ số. Lượng tử hóa thường là quy trình trong đó các hệ số biến đổi được lượng tử hóa để có thể giảm bớt lượng dữ liệu được sử dụng để biểu diễn các hệ số biến đổi, nhằm đạt được hiệu quả nén cao hơn. Sau khi bộ mã hóa dữ liệu video 20 lượng tử hóa khối hệ số, bộ mã hóa dữ liệu video 20 có thể mã hóa entropy các phần tử cú pháp chỉ báo các hệ số biến đổi đã được lượng tử hóa. Ví dụ, bộ mã hóa dữ liệu video 20 có thể thực hiện mã hóa số học nhị phân thích ứng theo ngữ cảnh (Context-Adaptive Binary Arithmetic Coding - CABAC) trên các phần tử cú pháp chỉ báo các hệ số biến đổi được lượng tử hóa.

Đối với CABAC, ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể chọn mô hình xác suất (hay còn gọi là mô hình ngữ cảnh) để mã hóa các ký hiệu gắn với khối dữ liệu video dựa trên ngữ cảnh. Ví dụ, mô hình ngữ cảnh (Ctx) có thể là chỉ số hoặc độ dịch mà được áp dụng để lựa chọn một trong số nhiều ngữ cảnh khác nhau, mỗi ngữ cảnh có thể tương ứng với một mô hình xác suất cụ thể. Theo đó,

mô hình xác suất khác nhau thường được xác định cho mỗi ngữ cảnh. Sau khi mã hóa hoặc giải mã bin, mô hình xác suất còn được cập nhật dựa trên giá trị của bin để phản ánh ước tính xác suất mới nhất cho bin đó. Ví dụ, mô hình xác suất có thể được duy trì dưới dạng một trạng thái trong máy trạng thái hữu hạn. Mỗi trạng thái cụ thể có thể tương ứng với giá trị xác suất cụ thể. Trạng thái tiếp theo, tương ứng với lần cập nhật mô hình xác suất, có thể tùy thuộc vào giá trị của bin hiện thời (ví dụ, bin hiện đang được mã hóa). Theo đó, việc lựa chọn mô hình xác suất có thể bị ảnh hưởng bởi các giá trị của các bin được mã hóa trước đó, do các giá trị chỉ báo, ít nhất là một phần, xác suất của bin có giá trị cho trước. Quy trình mã hóa ngữ cảnh được mô tả trên đây có thể được gọi chung là chế độ mã hóa thích nghi theo ngữ cảnh.

Do đó, bộ mã hóa dữ liệu video 20 có thể mã hóa ký hiệu đích bằng cách sử dụng mô hình xác suất. Tương tự, bộ giải mã dữ liệu video 30 có thể phân tích cú pháp ký hiệu đích bằng cách sử dụng mô hình xác suất. Trong một số trường hợp, bộ mã hóa dữ liệu video 20 có thể mã hóa các phần tử cú pháp bằng cách sử dụng sự kết hợp giữa mã hóa thích nghi theo ngữ cảnh và mã hóa không thích nghi theo ngữ cảnh. Ví dụ, bộ mã hóa dữ liệu video 20 có thể mã hóa theo ngữ cảnh các bin bằng cách lựa chọn mô hình xác suất hoặc “mô hình ngữ cảnh” mà hoạt động theo ngữ cảnh để mã hóa một số bin. Ngược lại, đối với các bin khác, bộ mã hóa dữ liệu video 20 có thể mã hóa bỏ qua các bin này bằng cách bỏ qua, hoặc loại bỏ quy trình mã hóa số học thông thường khi mã hóa các bin này. Trong các ví dụ này, bộ mã hóa dữ liệu video 20 có thể sử dụng mô hình xác suất cố định để mã hóa bỏ qua các bin. Tức là, các bin được mã hóa bỏ qua không có các cập nhật về ngữ cảnh hoặc xác suất.

Bộ mã hóa dữ liệu video 20 có thể xuất ra dòng bit bao gồm các phần tử cú pháp được mã hóa entropy. Dòng bit này cũng có thể bao gồm các phần tử cú pháp không được mã hóa entropy. Dòng bit có thể bao gồm chuỗi các bit mà tạo thành dạng biểu diễn của các hình ảnh được mã hóa và dữ liệu kết hợp. Dòng bit này có thể bao gồm chuỗi các đơn vị lớp trừu tượng hóa mạng (network abstraction layer - NAL). Mỗi trong số các đơn vị NAL bao gồm phần đầu đơn vị NAL và đóng gói phần tải tin chuỗi byte thô (raw byte sequence payload - RBSP). Phần đầu đơn vị NAL có thể bao gồm phần tử cú pháp chỉ báo mã kiểu đơn vị NAL. Mã kiểu đơn vị NAL được chỉ rõ bởi phần đầu đơn vị NAL của đơn vị NAL chỉ báo kiểu đơn vị NAL. RBSP có thể là cấu trúc cú pháp chứa số lượng byte nguyên mà được đóng gói trong đơn vị NAL.

Trong một số trường hợp, RBSP bao gồm các bit 0.

Các kiểu đơn vị NAL khác nhau có thể đóng gói các kiểu RBSP khác nhau. Ví dụ, kiểu đơn vị NAL thứ nhất có thể đóng gói RBSP cho tập tham số hình ảnh (picture parameter set - PPS), kiểu đơn vị NAL thứ hai có thể đóng gói RBSP cho lát mã hóa, kiểu đơn vị NAL thứ ba có thể đóng gói RBSP cho thông tin nâng cao bổ sung (supplemental enhancement thông tin - SEI), v.v.. Đơn vị NAL mà đóng gói các RBSP cho dữ liệu mã hóa video (trái với các RBSP cho các tập thông số và các thông báo SEI) có thể được gọi là các đơn vị NAL lớp mã hóa video (video coding layer - VCL).

Bộ giải mã dữ liệu video 30 có thể nhận dòng bit được tạo ra bởi bộ mã hóa dữ liệu video 20. Ngoài ra, bộ giải mã dữ liệu video 30 có thể phân tích cú pháp dòng bit để giải mã các phần tử cú pháp từ dòng bit này. Bộ giải mã dữ liệu video 30 có thể tái tạo các hình ảnh của dữ liệu video dựa ít nhất một phần vào các phần tử cú pháp được giải mã từ dòng bit. Nói chung quy trình tái tạo dữ liệu video có thể ngược với quy trình được thực hiện bởi bộ mã hóa dữ liệu video 20. Ví dụ, bộ giải mã dữ liệu video 30 có thể sử dụng các MV của các PU để xác định các khối dự báo cho các PU được dự báo liên cấu trúc của CU hiện thời. Tương tự, bộ giải mã dữ liệu video 30 có thể tạo ra các khối được dự báo nội cấu trúc cho các PU của CU hiện thời. Ngoài ra, bộ giải mã dữ liệu video 30 có thể lượng tử hóa ngược các khối hệ số biến đổi gắn với các TU của CU hiện thời. Bộ giải mã dữ liệu video 30 có thể thực hiện biến đổi ngược trên các khối hệ số biến đổi để tái tạo các khối biến đổi gắn với các TU của CU hiện thời. Bộ giải mã dữ liệu video 30 có thể tái tạo khối mã hóa của CU hiện thời bằng cách bổ sung các mẫu của khối dự báo cho các PU của CU hiện thời vào các giá trị dư tương ứng thu được từ quy trình lượng tử hóa ngược và biến đổi ngược các khối biến đổi của các TU của CU hiện thời. Bằng cách tái tạo các khối mã hóa cho mỗi CU của hình ảnh, bộ giải mã dữ liệu video 30 có thể tái tạo hình ảnh này.

Theo một số ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để thực hiện mã hóa dựa trên bảng màu. Ví dụ, trong kỹ thuật mã hóa dựa trên bảng màu, thay vì thực hiện các kỹ thuật mã hóa dự báo nội cấu trúc hoặc dự báo liên cấu trúc được mô tả trên đây, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể mã hóa bảng màu dưới dạng một bảng các màu sắc hoặc các giá trị điểm ảnh biểu diễn dữ liệu video của một vùng cụ thể (ví dụ, khối cho trước). Theo cách này, thay vì mã hóa các giá trị điểm ảnh thực hoặc các phần dư của

chúng cho khối dữ liệu video hiện thời, bộ mã hóa dữ liệu video có thể mã hóa các giá trị chỉ số cho một hoặc nhiều giá trị điểm ảnh của khối hiện thời, trong đó các giá trị chỉ số này biểu thị các mục nhập trong bảng màu mà được sử dụng để biểu diễn giá trị điểm ảnh của khối hiện thời.

Ví dụ, bộ mã hóa dữ liệu video 20 có thể mã hóa khối dữ liệu video bằng cách xác định bảng màu cho khối, định vị mục nhập trong bảng màu để biểu diễn giá trị của mỗi điểm ảnh, và mã hóa bảng màu và các giá trị chỉ số cho các điểm ảnh liên kết giá trị điểm ảnh với bảng màu. Bộ giải mã dữ liệu video 30 có thể thu được, từ dòng bit mã hóa, bảng màu cho khối, cũng như các giá trị chỉ số cho các điểm ảnh của khối này. Bộ giải mã dữ liệu video 30 có thể so khớp các giá trị chỉ số của các điểm ảnh riêng lẻ với các mục nhập của bảng màu để tái tạo các giá trị điểm ảnh của khối. Trong các trường hợp mà giá trị chỉ số gắn với điểm ảnh riêng lẻ không khớp với giá trị chỉ số bất kỳ của bảng màu tương ứng cho khối, bộ giải mã dữ liệu video 30 có thể nhận dạng điểm ảnh đó là điểm ảnh thoát, để mã hóa dựa trên bảng màu.

Như được mô tả chi tiết hơn dưới đây, ý tưởng cơ bản của kỹ thuật mã hóa dựa trên bảng màu đó là, đối với khối dữ liệu video cho trước cần được mã hóa, bộ mã hóa dữ liệu video 20 có thể suy ra bảng màu mà bao gồm các giá trị điểm ảnh trội nhất trong khối hiện thời. Ví dụ, bảng màu này có thể chỉ số lượng giá trị điểm ảnh được xác định hoặc giả định là trội và/hoặc đại diện cho CU hiện thời. Trước tiên, bộ mã hóa dữ liệu video 20 có thể truyền kích thước và các phần tử của bảng màu đến bộ giải mã dữ liệu video 30. Ngoài ra, bộ mã hóa dữ liệu video 20 có thể mã hóa các giá trị điểm ảnh trong khối cho trước theo trình tự quét nhất định. Đối với mỗi điểm ảnh có trong khối cho trước, bộ mã hóa dữ liệu video 20 có thể báo hiệu giá trị chỉ số mà ánh xạ giá trị điểm ảnh đến mục nhập tương ứng trong bảng màu. Nếu giá trị điểm ảnh không có trong bảng màu (tức là, không có mục nhập bảng màu nào tồn tại để chỉ rõ giá trị điểm ảnh cụ thể của khối được mã hóa trong bảng màu), thì điểm ảnh này được xác định là “điểm ảnh thoát”. Theo kỹ thuật mã hóa dựa trên bảng màu, bộ mã hóa dữ liệu video 20 có thể mã hóa và báo hiệu giá trị chỉ số dành riêng cho điểm ảnh thoát. Trong một số ví dụ, bộ mã hóa dữ liệu video 20 còn có thể mã hóa và báo hiệu giá trị điểm ảnh (hoặc phiên bản lượng tử hóa của nó) cho điểm ảnh thoát có trong khối cho trước. Ví dụ, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định liệu giá trị điểm ảnh có khớp hoặc nếu không thì gần với mục nhập bảng màu dựa trên giá trị

đo độ méo (ví dụ, MSE, SAD, và tương tự) hay không.

Khi nhận dòng bit video mã hóa được báo hiệu bởi bộ mã hóa dữ liệu video 20, bộ giải mã dữ liệu video 30 trước tiên có thể xác định bảng màu dựa trên thông tin nhận được từ bộ mã hóa dữ liệu video 20. Bộ giải mã dữ liệu video 30 sau đó có thể ánh xạ các giá trị chỉ số nhận được gắn với các vị trí điểm ảnh trong khối cho trước lên các mục nhập của bảng màu để tái tạo các giá trị điểm ảnh của khối cho trước. Trong một số trường hợp, bộ giải mã dữ liệu video 30 có thể xác định rằng điểm ảnh của khối được mã hóa bằng bảng màu là điểm ảnh thoát, chẳng hạn như bằng cách xác định rằng điểm ảnh được mã hóa bằng bảng màu với giá trị chỉ số dành riêng cho các điểm ảnh thoát. Trong các trường hợp mà bộ giải mã dữ liệu video 30 nhận dạng điểm ảnh thoát trong khối được mã hóa bằng bảng màu, bộ giải mã dữ liệu video 30 có thể nhận giá trị điểm ảnh (hoặc phiên bản lượng tử hóa của nó) cho điểm ảnh thoát có trong khối cho trước. Bộ giải mã dữ liệu video 30 có thể tái tạo khối được mã hóa bằng bảng màu bằng cách ánh xạ các giá trị điểm ảnh riêng lẻ lên các mục nhập bảng màu tương ứng, và bằng cách sử dụng giá trị điểm ảnh (hoặc phiên bản được lượng tử hóa của nó) để tái tạo các điểm ảnh thoát bất kỳ có trong khối được mã hóa bằng bảng màu.

Như nêu trên, trong chế độ mã hóa theo bảng màu làm ví dụ, bảng màu có thể bao gồm các mục nhập được đánh số bởi chỉ số. Mỗi mục nhập có thể biểu diễn các giá trị hoặc cường độ thành phần màu sắc (ví dụ, trong các không gian màu chẳng hạn như YCbCr, RGB, YUV, CMYK, hoặc các định dạng khác), có thể được dùng làm bộ dự báo cho khối hoặc làm các mẫu khối tái tạo cuối cùng. Như được mô tả trong tài liệu chuẩn của JCTVC-Q0094 (Wei Pu và cộng sự, “AHG10: Suggested Software for Palette Coding based on RExt6.0”, JCTVC-Q0094, Valencia, ES, 27/3– 4/4/2014), bảng màu có thể bao gồm các mục nhập được sao chép từ bảng màu của bộ dự báo. Bảng màu của bộ dự báo có thể bao gồm các mục nhập bảng màu từ các khối được mã hóa trước đó bằng cách sử dụng chế độ bảng màu hoặc các mẫu tái tạo khác. Đối với mỗi mục nhập trong bảng màu của bộ dự báo, cờ nhị phân được gửi đi để chỉ báo liệu mục nhập đó có được sao chép cho bảng màu hiện thời hay không (được chỉ báo bằng cờ = 1). Đây được gọi là vector dự báo bảng màu nhị phân. Ngoài ra, bảng màu hiện thời có thể bao gồm các mục nhập mới được báo hiệu rõ ràng. Số lượng mục nhập mới cũng có thể được báo hiệu.

Theo ví dụ khác, trong chế độ bảng màu, bảng màu có thể bao gồm các mục

nhập được đánh số bằng chỉ số biểu diễn các giá trị thành phần màu sắc mà có thể được dùng làm bộ dữ báo cho các mẫu khối hoặc làm các mẫu khối được tái tạo cuối cùng. Mỗi mục nhập trong bảng màu có thể chứa, ví dụ, một thành phần độ chói (ví dụ, giá trị độ chói), hai thành phần màu (ví dụ, hai giá trị màu), hoặc ba thành phần màu sắc (ví dụ, RGB, YUV, v.v.). Các mục nhập bảng màu được giải mã trước có thể được lưu trữ vào danh sách. Danh sách này có thể được dùng để dự báo các mục nhập bảng màu trong CU của chế độ bảng màu hiện thời, chẳng hạn. Vectơ dự báo nhị phân có thể được báo hiệu trong dòng bit để chỉ báo các mục nhập nào trong danh sách được tái sử dụng trong bảng màu hiện thời. Trong một số ví dụ, kỹ thuật mã hóa loạt dài có thể được dùng để nén bộ dữ báo bảng màu nhị phân. Ví dụ, giá trị độ dài loạt có thể được mã hóa bằng cách sử dụng mã Exp-Golomb bậc 0.

Trong bản mô tả này, giả sử rằng mỗi mục nhập bảng màu chỉ rõ các giá trị cho toàn bộ các thành phần màu sắc của một mẫu. Tuy nhiên, các khái niệm của sáng chế có thể ứng dụng được cho việc sử dụng bảng màu riêng lẻ và/hoặc mục nhập bảng màu riêng lẻ đối với mỗi thành phần màu sắc. Ngoài ra, giả sử rằng các mẫu trong khối được xử lý bằng cách sử dụng trình tự quét mảnh ngang. Tuy nhiên, các trình tự quét khác như trình tự quét mảnh dọc cũng áp dụng được. Như nêu trên, bảng màu có thể chứa các mục nhập bảng màu được dự báo, ví dụ, được dự báo từ (các) bảng màu dùng để mã hóa (các) khối trước đó, và các mục nhập mới có thể đặc trưng cho khối hiện thời và được báo hiệu rõ ràng. Bộ mã hóa và bộ giải mã có thể biết số lượng các mục nhập bảng màu được dự báo và các mục nhập bảng màu mới và tổng của chúng có thể chỉ báo tổng kích thước bảng màu trong khối.

Như được đề xuất trong ví dụ của JCTVC-Q0094 được viện dẫn trên đây, mỗi mẫu trong khối được mã hóa với bảng màu có thể thuộc một trong ba chế độ, như được đề cập sau đây:

- Chế độ thoát. Trong chế độ này, giá trị mẫu không được bao gồm trong bảng màu dưới dạng mục nhập bảng màu và giá trị mẫu được lượng tử hóa được báo hiệu rõ ràng đối với tất cả các thành phần màu sắc. Tương tự với việc báo hiệu các mục nhập bảng màu mới, mặc dù là dành cho các mục nhập bảng màu mới nhưng các giá trị thành phần màu sắc không được lượng tử hóa.
- Chế độ CopyAbove (còn gọi là chế độ CopyFromTop). Trong chế độ này, chỉ số mục nhập bảng màu dành cho mẫu hiện thời được sao chép từ mẫu nằm ngay

trên mẫu hiện thời trong khối mẫu. Trong các ví dụ khác, đối với chế độ copyabove, khối dữ liệu video có thể được chuyển vị do đó mẫu ở trên khối thực ra là mẫu ở bên trái khối.

- Chế độ giá trị (còn gọi là chế độ chỉ số). Trong chế độ này, giá trị của chỉ số mục nhập bảng màu được báo hiệu rõ ràng.

Như được mô tả ở đây, chỉ số mục nhập bảng màu có thể được gọi là chỉ số bảng màu hoặc đơn giản là chỉ số. Các thuật ngữ này có thể được dùng thay đổi cho nhau để mô tả các kỹ thuật của sáng chế. Ngoài ra, như được mô tả chi tiết hơn dưới đây, chỉ số bảng màu có thể có một hoặc nhiều giá trị màu sắc hoặc cường độ liên quan. Ví dụ, chỉ số bảng màu có thể có một giá trị màu sắc hoặc cường độ kết hợp gắn với một thành phần màu hoặc cường độ của điểm ảnh (ví dụ, thành phần Đỏ của dữ liệu RGB, thành phần Y của dữ liệu YUV, hoặc tương tự). Theo ví dụ khác, chỉ số bảng màu có thể có nhiều giá trị màu sắc hoặc cường độ kết hợp. Trong một số trường hợp, kỹ thuật mã hóa dữ liệu video dựa trên bảng màu có thể được áp dụng để mã hóa dữ liệu video đen trắng. Theo đó, “giá trị màu” thường có thể dùng để chỉ thành phần màu hoặc không màu bất kỳ dùng để tạo ra giá trị điểm ảnh.

Giá trị loạt có thể biểu thị loạt các giá trị chỉ số bảng màu mà được mã hóa bằng cách sử dụng cùng một chế độ mã hóa theo bảng màu. Ví dụ, theo chế độ giá trị, bộ mã hóa dữ liệu video (ví dụ, bộ mã hóa dữ liệu video 20 hoặc bộ giải mã dữ liệu video 30) có thể mã hóa giá trị chỉ số và giá trị loạt mà chỉ báo số lượng mẫu liên tiếp theo sau trong trình tự quét có cùng một giá trị chỉ số và được mã hóa với chỉ số bảng màu. Theo chế độ CopyAbove, bộ mã hóa dữ liệu video có thể mã hóa chỉ báo biểu thị rằng giá trị chỉ số dành cho giá trị mẫu hiện thời giống với giá trị chỉ số của mẫu lân cận trên (ví dụ, mẫu có vị trí bên trên mẫu hiện đang được mã hóa trong khối) và giá trị loạt chỉ báo số lượng mẫu liên tiếp theo sau trong trình tự quét mà cũng sao chép giá trị chỉ số từ mẫu lân cận trên. Theo đó, trong các ví dụ ở trên, loạt các giá trị chỉ số bảng màu được dùng để chỉ loạt các giá trị bảng màu có cùng giá trị hoặc loạt các giá trị chỉ số được sao chép từ các mẫu lân cận trên.

Do đó, loạt có thể chỉ rõ, đối với chế độ cho trước, số lượng các mẫu theo sau trong cùng một chế độ. Trong một số trường hợp, việc báo hiệu giá trị chỉ số và giá trị loạt có thể giống với việc mã hóa độ dài loạt. Theo một ví dụ nhằm mục đích minh họa, chuỗi giá trị chỉ số bảng màu liên tiếp của khối chỉ số tương ứng với khối dữ liệu

video có thể là 0, 2, 2, 2, 2, 5. Mỗi giá trị chỉ số tương ứng với một mẫu trong khối dữ liệu video. Trong ví dụ này, bộ mã hóa dữ liệu video có thể mã hóa mẫu thứ hai (ví dụ, giá trị chỉ số bảng màu thứ nhất là “2”) bằng cách sử dụng chế độ giá trị. Sau khi mã hóa giá trị chỉ số 2, bộ mã hóa dữ liệu video có thể mã hóa loạt số 3, loạt này chỉ báo rằng ba mẫu sau cũng có cùng giá trị chỉ số bảng màu là 2. Theo cách tương tự, việc mã hóa giá trị loạt gồm bốn chỉ số bảng màu sau khi mã hóa một chỉ số bằng cách sử dụng chế độ CopyAbove có thể chỉ báo rằng tổng năm chỉ số bảng màu được sao chép từ các giá trị chỉ số bảng màu tương ứng trong hàng bên trên vị trí mẫu hiện đang được mã hóa.

Bằng cách sử dụng bảng màu, bộ mã hóa dữ liệu video 20 và/hoặc bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để mã hóa khối mẫu (ví dụ, khối dữ liệu video) thành khối chỉ số, trong đó khối chỉ số là khối gồm các giá trị chỉ số ánh xạ đến một hoặc nhiều mục nhập bảng màu, và, trong một số ví dụ, một hoặc nhiều giá trị điểm ảnh thoát. Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa entropy khối chỉ số để nén khối chỉ số này. Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã entropy khối chỉ số được mã hóa để tạo ra khối chỉ số mà bộ giải mã dữ liệu video 30 có thể tạo ra khối mẫu (ví dụ, khối dữ liệu video được mã hóa bởi bộ mã hóa 20) từ đó. Ví dụ, kỹ thuật mã hóa entropy dựa trên độ dài loạt có thể được dùng để nén và giải nén khối chỉ số. Trong một số ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để lần lượt mã hóa và giải mã entropy khối chỉ số bằng cách sử dụng kỹ thuật CABAC.

Để áp dụng kỹ thuật mã hóa CABAC vào thông tin (ví dụ, phân tử cú pháp, khối chỉ số chẳng hạn như các giá trị chỉ số của khối chỉ số, hoặc thông tin khác), bộ mã hóa dữ liệu video (ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30) có thể thực hiện quy trình nhị phân hóa trên thông tin. Nhị phân hóa là quy trình chuyển đổi thông tin thành dãy gồm một hoặc nhiều bit. Mỗi dãy gồm một hoặc nhiều bit có thể được gọi là “các bin”. Nhị phân hóa là quy trình không tổn hao và có thể bao gồm một kỹ thuật hoặc kết hợp của các kỹ thuật mã hóa sau đây: mã hóa độ dài cố định, mã hóa đơn phân, mã hóa đơn phân rút gọn, mã hóa Rice rút gọn, mã hóa Golomb, mã hóa Golomb hàm mũ, mã hóa Golomb-Rice, dạng bất kỳ của mã hóa Golomb, dạng bất kỳ của mã hóa Rice, và dạng bất kỳ của mã hóa entropy. Ví dụ, quy trình nhị phân hóa có thể bao gồm bước biểu diễn giá trị nguyên 5 dưới dạng

00000101 bằng cách sử dụng kỹ thuật độ dài cố định 8 bit hoặc dưới dạng 11110 bằng cách sử dụng kỹ thuật mã hóa đơn phân.

Sau khi nhị phân hóa, bộ mã hóa dữ liệu video có thể nhận dạng ngữ cảnh mã hóa. Ngữ cảnh mã hóa có thể nhận dạng xác suất mã hóa các bin có các giá trị cụ thể. Ví dụ, ngữ cảnh mã hóa có thể chỉ báo xác suất là 0,7 để mã hóa bin có giá trị 0 và xác suất là 0,3 để mã hóa bin có giá trị 1. Sau khi nhận dạng ngữ cảnh mã hóa, bộ mã hóa dữ liệu video có thể mã hóa số học bin đó dựa trên ngữ cảnh, đây được gọi là mã hóa trong chế độ ngữ cảnh. Các bin được mã hóa bằng cách sử dụng kỹ thuật mã hóa trong chế độ ngữ cảnh CABAC có thể được gọi là “các bin ngữ cảnh”.

Hơn nữa, ngoài việc thực hiện mã hóa trong chế độ ngữ cảnh trên tất cả các bin, bộ mã hóa dữ liệu video (ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30) có thể mã hóa một số bin bằng cách sử dụng quy trình mã hóa CABAC trong chế độ bỏ qua (ví dụ, mã hóa trong chế độ bỏ qua). Mã hóa trong chế độ bỏ qua là quy trình mã hóa số học một bin không sử dụng ngữ cảnh thích nghi (ví dụ, ngữ cảnh mã hóa). Tức là, cơ cấu mã hóa trong chế độ bỏ qua không lựa chọn ngữ cảnh và có thể giả định xác suất là 0,5 cho cả hai ký hiệu (0 và 1). Mặc dù quy trình mã hóa trong chế độ bỏ qua có thể không hiệu quả về mặt băng thông như quy trình mã hóa trong chế độ ngữ cảnh, nhưng có thể ít tốn kém hơn về mặt tính toán khi thực hiện mã hóa trong chế độ bỏ qua trên một bin thay vì thực hiện mã hóa trong chế độ ngữ cảnh trên bin đó. Hơn nữa, việc thực hiện mã hóa trong chế độ bỏ qua có thể cho phép mức song song hóa và công suất cao hơn. Các bin được mã hóa bằng cách sử dụng kỹ thuật mã hóa trong chế độ bỏ qua có thể được gọi là “bin bỏ qua”.

Bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình với bộ mã hóa CABAC (ví dụ, bộ mã hóa CABAC và bộ giải mã CABAC). Bộ mã hóa CABAC có thể bao gồm cơ cấu mã hóa trong chế độ ngữ cảnh để thực hiện mã hóa trong chế độ ngữ cảnh CABAC và cơ cấu mã hóa trong chế độ bỏ qua để thực hiện mã hóa trong chế độ bỏ qua. Nếu bin được mã hóa trong chế độ ngữ cảnh, thì cơ cấu mã hóa trong chế độ ngữ cảnh được dùng để mã hóa bin này. Cơ cấu mã hóa trong chế độ ngữ cảnh có thể phải cần nhiều hơn hai vòng xử lý để mã hóa một bin. Tuy nhiên, với thiết kế đường ống thích hợp, cơ cấu mã hóa trong chế độ ngữ cảnh có thể chỉ cần $n+M$ vòng để mã hóa n bin, trong đó M là thông tin chi phí để bắt đầu đường ống. M thường lớn hơn 0.

Khi bắt đầu quy trình mã hóa CABAC (tức là, mỗi lần chuyển từ chế độ bỏ qua sang chế độ ngữ cảnh và ngược lại), thông tin chi phí đường ống được đưa vào. Nếu bin được mã hóa trong chế độ bỏ qua, thì cơ cấu mã hóa trong chế độ bỏ qua được dùng để mã hóa bin này. Cơ cấu mã hóa trong chế độ bỏ qua có thể được mong đợi là chỉ cần duy nhất một vòng để mã hóa thông tin n bit, trong đó n có thể lớn hơn một. Do đó, tổng số vòng để mã hóa tập hợp các bin bỏ qua và bin ngữ cảnh có thể được giảm bớt nếu tất cả các bin bỏ qua trong tập hợp được mã hóa cùng nhau và tất cả các bin ngữ cảnh trong tập hợp được mã hóa cùng nhau. Cụ thể, việc mã hóa các bin bỏ qua cùng nhau trước hoặc sau khi chuyển sang mã hóa trong chế độ ngữ cảnh có thể tiết kiệm chi phí cần thiết để khởi động lại cơ cấu mã hóa trong chế độ ngữ cảnh. Ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để chuyển giữa chế độ bỏ qua và chế độ ngữ cảnh trong khi lần lượt mã hóa hoặc giải mã khối dữ liệu video bằng cách sử dụng chế độ bảng màu. Theo ví dụ khác, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giảm số lần mà quy trình mã hóa hoặc giải mã chuyển giữa chế độ bỏ qua và chế độ ngữ cảnh khi mã hóa hoặc giải mã khối dữ liệu video bằng cách sử dụng chế độ bảng màu.

Các kỹ thuật được mô tả trong bản mô tả này có thể bao gồm các kỹ thuật dùng cho các tổ hợp khác nhau của một hoặc nhiều hoạt động như báo hiệu các chế độ mã hóa dữ liệu video dựa trên bảng màu, truyền bảng màu, suy ra bảng màu, báo hiệu trình tự quét, suy ra trình tự quét, và truyền các giá trị ánh xạ mã hóa dữ liệu video dựa trên bảng màu và các phần tử cú pháp khác. Ví dụ, các kỹ thuật trong bản mô tả này có thể hướng đến mã hóa entropy thông tin bảng màu. Trong một số ví dụ, các kỹ thuật trong bản mô tả này, trong số các kỹ thuật khác, được dùng để tăng hiệu quả mã hóa và giảm sự mã hóa không hiệu quả liên quan đến kỹ thuật mã hóa dữ liệu video dựa trên bảng màu. Do đó, như được mô tả chi tiết hơn dưới đây, các kỹ thuật trong bản mô tả này, trong một số trường hợp, cải thiện hiệu quả và cải thiện tốc độ bit khi mã hóa dữ liệu video bằng cách sử dụng chế độ bảng màu.

Như được mô tả trên đây, trong thiết kế chế độ bảng màu hiện thời trong mã hóa nội dung màn hình, các phần tử cú pháp **alette_index_idc** và **palette_escape_val** được mã hóa CABAC trong chế độ bỏ qua, và được đan xen với các phần tử cú pháp khác (ví dụ, **palette_run_msb_id_plus1**) mà được mã hóa CABAC trong chế độ ngữ

cảnh. Tuy nhiên, có thể có lợi nếu nhóm thông tin được mã hóa trong chế độ bỏ qua (ví dụ, các phần tử cú pháp) với nhau, điều này có thể cải thiện hiệu quả mã hóa và/hoặc giảm độ phức tạp mã hóa - giải mã.

Phần tử cú pháp **palette_index_idc** có thể là chỉ báo của chỉ số đối với mảng được biểu diễn bởi các mục nhập `currentPaletteEntries`, như được quy định trong, ví dụ, JCTVC-S1005. Giá trị **palette_index_idc** có thể nằm trong khoảng từ 0 đến $(adjustedIndexMax-1)$, bao gồm cả các giá trị đầu mút. Phần tử cú pháp **palette_escape_val** có thể chỉ rõ giá trị mẫu thoát mã hóa được lượng tử hóa cho một thành phần, như được xác định trong, ví dụ, JCTVC-S1005. **palette_run_msb_id_plus1** trừ 1 có thể chỉ rõ chỉ số của bit quan trọng nhất trong dạng biểu diễn nhị phân của `paletteRun`, như được xác định trong, ví dụ, JCTVC-S1005. Biến `paletteRun` này có thể chỉ rõ số vị trí liên tiếp trừ 1 có chỉ số bảng màu giống với vị trí của hàng trên khi `palette_run_type_flag` bằng `COPY_ABOVE_MODE` hoặc chỉ rõ số vị trí liên tiếp trừ 1 có cùng chỉ số bảng màu khi `palette_run_type_flag` bằng `COPY_INDEX_MODE`, như được xác định trong, ví dụ, JCTVC-S1005. Phần mô tả chi tiết bổ sung liên quan đến **palette_index_idc**, **palette_escape_val**, **palette_run_msb_id_plus1**, `currentPaletteEntries`, `adjustedIndexMax`, và `paletteRun` có thể được tìm thấy trong JCTVC-S1005.

Trong một số ví dụ, sáng chế mô tả phương pháp nhóm tất cả các phần tử cú pháp **palette_index_idc** ở trước đoạn mã hóa khối chỉ số bảng màu để cải thiện công suất CABAC. Ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa tất cả các phần tử cú pháp **palette_index_idc** ở trước đoạn mã hóa khối chỉ số bảng màu. Ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa tất cả các phần tử cú pháp **palette_index_idc** trước khi mã hóa các phần tử cú pháp cần được mã hóa trong chế độ ngữ cảnh. Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã tất cả các phần tử cú pháp **palette_index_idc** ở trước đoạn mã hóa khối chỉ số bảng màu. Ví dụ, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã tất cả các phần tử cú pháp **palette_index_idc** trước khi giải mã các phần tử cú pháp được mã hóa trong chế độ ngữ cảnh.

Theo ví dụ khác, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa trong chế độ bỏ qua tất cả các phần tử cú pháp **palette_index_idc** ở trước đoạn mã hóa khối chỉ số bảng màu sao cho tất cả các phần tử cú pháp **palette_index_idc** được

mã hóa trước khi mã hóa (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`). Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã tất cả các phần tử cú pháp **palette_index_idc** cho khối ở trước đoạn mã hóa khối chỉ số bảng màu của khối này sao cho tất cả các phần tử cú pháp **palette_index_idc** được giải mã trước khi giải mã (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`).

(Các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`).

Theo ví dụ khác, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa tất cả các phần tử cú pháp **palette_index_idc** trước khi mã hóa ngữ cảnh (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`). Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã tất cả các phần tử cú pháp **palette_index_idc** trước khi giải mã ngữ cảnh (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`).

Theo ví dụ khác, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa tất cả các phần tử cú pháp **palette_index_idc** trong đoạn mã hóa khối bảng màu trước khi mã hóa các phần tử cú pháp cần được mã hóa trong chế độ ngữ cảnh. Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã tất cả các phần tử cú pháp **palette_index_idc** trong đoạn mã hóa khối bảng màu trước khi giải mã các phần tử cú pháp được mã hóa trong chế độ ngữ cảnh. Theo ví dụ khác, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa tất cả các phần tử cú pháp **palette_index_idc** trong đoạn mã hóa khối bảng màu trước khi mã hóa ngữ cảnh (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`). Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã tất cả các phần tử cú pháp **palette_index_idc** trong đoạn mã hóa khối bảng màu trước khi giải mã ngữ cảnh (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế

độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`).

Nói chung, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để không đan xen mã hóa hoặc giải mã **palette_index_idc** trong chế độ bỏ qua với các phần tử cú pháp mà lần lượt cần được mã hóa hoặc giải mã bằng cách sử dụng chế độ ngữ cảnh. Ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để không đan xen mã hóa hoặc giải mã **palette_index_idc** trong chế độ bỏ qua với (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`) mà lần lượt cần được mã hóa hoặc giải mã bằng cách sử dụng chế độ ngữ cảnh. Theo ví dụ khác, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa bỏ qua tất cả các thể hiện của phần tử cú pháp **palette_index_idc** trước khi mã hóa ngữ cảnh phần tử cú pháp cần chế độ ngữ cảnh. Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã bỏ qua tất cả các thể hiện của phần tử cú pháp **palette_index_idc** trước khi giải mã ngữ cảnh phần tử cú pháp cần chế độ ngữ cảnh. Theo ví dụ khác, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa bỏ qua tất cả các thể hiện của phần tử cú pháp **palette_index_idc** trước khi mã hóa ngữ cảnh (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`). Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã bỏ qua tất cả các thể hiện của phần tử cú pháp **palette_index_idc** trước khi giải mã ngữ cảnh (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`).

Bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 cũng có thể lần lượt mã hóa và giải mã giá trị biểu diễn số lần xuất hiện của phần tử cú pháp **palette_index_idc**. Bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể sử dụng giá trị biểu diễn số lần xuất hiện của phần tử cú pháp **palette_index_idc** để lần lượt mã hóa hoặc giải mã mỗi trong số các phần tử cú pháp **palette_index_idc**. Các kỹ thuật được mô tả trong bản mô tả này cũng có thể loại bỏ phần thừa của các phần tử cú pháp liên quan đến độ dài loạt của bảng màu, và loại bỏ phần thừa của phần tử cú pháp **palette_run_type_flag** và **palette_index_idc**.

Trong một số ví dụ, sáng chế mô tả phương pháp nhóm tất cả các phần tử cú

pháp **palette_escape_val** ở trước đoạn mã hóa khối chỉ số bảng màu của khối (ví dụ, PU hoặc CU) để cải thiện công suất mã hóa CABAC. Ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa tất cả các phần tử cú pháp **palette_escape_val** ở trước đoạn mã hóa khối chỉ số bảng màu của khối. Ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa trong chế độ bỏ qua tất cả các phần tử cú pháp **palette_escape_val** ở trước đoạn mã hóa khối chỉ số bảng màu sao cho tất cả các phần tử cú pháp **palette_escape_val** được mã hóa trước khi mã hóa (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`). Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã tất cả các phần tử cú pháp **palette_escape_val** cho khối ở trước đoạn mã hóa khối chỉ số bảng màu của khối này sao cho tất cả các phần tử cú pháp **palette_escape_val** được giải mã trước khi giải mã (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`). Theo ví dụ khác, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa tất cả các phần tử cú pháp **palette_escape_val** trước khi mã hóa các phần tử cú pháp cần được mã hóa trong chế độ ngữ cảnh. Ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa tất cả các phần tử cú pháp **palette_escape_val** trước khi mã hóa ngữ cảnh (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, `palette_run_msb_id_plus1`). Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã tất cả các phần tử cú pháp **palette_escape_val** cho khối ở trước đoạn mã hóa khối chỉ số bảng màu của khối này. Ví dụ, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã tất cả các phần tử cú pháp **palette_escape_val** trước khi giải mã các phần tử cú pháp được mã hóa trong chế độ ngữ cảnh trong khối.

Theo ví dụ khác, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa tất cả các phần tử cú pháp **palette_escape_val** trong đoạn mã hóa khối bảng màu của khối trước khi mã hóa các phần tử cú pháp cần được mã hóa trong chế độ ngữ cảnh. Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã tất cả các phần tử cú pháp **palette_escape_val** trong đoạn mã hóa khối bảng màu của khối trước khi giải mã các phần tử cú pháp được mã hóa trong chế độ ngữ cảnh của khối này.

Nói chung, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để không đan xen mã hóa hoặc giải mã phần tử cú pháp **palette_escape_val** cho khối (ví dụ, PU hoặc CU) trong chế độ bỏ qua với các phần tử cú pháp mà lần lượt cần được mã hóa hoặc giải mã bằng cách sử dụng chế độ ngữ cảnh cho khối này. Ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để không đan xen mã hóa hoặc giải mã phần tử cú pháp **palette_escape_val** trong chế độ bỏ qua với (các) phần tử cú pháp liên quan đến kiểu loạt bảng màu (ví dụ, chế độ CopyAbove hoặc chế độ chỉ số) và/hoặc độ dài loạt (ví dụ, **palette_run_msb_id_plus1**) mà lần lượt cần được mã hóa hoặc giải mã bằng cách sử dụng chế độ ngữ cảnh. Theo ví dụ khác, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa bỏ qua tất cả các thể hiện của phần tử cú pháp **palette_escape_val** cho khối trước khi mã hóa ngữ cảnh phần tử cú pháp cần chế độ ngữ cảnh. Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã bỏ qua tất cả các thể hiện của phần tử cú pháp **palette_escape_val** của khối (ví dụ, PU hoặc CU) trước khi giải mã ngữ cảnh phần tử cú pháp cần chế độ ngữ cảnh của khối này.

Bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 cũng có thể lần lượt mã hóa và giải mã giá trị biểu diễn số lần xuất hiện của phần tử cú pháp **palette_escape_val** cho khối. Bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể sử dụng giá trị biểu diễn số lần xuất hiện của phần tử cú pháp **palette_escape_val** để lần lượt mã hóa hoặc giải mã mỗi trong số các phần tử cú pháp **palette_escape_val** cho khối này. Các kỹ thuật được mô tả trong bản mô tả này có thể giảm tải động của phần tử cú pháp **palette_index_idc** cho khối, điều này có thể làm cho hiệu quả mã hóa được cải thiện.

Các kỹ thuật, khía cạnh, và/hoặc ví dụ được mô tả ở đây có thể được sử dụng kết hợp với nhau theo tổ hợp bất kỳ hoặc tách rời nhau. Ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tổ hợp thích hợp bất kỳ của một hoặc nhiều kỹ thuật, khía cạnh và/hoặc ví dụ được mô tả ở đây.

Trong một số ví dụ, để cải thiện công suất mã hóa CABAC, bộ mã hóa dữ liệu video (ví dụ, bộ mã hóa dữ liệu video 20) có thể được tạo cấu hình để nhóm tất cả các lần xuất hiện của phần tử cú pháp **palette_index_idc** như được mô tả ở trên. Ví dụ, bộ mã hóa dữ liệu video (ví dụ, bộ mã hóa dữ liệu video 20) có thể được tạo cấu hình để

nhóm tất cả các lần xuất hiện của phần tử cú pháp **palette_index_idc** trong khối hiện thời (ví dụ, PU hoặc CU) ở trước đoạn mã hóa chỉ số cho khối hiện thời. Tương tự, bộ giải mã dữ liệu video (ví dụ, bộ giải mã dữ liệu video 30) có thể được tạo cấu hình để giải mã tất cả các phần tử cú pháp **palette_index_idc** như được mô tả ở trên. Fig.7 minh họa một ví dụ trong đó bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để nhóm tất cả các lần xuất hiện của phần tử cú pháp **palette_index_idc** trong khối hiện thời (ví dụ, CU), ví dụ, ở trước khối mã hóa chỉ số liên quan đến tài liệu của R. Joshi và J. Xu, “High efficient video coding (HEVC) screen content coding: Draft 2”, JCTVC-S1005, Phần 7.3.3.8. Khía cạnh này của sáng chế được gọi là Khía cạnh 1. Cụ thể, Fig.7 minh họa ví dụ về bộ mã hóa dữ liệu video 20 dịch chuyển một thể hiện của phần tử cú pháp **palette_index_idc** đến trước khối mã hóa chỉ số (mà còn có thể được gọi là đoạn mã hóa khối bảng màu hoặc trước khối mã hóa chỉ số). Nhờ việc dịch chuyển thể hiện được minh họa của phần tử cú pháp **palette_index_idc**, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để cải thiện công suất mã hóa CABAC bằng cách mã hóa tất cả các thể hiện của phần tử cú pháp **palette_index_idc** bằng cách sử dụng chế độ bỏ qua và chuyển sang chế độ ngữ cảnh để mã hóa thông tin bảng màu xuất hiện sau khi tất cả các thể hiện của phần tử cú pháp **palette_index_idc** trong khối mã hóa chỉ số được mã hóa trong chế độ bỏ qua.

Theo bản mô tả JCTVC-S1005, một thể hiện của phần tử cú pháp **palette_index_idc** sẽ được mã hóa trong chế độ bỏ qua, sau đó một thể hiện của phần tử cú pháp liên quan đến kiểu loại bảng màu và một thể hiện của phần tử cú pháp **palette_run_msb_id_plus1** sẽ được mã hóa trong chế độ ngữ cảnh, và quy trình này sẽ lặp lại vòng lặp while ($\text{scanPos} < \text{nCbS} * \text{nCbS}$), nghĩa là bộ mã hóa dữ liệu video sẽ chuyển qua lại giữa mã hóa trong chế độ bỏ qua và mã hóa trong chế độ ngữ cảnh bởi vì các phần tử cú pháp cần được mã hóa sử dụng chế độ bỏ qua không được nhóm với nhau. Quy trình này được thể hiện trên Fig.7 với hình elip ngay dưới vòng lặp while “while($\text{scanPos} < \text{nCbS} * \text{nCbS}$)” (tức là, hình elip này không bao gồm thông tin thể hiện rằng phần tử cú pháp liên quan đến kiểu loại bảng màu được mã hóa bằng cách sử dụng chế độ ngữ cảnh), hộp bao quanh lệnh-if với hậu thức của phần tử cú pháp **palette_index_idc** bên dưới vòng lặp while “while($\text{scanPos} < \text{nCbS} * \text{nCbS}$)”, và mã giả tiếp theo. Tuy nhiên, như được mô tả ở trên, Fig.7 cũng thể hiện Khía cạnh 1 của sáng chế, đó là việc nhóm (cũng có thể được gọi là dịch chuyển) một hoặc nhiều

thể hiện của phần tử cú pháp **palette_index_idc** đến, ví dụ, trước khối mã hóa chỉ số. Nhờ dịch chuyển một hoặc nhiều phần tử cú pháp (ví dụ, hoặc thông tin bảng màu khác) cần được mã hóa bằng cách sử dụng chế độ bỏ qua, bộ mã hóa dữ liệu video (ví dụ, bộ mã hóa dữ liệu video 20) có thể tăng công suất mã hóa entropy bằng cách giảm số lần mà bộ mã hóa dữ liệu video hoặc bộ giải mã dữ liệu video phải chuyển giữa mã hóa trong chế độ bỏ qua và mã hóa trong chế độ ngữ cảnh. Tương tự, nhờ dịch chuyển một hoặc nhiều phần tử cú pháp theo cách như vậy, công suất của bộ giải mã dữ liệu video (ví dụ, bộ giải mã dữ liệu video 30) có thể tăng lên do số lần bộ giải mã dữ liệu video phải chuyển giữa giải mã trong chế độ bỏ qua và giải mã trong chế độ ngữ cảnh được giảm bớt. Trong một số ví dụ về các kỹ thuật được mô tả trong bản mô tả này, tất cả các thể hiện của phần tử cú pháp **palette_index_idc** sẽ được mã hóa trong chế độ bỏ qua trước khi một thể hiện của phần tử cú pháp **palette_run_msb_id_plus1** được mã hóa trong chế độ ngữ cảnh.

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để báo hiệu số lần xuất hiện (ví dụ, các thể hiện) của phần tử cú pháp **palette_index_idc** bằng cách sử dụng phần tử cú pháp có tên là, ví dụ, **num_palette_index**. Ví dụ, bộ mã hóa dữ liệu video 20 có thể báo hiệu giá trị cho phần tử cú pháp **num_palette_index** trong dòng bit, trong đó giá trị này biểu thị số lần xuất hiện của phần tử cú pháp **palette_index_idc**. Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để không báo hiệu giá trị chỉ số là phần tử cú pháp **palette_index_idc**. Trong các ví dụ này, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để suy ra giá trị chỉ số. Ví dụ, lần xuất hiện của phần tử cú pháp **palette_index_idc** có thể được gộp vào phần tử cú pháp **num_palette_index**, mà có thể bằng số lần mà kiểu loạt (ví dụ, COPY_INDEX_MODE) xuất hiện trong khối cụ thể. Thậm chí khi kiểu loạt (ví dụ, COPY_INDEX_MODE) được suy ra hoặc phần tử cú pháp **palette_index_idc** được suy ra, lần xuất hiện vẫn gộp vào phần tử cú pháp **num_palette_index**. Như được sử dụng ở đây, việc nhắc đến một số chỉ số được phân tích cú pháp, giải mã, hoặc các chỉ số còn lại cần giải mã, trong một số ví dụ, có thể chỉ số lượng COPY_INDEX_MODE bất kể việc liệu chế độ hay chỉ số có được suy ra hay không. Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định số lần xuất hiện (ví dụ, các thể hiện) của phần tử cú pháp **palette_index_idc**, ví dụ, bằng cách giải mã giá trị được mã hóa tương ứng với phần tử cú pháp **num_palette_index** từ dòng bit. Khía cạnh này của sáng chế được

gọi là Khía cạnh 2. Bộ mã hóa dữ liệu video 20 và/hoặc bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để thực hiện Khía cạnh 1 khi có Khía cạnh 2 hoặc không có Khía cạnh 2. Theo cú pháp, Khía cạnh 2 có thể, theo một số ví dụ, được xác định như sau:

<code>indices_idc_coding() {</code>	
<code> num_palette_index</code>	<code>ae(v)</code>
<code> for (i = 0; i < NB_SUBFR; i++)</code>	
<code> palette_index_idc</code>	<code>ae(v)</code>
<code>}</code>	

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để thực hiện (ví dụ, bằng cách kích hoạt) các Khía cạnh 1 và 2 chỉ khi biến `indexMax` lớn hơn 1. Khía cạnh này của sáng chế được gọi là Khía cạnh 3. Biến `indexMax` có thể chỉ rõ số lượng giá trị riêng mà chỉ số bảng màu được dùng cho đơn vị mã hóa hiện thời. Trong một số ví dụ, biến `indexMax` có thể chỉ số lượng của (kích thước bảng màu + `palette_escape_val_present_flag`).

Trong một số ví dụ, các Khía cạnh 1 và 2 có thể bị hủy kích hoạt khi: (a) không có điểm ảnh thoát (tức là, `palette_escape_val_present_flag` == 0) trong khối hiện thời và kích thước bảng màu nhỏ hơn 2; hoặc (b) có thể có ít nhất một điểm ảnh thoát (tức là, `palette_escape_val_present_flag` == 1) trong khối hiện thời và kích thước bảng màu bằng 0. Trong các ví dụ khác, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để thực hiện (ví dụ, bằng cách kích hoạt) các Khía cạnh 1 và 2 chỉ khi biến `indexMax` lớn hơn 2. Tương tự, trong các ví dụ mà biến `indexMax` bằng (kích thước bảng màu + `palette_escape_val_present_flag`), các Khía cạnh 1 và 2 có thể được kích hoạt (ví dụ, được thực hiện) khi `indexMax` lớn hơn 1. Ví dụ, nếu kích thước bảng màu bằng 0 và `palette_escape_val_present_flag` bằng 1, thì tất cả các điểm ảnh trong khối là các điểm ảnh thoát; và, do đó các chỉ số đều đã được biết. Theo ví dụ khác, nếu `palette_escape_val_present_flag` bằng 0 và kích thước bảng màu bằng 1, ngoài ra, mỗi điểm ảnh có chỉ số 0; và, do đó, việc báo hiệu các chỉ số có thể không cần thiết.

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để

thực hiện các Khía cạnh 1 và 2 sao cho lần xuất hiện cuối cùng (ví dụ, thể hiện) của phần tử cú pháp **palette_run_type_flag**[xC][yC] được báo hiệu bởi bộ mã hóa dữ liệu video 20 ở trước đoạn mã hóa khối chỉ số bảng màu. Khía cạnh này của sáng chế được gọi là Khía cạnh 4. Cụ thể, bảng cú pháp có thể được cập nhật bằng cách, theo một số ví dụ, bổ sung phần tử cú pháp mới **palette_last_run_type_flag** như sau:

indices_idc_coding() {	
num_palette_index	ae(v)
for (i = 0; i < num_palette_index; i++)	
palette_index_idc	ae(v)
palette_last_run_type_flag	ae(v)
}	

Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định lần xuất hiện cuối cùng (ví dụ, thể hiện) của phần tử cú pháp **palette_run_type_flag**[xC][yC], ví dụ, bằng cách giải mã phần tử cú pháp **palette_last_run_type_flag** mã hóa từ dòng bit. Phần tử cú pháp **palette_last_run_type_flag** có thể được mã hóa trong chế độ bỏ qua hoặc được mã hóa trong chế độ ngữ cảnh bằng, ví dụ, mã hóa CABAC. Trong các ví dụ mà phần tử cú pháp **palette_last_run_type_flag** được mã hóa trong chế độ ngữ cảnh, phần tử cú pháp **palette_last_run_type_flag** có thể dùng chung (các) ngữ cảnh giống nhau với phần tử cú pháp **palette_run_type_flag**[xC][yC], hoặc phần tử cú pháp **palette_last_run_type_flag** có thể có (các) ngữ cảnh riêng của nó mà độc lập với (các) ngữ cảnh của phần tử cú pháp **palette_run_type_flag**[xC][yC].

Trong một số ví dụ, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã phần tử cú pháp **palette_index_idc** sao cho quy trình điều chỉnh dải động bị vô hiệu hóa cho lần xuất hiện thứ nhất (ví dụ, thể hiện) của phần tử cú pháp **palette_index_idc**. Khía cạnh này của sáng chế được gọi là Khía cạnh 5. Cụ thể, sáng chế sử dụng quy trình rất giống thủ tục suy ra của biến **adjustedIndexMax** được quy định trong JCTVC-S1005 Mục 7.4.9.6. Nhằm mục đích so sánh, JCTVC-S1005 mô tả rằng biến **adjustedIndexMax** có thể được suy ra như sau:

adjustedIndexMax = **indexMax**

if(**scanPos** > 0)

$adjustedIndexMax - = 1$

Tuy nhiên, theo Khía cạnh 5 của sáng chế, biến `adjustedIndexMax` có thể được suy ra như được nêu dưới đây. Ví dụ, đối với mỗi khối, biến `isFirstIndex` được khởi tạo bằng 1 trước khi phân tích cú pháp. Trong một số ví dụ, biến `adjustedIndexMax` có thể được suy ra như sau:

```
adjustedIndexMax = indexMax
palette_index_idc
if( isFirstIndex ) {
    adjustedIndexMax - = isFirstIndex
    isFirstIndex = 0
}
```

Trong một số ví dụ, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để kiểm tra một hoặc nhiều điều kiện trước khi phân tích cú pháp và giải mã `paletteRun`. Khía cạnh này của sáng chế được gọi là Khía cạnh 6. Biến `paletteRun` có thể chỉ rõ số vị trí liên tiếp trừ 1 có chỉ số bảng màu giống với vị trí trong hàng trên khi `palette_run_type_flag` bằng `COPY_ABOVE_MODE` hoặc chỉ rõ số vị trí liên tiếp trừ 1 có chỉ số bảng màu giống nhau khi `palette_run_type_flag` bằng `COPY_INDEX_MODE`, ví dụ, như được mô tả trong JCTVC-S1005.

Liên quan đến một hoặc nhiều điều kiện mà bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để kiểm tra, nếu bộ giải mã dữ liệu video 30 xác định rằng một hoặc nhiều điều kiện được thỏa mãn, thì bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để bỏ qua quy trình phân tích cú pháp và giải mã đối với các phần tử cú pháp liên quan đến `paletteRun` hiện thời (tức là `palette_run_msb_id_plus1` và `palette_run_refinement_bits`). Trong ví dụ này, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để ngầm suy ra biến `paletteRun` hiện thời khi chạy đến cuối khối hiện thời, tức là, bằng biến `maxPaletteRun`. Danh sách một hoặc nhiều điều kiện liên quan đến Khía cạnh 6 bao gồm: (i) số phần tử cú pháp `palette_index_idc` được phân tích cú pháp/giải mã bằng `num_palette_index`; hoặc, theo cách khác, biến `paletteIndicesLeft` có thể được xác định bằng `num_palette_index` trừ đi số lượng chỉ số nhận được, và với định nghĩa như vậy, điều kiện này có thể được viết là `paletteIndicesLeft` bằng không; và/hoặc (ii) kiểu loạt bảng màu hiện thời `palette_run_type_flag[ xC ][ yC ]` bằng với kiểu loạt bảng màu cuối cùng

palette_last_run_type_flag.

Trong một số ví dụ, nếu các điều kiện (i) và (ii) nêu trên đối với Khía cạnh 6 không được thỏa mãn đồng thời, thì bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa độ dài loạt bảng màu trong dòng bit. Khía cạnh này của sáng chế được gọi là Khía cạnh 7. Trong các ví dụ khác, nếu các điều kiện (i) và (ii) nêu trên đối với Khía cạnh 6 không được thỏa mãn đồng thời, thì bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa độ dài loạt bảng màu trong dòng bit. Theo bản dự thảo hiện thời JCTVC-S1005, cần có tham số xác định độ dài loạt thu được lớn nhất là đầu vào, trong đó tham số này bằng $\text{maxPaletteRun} = \text{nCbS} * \text{nCbS} - \text{scanPos} - 1$. Tuy nhiên theo sáng chế, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để giảm tham số chỉ rõ độ dài loạt thu được lớn nhất về $\text{maxPaletteRun} = \text{nCbS} * \text{nCbS} - \text{scanPos} - 1 - \text{paletteIndicesLeft}$ để cải thiện hiệu quả mã hóa. Như được sử dụng ở đây, nCbS xác định kích thước của khối hiện thời.

Trong một số ví dụ, ràng buộc tiêu chuẩn có thể được áp dụng cho bộ mã hóa dữ liệu video 20 yêu cầu bộ mã hóa này không bao giờ báo hiệu bảng màu có các mục nhập không sử dụng nếu khối không có trong chế độ dùng chung bảng màu (tức là, **palette_share_flag**[x0][y0] = 0). Khía cạnh này của sáng chế được gọi là Khía cạnh 8.

Trong một số ví dụ, đối với chế độ bảng màu không sử dụng chế độ dùng chung bảng màu, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để bỏ qua việc giải mã lần xuất hiện hiện thời (ví dụ, thể hiện) của phần tử cú pháp **palette_index_idc** khi một hoặc nhiều điều kiện sau đây được thỏa mãn: điều kiện 1 trong đó **num_palette_index** bằng **indexMax**, và điều kiện 2 trong đó **paletteIndicesLeft** = 1. Trong các ví dụ này, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để ngầm suy ra giá trị cho lần xuất hiện hiện thời của phần tử cú pháp **palette_index_idc** dưới dạng chỉ số tức là trong bảng màu, nhưng vẫn chưa xuất hiện trong giá trị ánh xạ chỉ số trong quy trình giải mã (ví dụ, chưa xuất hiện trong giá trị ánh xạ chỉ số cho đến điểm này trong quy trình giải mã). Khía cạnh này của sáng chế được gọi là Khía cạnh 9.

Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để suy ra giá trị cho lần xuất hiện hiện thời của phần tử cú pháp **palette_index_idc** như nêu trên đối với Khía cạnh 9 bởi vì điều kiện 1 yêu cầu rằng mỗi chỉ số giữa 0 và (**indexMax** - 1), bao gồm cả các giá trị đầu mút, được báo hiệu và chỉ được báo hiệu một lần. Do đó, sau khi các

giá trị chỉ số ($\text{indexMax} - 1$) thứ nhất được báo hiệu, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để suy ra giá trị chỉ số cuối cùng là số giữa 0 và ($\text{indexMax} - 1$), số này vẫn chưa xuất hiện trong quy trình giải mã đối với ánh xạ chỉ số hiện thời.

Trong một số ví dụ, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để bỏ qua việc giải mã lần xuất hiện hiện thời (ví dụ, thể hiện) của phần tử cú pháp **palette_run_type_flag**[xC][yC] khi một hoặc cả hai điều kiện sau đây được thỏa mãn: điều kiện 1 trong đó **paletteIndicesLeft** bằng 0, và điều kiện 2 trong đó điểm ảnh hiện thời ở vị trí cuối cùng của khối theo trình tự quét. Trong các ví dụ này, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để ngầm suy ra giá trị cho lần xuất hiện hiện thời của phần tử cú pháp **palette_run_type_flag**[xC][yC]. Ví dụ, khi điều kiện 1 được thỏa mãn, phần tử cú pháp **palette_run_type_flag**[xC][yC] của bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để suy ra giá trị cho lần xuất hiện hiện thời của phần tử cú pháp **palette_run_type_flag**[xC][yC] là COPY_ABOVE_MODE. Theo ví dụ khác, khi điều kiện 1 được thỏa mãn, phần tử cú pháp **palette_run_type_flag**[xC][yC] của bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để suy ra giá trị cho lần xuất hiện hiện thời của phần tử cú pháp **palette_run_type_flag**[xC][yC] là COPY_INDEX_MODE nếu **paletteIndicesLeft** > 0, và là COPY_ABOVE_MODE nếu **paletteIndicesLeft** = 0. Khía cạnh này của sáng chế được gọi là Khía cạnh 10.

Như được mô tả ở đây, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định khi nào một điều kiện được thỏa mãn. Ví dụ, đối với Khía cạnh 10, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định xem điều kiện 1 có được thỏa mãn hay không. Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định xem điều kiện 2 có được thỏa mãn hay không. Đáp lại việc xác định rằng điều kiện 1 hoặc điều kiện 2 được thỏa mãn, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để suy ra giá trị cho lần xuất hiện hiện thời của phần tử cú pháp **palette_run_type_flag**[xC][yC] như được nêu trên đây.

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để lần lượt mã hóa hoặc giải mã phần tử cú pháp **num_palette_index** bằng cách sử dụng họ mã golomb bất kỳ. Ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để lần lượt mã hóa hoặc giải mã phần tử cú pháp **num_palette_index** bằng cách sử dụng, ví dụ, mã

Golomb Rice, mã Golomb hàm mũ, mã Rice rút gọn, mã đơn phân, hoặc dạng ghép nối của mã Golomb Rice và mã Golomb hàm mũ. Khía cạnh này của sáng chế được gọi là Khía cạnh 11.

Trong các ví dụ khác, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để lần lượt mã hóa hoặc giải mã phần tử cú pháp **num_palette_index** bằng cách sử dụng phiên bản rút gọn bất kỳ của họ mã golomb bất kỳ. Ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để lần lượt mã hóa hoặc giải mã phần tử cú pháp **num_palette_index** bằng cách sử dụng, ví dụ, mã Golomb Rice rút gọn, mã Golomb hàm mũ rút gọn, mã Rice rút gọn, mã đơn phân rút gọn, hoặc dạng ghép nối của mã Rice rút gọn và mã Golomb hàm mũ rút gọn như mã được dùng để mã hóa các phần tử cú pháp **coeff_abs_level_remaining**. Khía cạnh này của sáng chế được gọi là Khía cạnh 12.

Trong một số ví dụ, mọi tham số golomb liên quan đến Khía cạnh 11 hoặc 12 phụ thuộc vào kích thước CU, **indexMax**, kích thước bảng màu, và/hoặc phần tử cú pháp **palette_escape_val_present_flag**. Sự phụ thuộc này có thể được biểu diễn dưới dạng các phương trình hoặc bảng tra cứu. Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để báo hiệu bảng tra cứu hoặc các tham số trong các phương trình sao cho chúng được nhận bởi bộ giải mã dữ liệu video 30 trong, ví dụ, SPS/PPS/phần đầu lát. Ngoài ra hoặc theo cách khác, các tham số có thể được cập nhật thích ứng theo từng khối. Khía cạnh này của sáng chế được gọi là Khía cạnh 13. Trong một số ví dụ, tham số golomb **cRiceParam** có thể phụ thuộc vào **indexMax**, kích thước bảng màu, và/hoặc **palette_escape_val_present_flag**. Tham số golomb **cRiceParam** có thể thay đổi từ khối này sang khối khác.

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa dự báo **num_palette_index** bằng cách báo hiệu hiệu số giữa giá trị của **num_palette_index** và giá trị bù, mà có thể được biểu diễn bởi phần tử cú pháp có tên là, ví dụ, **numPaletteIndexCoded**. Khía cạnh này của sáng chế được gọi là Khía cạnh 14. Ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa dự báo **num_palette_index** bằng cách báo hiệu giá trị cho **numPaletteIndexCoded**, trong đó $\text{numPaletteIndexCoded} = \text{num_palette_index} - \text{IndexOffsetValue}$. Tương tự, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã dự báo **num_palette_index** bằng cách, ví dụ, xác định giá trị cho **numPaletteIndexCoded** từ dòng bit. Bởi vì

$\text{numPaletteIndexCoded} = \text{num_palette_index} - \text{IndexOffsetValue}$, nên bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định giá trị của **num_palette_index** dựa vào giá trị đã được xác định của $\text{numPaletteIndexCoded}$ và giá trị của IndexOffsetValue .

Trong một số ví dụ, biến IndexOffsetValue có thể là hằng số. Ví dụ, IndexOffsetValue có thể bằng giá trị hằng số X đối với chế độ dùng chung bảng màu hoặc có thể bằng giá trị hằng số Y đối với chế độ không dùng chung bảng màu, trong đó X và Y là các số nguyên. Trong một số ví dụ, X và Y có thể giống nhau (ví dụ, X bằng Y ví dụ như bằng 1). Trong các ví dụ khác, X và Y có thể khác nhau (ví dụ, X không bằng Y). Ví dụ, IndexOffsetValue có thể bằng 9 khi chế độ dùng chung bảng màu được sử dụng, và IndexOffsetValue có thể bằng 33 khi chế độ không dùng chung bảng màu được sử dụng. Trong một số ví dụ, biến IndexOffsetValue có thể phụ thuộc vào phần tử cú pháp **palette_share_flag**[x0][y0]. Trong các ví dụ khác, biến IndexOffsetValue có thể phụ thuộc vào biến indexMax . Ví dụ, biến IndexOffsetValue có thể bằng biến indexMax . Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để báo hiệu biến IndexOffsetValue trong SPS/PPS/phần đầu lát. Theo cách khác hoặc ngoài ra, biến IndexOffsetValue có thể được cập nhật thích ứng theo từng khối, nghĩa là giá trị tương ứng với biến IndexOffsetValue có thể được cập nhật thích ứng theo từng khối.

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để lần lượt mã hóa hoặc giải mã $\text{numPaletteIndexCoded}$ bằng cách sử dụng họ mã golomb bất kỳ hoặc họ golomb rút gọn bất kỳ, ví dụ như dạng ghép nối của mã Golomb Rice và mã Golomb hàm mũ. Ví dụ, khi IndexOffsetValue bằng 1, $\text{numPaletteIndexCoded}$ bằng **num_palette_index** - 1.

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để lần lượt mã hóa hoặc giải mã phần tử cú pháp $\text{numPaletteIndexCoded}$ bằng cách sử dụng họ mã golomb bất kỳ. Ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để lần lượt mã hóa hoặc giải mã phần tử cú pháp $\text{numPaletteIndexCoded}$ bằng cách sử dụng, ví dụ, mã Golomb Rice, mã Golomb hàm mũ, mã Rice rút gọn, mã đơn phân, hoặc dạng ghép nối của mã Golomb Rice và mã Golomb hàm mũ.

Trong các ví dụ khác, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video

30 có thể được tạo cấu hình để lần lượt mã hóa hoặc giải mã `numPaletteIndexCoded` bằng cách sử dụng phiên bản rút gọn bất kỳ của họ mã golomb bất kỳ. Ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để lần lượt mã hóa hoặc giải mã phần tử cú pháp `numPaletteIndexCoded` bằng cách sử dụng, ví dụ, mã Golomb Rice rút gọn, mã Golomb hàm mũ rút gọn, mã Rice rút gọn, mã đơn phân rút gọn, hoặc dạng ghép nối của mã Rice rút gọn và mã Golomb hàm mũ rút gọn ví dụ như mã được dùng để mã hóa các phần tử cú pháp `coeff_abs_level_remaining`.

Để mã hóa `numPaletteIndexCoded`, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để xác định ký hiệu của phần tử cú pháp `numPaletteIndexCoded`. Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để báo hiệu cờ biểu thị liệu giá trị của phần tử cú pháp `numPaletteIndexCoded` có là giá trị âm hay không (ví dụ, liệu ký hiệu xác định là giá trị dương hay âm). Khía cạnh này của sáng chế được gọi là Khía cạnh 15. Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để báo hiệu cờ, và sau đó báo hiệu giá trị của phần tử cú pháp `numPaletteIndexCoded`. Trong các ví dụ khác, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để báo hiệu giá trị của phần tử cú pháp `numPaletteIndexCoded`, và sau đó báo hiệu cờ. Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa cờ bằng cách sử dụng chế độ bỏ qua hoặc chế độ ngưỡng cảnh. Nếu mã hóa trong chế độ ngưỡng cảnh, các ngưỡng cảnh có thể phụ thuộc vào kích thước CU, `indexMax`, kích thước bảng màu, và/hoặc **`palette_escape_val_present_flag`**.

Như được mô tả ở trên, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để xác định ký hiệu của phần tử cú pháp `numPaletteIndexCoded` theo một số ví dụ. Nếu ký hiệu xác định được của phần tử cú pháp `numPaletteIndexCoded` là âm, thì bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa giá trị của phần tử cú pháp ($1 - \text{numPaletteIndexCoded}$) trong dòng bit. Nếu ký hiệu xác định được của phần tử cú pháp `numPaletteIndexCoded` là dương, thì bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa giá trị của phần tử cú pháp `numPaletteIndexCoded` trong dòng bit. Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa giá trị của phần tử cú pháp ($1 - \text{numPaletteIndexCoded}$) hoặc giá trị của phần tử cú pháp `numPaletteIndexCoded` bằng cách sử dụng các tham số mã golomb khác nhau phụ thuộc vào, ví dụ, ký hiệu của phần tử cú pháp `numPaletteIndexCoded`, kích thước CU, `indexMax`, kích thước bảng màu, và/hoặc **`palette_escape_val_present_flag`**.

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để biểu diễn các phần âm của numPaletteIndexCoded bằng cách sử dụng phép toán ánh xạ, đây có thể là cách bổ sung hoặc thay thế cho Khía cạnh 15. Khía cạnh này của sáng chế được gọi là Khía cạnh 16. Ví dụ, khoảng ánh xạ có thể được đưa vào và được định nghĩa là biến mapInterval. Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để, bằng cách sử dụng biến mapInterval, ánh xạ các giá trị âm của numPaletteIndexCoded lên các giá trị dương có khoảng cách bằng nhau bằng: $\text{mapInterval} \times (-\text{numPaletteIndexCoded}) - 1$. Giá trị dương tương ứng của numPaletteIndexCoded có thể được dịch chuyển để phù hợp với các vị trí chứa các giá trị âm được ánh xạ.

Ví dụ, nếu $\text{mapInterval} = 2$, và numPaletteIndexCoded được chọn từ $\{-3, -2, -1, 0, 1, 2, 3\}$, thì việc ánh xạ có thể được minh họa như trong Bảng I dưới đây. Trong ví dụ này, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa các giá trị của numPaletteIndexCode bằng cách sử dụng các giá trị được ánh xạ trong Bảng I. Ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa entropy các giá trị ánh xạ thành dạng nhị phân.

Bảng I. Ví dụ ánh xạ từ mã

numPaletteIndexCoded	giá trị được ánh xạ
-3	5
-2	3
-1	1
0	0
1	2
2	4
3	6

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để biểu diễn các phần âm của numPaletteIndexCoded bằng cách sử dụng phép toán ánh xạ như được mô tả liên quan đến Khía cạnh 16. Bộ mã hóa dữ liệu video 20 cũng có thể được tạo cấu hình để loại bỏ một hoặc nhiều phần dư mà có thể có khi thực hiện Khía cạnh 16. Khía cạnh này của sáng chế được gọi là Khía cạnh 17. Ví dụ, số lượng giá trị âm của numPaletteIndexCoded có thể nằm trong khoảng từ $A = \{-1, -2, \dots, -$

$\text{IndexOffsetValue} + 1\}$. Theo ví dụ khác, số lượng giá trị âm của $\text{numPaletteIndexCode}$ có thể nằm trong khoảng từ $A=\{-1, -2, \dots, -\text{IndexOffsetValue} + 1, \text{IndexOffsetValue}\}$. Trong các ví dụ này, giá trị được ánh xạ chỉ cần dành các vị trí $(\text{IndexOffsetValue} - 1)$ hoặc IndexOffsetValue cho các giá trị $\text{numPaletteIndexCoded}$ âm. Ví dụ, nếu $\text{mapInterval} = 2$, và $\text{numPaletteIndexCoded}$ được chọn từ $\{-3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8\}$, thì việc ánh xạ được minh họa trong Bảng II dưới đây. Trong ví dụ này, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa các giá trị của $\text{numPaletteIndexCode}$ bằng cách sử dụng các giá trị được ánh xạ trong Bảng II. Ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa entropy các giá trị được ánh xạ thành dạng nhị phân.

Bảng II. Ví dụ ánh xạ từ mã

$\text{numPaletteIndexCoded}$	giá trị được ánh xạ
-3	5
-2	3
-1	1
0	0
1	2
2	4
3	6
4	7
5	8
6	9
7	10
8	11

Như thể hiện trong Bảng II trên đây, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa các giá trị ánh xạ tương ứng với các giá trị của $\text{numPaletteIndexCode}$ sao cho các giá trị âm và các giá trị dương của $\text{numPaletteIndexCode}$ không đan xen sau một giá trị nhất định. Ví dụ, theo ví dụ trong Bảng II trên đây, không có sự đan xen các giá trị dương và các giá trị âm của $\text{numPaletteIndexCoded}$ qua các giá trị ánh xạ bắt đầu với giá trị 3 của

numPaletteIndexCoded (tức là, các giá trị dương từ 3 đến 8 của numPaletteIndexCoded ánh xạ đến các giá trị được ánh xạ từ 6 đến 11).

Như được mô tả trên đây, bộ mã hóa dữ liệu video 20 cũng có thể được tạo cấu hình để loại bỏ một hoặc nhiều phần dư mà có thể có khi thực hiện Khía cạnh 16. Ví dụ về phần dư khác với ví dụ phần dư được mô tả trên đây bao gồm: Vì **num_palette_index** bị chặn trên bởi tổng số điểm ảnh trong khối hiện thời, cho nên numPaletteIndexCoded cũng bị chặn trên. Do đó, sau khi phân bổ các vị trí cho tất cả các khả năng của từ mã dương, các giá trị âm có thể được ánh xạ đến các vị trí sau đây mà không bị đan xen. Ví dụ, nếu mapInterval = 2, và numPaletteIndexCoded được chọn từ {-5, -4, -3, -2, -1, 0, 1, 2, 3}, ánh xạ được minh họa trong Bảng III dưới đây. Trong ví dụ này, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa các giá trị của numPaletteIndexCode bằng cách sử dụng các giá trị ánh xạ trong Bảng III. Ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa entropy các giá trị được ánh xạ thành dạng nhị phân.

Bảng III. Ví dụ ánh xạ từ mã

numPaletteIndexCoded	giá trị được ánh xạ
-5	8
-4	7
-3	5
-2	3
-1	1
0	0
1	2
2	4
3	6

Như được thể hiện trên Bảng III trên đây, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa các giá trị ánh xạ tương ứng với các giá trị của numPaletteIndexCode sao cho các giá trị âm và các giá trị dương của numPaletteIndexCode không bị đan xen sau một giá trị nhất định. Ví dụ, trong ví dụ trên Bảng III trên đây, không có sự đan xen các giá trị dương và các giá trị âm của

numPaletteIndexCoded qua các giá trị ánh xạ bắt đầu với giá trị 4 của numPaletteIndexCoded (tức là, các giá trị âm -4 và -5 của numPaletteIndexCoded ánh xạ đến các giá trị được ánh xạ 7 và 8).

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để tách riêng mối quan hệ giữa chỉ số bảng màu và giá trị loạt bảng màu. Khía cạnh này của sáng chế được gọi là Khía cạnh 18. Ví dụ, thay vì chấp nhận các ngữ cảnh mã hóa loạt bảng màu phụ thuộc vào các chỉ số đã phân tích cú pháp và giải mã, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để tạo ra các ngữ cảnh mã hóa loạt bảng màu phụ thuộc vào độ dài loạt bảng màu trước đó hoặc phụ thuộc vào **palette_run_msb_id_plus1** của loạt trước đó, **indexMax**, và/hoặc kích thước CU.

Trong một số ví dụ, để nhóm thêm các bin bỏ qua, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để báo hiệu số lượng chỉ số thoát trong khối bảng màu cũng như các giá trị thoát trước khi báo hiệu kiểu loạt bảng màu (tức là **palette_run_type_flag[xC][yC]**) như sau. Khía cạnh này của sáng chế được gọi là Khía cạnh 19. Các phần in nghiêng minh họa các thay đổi so với (các) phiên bản trước đó của JCT-VC S1005, và các phần in đậm cũng như “ae(v)” trong cột phải biểu thị việc báo hiệu phân tử cú pháp.

...	
if(currentPaletteSize != 0)	
palette_escape_val_present_flag	ae(v)
<i>if(palette_escape_val_present_flag (indexMax > 0))</i>	
<i>escape_idc_coding()</i>	
if(palette_escape_val_present_flag) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_palette_abs	ae(v)
if(cu_qp_delta_palette_abs)	
cu_qp_delta_palette_sign_flag	ae(v)
}	
if(cu_chroma_qp_offset_enabled_flag &&	

!IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_palette_offset_flag	ae(v)
if(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	
cu_chroma_qp_palette_offset_idx	ae(v)
}	
}	
if(indexMax > 0)	
palette_transpose_flag	ae(v)
scanPos = 0	
while(scanPos < nCbS * nCbS) {	
...	

Trong ví dụ ở trên, `escape_idc_coding()` bao gồm bước báo hiệu số lượng chỉ số thoát và các giá trị thoát tương ứng với mỗi chỉ số thoát. Số lượng chỉ số thoát trong khối bảng màu có thể không được báo hiệu nếu **palette_escape_val_present_flag** bằng 0 hoặc nếu `indexMax` bằng 0. Trong trường hợp trước, số lượng chỉ số thoát được suy ra là bằng 0 và không có giá trị thoát nào được báo hiệu. Trong trường hợp sau với `indexMax` bằng 0, số lượng chỉ số thoát được suy ra bằng kích thước khối khi **palette_escape_val_present_flag** bằng 1 và các giá trị thoát được báo hiệu, hoặc số lượng chỉ số thoát được suy ra là bằng không khi **palette_escape_val_present_flag** bằng 0.

Trong một số ví dụ, bộ mã hóa dữ liệu video 2D có thể được tạo cấu hình để báo hiệu số lượng chỉ số thoát bằng cách sử dụng họ mã golomb. Khía cạnh này của sáng chế được gọi là Khía cạnh 20. Ví dụ, bộ mã hóa dữ liệu video 2D có thể được tạo cấu hình để báo hiệu số lượng chỉ số thoát bằng cách sử dụng, ví dụ, mã Golomb Rice, mã Golomb hàm mũ, mã Rice rút gọn, mã đơn phân, hoặc dạng ghép nối của mã Golomb Rice và mã Golomb hàm mũ. Các phiên bản rút gọn của các mã ở trên có thể được sử dụng với tập hợp tối đa bằng kích thước khối.

Trong một số ví dụ, sáng chế đề xuất áp dụng giá trị hạn chế tiêu chuẩn đối với

phần tử cú pháp **palette_escape_val_present_flag** mà khi **palette_escape_val_present_flag** bằng 0, không có điểm ảnh thoát trong khối hiện thời. Khía cạnh này của sáng chế được gọi là Khía cạnh 21. Khi **palette_escape_val_present_flag** bằng 1, có *ít nhất một* điểm ảnh thoát trong khối hiện thời. Với hạn chế này, trong `escape_idc_coding()`, số lượng chỉ số thoát trừ đi 1 có thể được mã hóa thay cho số lượng chỉ số thoát để cải thiện hiệu suất mã hóa. Trong trường hợp đó, giá trị tối đa của họ mã golomb rút gọn theo đó có thể được điều chỉnh bằng $(\text{blockSize}-1)$.

Trong một số ví dụ, khi số lượng chỉ số thoát được báo hiệu trước khi mã hóa khối ảnh xạ chỉ số và khi tất cả các chỉ số thoát đã được mã hóa, thì `indexMax` có thể được giảm đi 1. Hơn nữa, nếu `indexMax` bằng 1, thì quy trình mã hóa chỉ số, mã hóa loạt và mã hóa chế độ kết thúc do các chỉ số cho tất cả các ví dụ còn lại có thể được suy ra. Khía cạnh này của sáng chế được gọi là Khía cạnh 22. Trong một ví dụ của Khía cạnh 22, giả định kích thước bảng màu bằng 1 và **palette_escape_val_present_flag** bằng 1. Thông thường, các giá trị chỉ số có thể có bằng 0 và 1, trong đó 1 được sử dụng cho (các) mẫu thoát. Theo Khía cạnh 22, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để báo hiệu số lượng giá trị/mẫu thoát. Sau đó, khi các chỉ số được báo hiệu và giá trị /mẫu thoát cuối cùng được bắt gặp, cả bộ mã hóa dữ liệu video 20 và/hoặc bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để suy ra (ví dụ, xác định) rằng không có thêm các giá trị/mẫu thoát. Do đó, bộ mã hóa dữ liệu video 20 và/hoặc bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định rằng chỉ giá trị chỉ số mà có thể xuất hiện từ giá trị/mẫu thoát cuối cùng đến phần cuối của khối bằng 0, nghĩa là bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để không báo hiệu chế độ, giá trị chỉ số, và/hoặc giá trị loạt từ giá trị/mẫu thoát cuối cùng đến phần cuối của khối.

Trong một số ví dụ, `escape_idc_coding()` được sử dụng kết hợp với `indices_idc_coding()`. Khía cạnh này của sáng chế được gọi là Khía cạnh 23. Trong một ví dụ, số lượng chỉ số thoát có thể được báo hiệu trước khi báo hiệu số lượng chỉ số. Trong trường hợp này, chỉ các chỉ số không thoát cần được báo hiệu trong `indices_idc_coding()`. Trong một ví dụ, số lượng chỉ số thoát có thể được báo hiệu sau khi báo hiệu số lượng chỉ số. Trong trường hợp này, giá trị tối đa của họ mã golomb rút gọn theo đó có thể được điều chỉnh bằng **num_palette_index**.

Bộ mã hóa dữ liệu video 20 và/hoặc bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để hoạt động theo các kỹ thuật được mô tả trong bản mô tả này. Nói chung, bộ mã hóa dữ liệu video 20 và/hoặc bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định rằng khối hiện thời được mã hóa trong chế độ bảng màu, mã hóa trong chế độ bỏ qua các thể hiện của phần tử cú pháp thứ nhất để tái tạo khối hiện thời, và sau khi mã hóa trong chế độ bỏ qua các thể hiện của phần tử cú pháp thứ nhất, giải mã trong chế độ ngữ cảnh các thể hiện của phần tử cú pháp thứ hai để tái tạo khối hiện thời.

Fig.2 là sơ đồ khối thể hiện bộ mã hóa dữ liệu video 20 làm ví dụ mà có thể thực hiện các kỹ thuật của sáng chế. Fig.2 được đưa ra nhằm mục đích giải thích và không nên được coi là giới hạn các kỹ thuật như được lấy làm ví dụ và mô tả chung trong bản mô tả này. Nhằm mục đích giải thích, bản mô tả này mô tả bộ mã hóa dữ liệu video 20 trong ngữ cảnh mã hóa HEVC và, ví dụ, phần mở rộng SCC của HEVC. Tuy nhiên, các kỹ thuật trong bản mô tả này có thể áp dụng được cho các phương pháp hoặc chuẩn mã hóa khác.

Bộ mã hóa dữ liệu video 20 thể hiện ví dụ của thiết bị mà có thể được tạo cấu hình để thực hiện các kỹ thuật mã hóa dựa trên bảng màu và mã hóa entropy (ví dụ, CABAC) theo các ví dụ khác nhau được mô tả trong bản mô tả này.

Trong ví dụ trên Fig.2, bộ mã hóa dữ liệu video 20 bao gồm đơn vị mã hóa khối 100, bộ nhớ dữ liệu video 101, đơn vị tạo ra phần dư 102, đơn vị xử lý biến đổi 104, đơn vị lượng tử hóa 106, đơn vị lượng tử hóa ngược 108, đơn vị xử lý biến đổi ngược 110, đơn vị tái tạo 112, đơn vị lọc 114, bộ đệm hình ảnh giải mã 116, và đơn vị mã hóa entropy 118. Đơn vị mã hóa khối 100 bao gồm đơn vị xử lý dự báo liên cấu trúc 120 và đơn vị xử lý dự báo nội cấu trúc 126. Đơn vị xử lý dự báo liên cấu trúc 120 bao gồm đơn vị đánh giá chuyển động và đơn vị bù chuyển động (không được thể hiện trên hình vẽ). Bộ mã hóa dữ liệu video 20 còn bao gồm đơn vị mã hóa dựa trên bảng màu 122 được tạo cấu hình để thực hiện các khía cạnh khác nhau của các kỹ thuật mã hóa dựa trên bảng màu được mô tả trong bản mô tả này. Theo các ví dụ khác, bộ mã hóa dữ liệu video 20 có thể bao gồm nhiều hơn, ít hơn hoặc các thành phần chức năng khác.

Bộ nhớ dữ liệu video 101 có thể lưu trữ dữ liệu video cần được mã hóa bởi các thành phần của bộ mã hóa dữ liệu video 20. Dữ liệu video được lưu trữ trong bộ nhớ

dữ liệu video 101 có thể được thu, ví dụ, từ nguồn dữ liệu video 18. Bộ đệm hình ảnh giải mã 116 có thể là bộ nhớ hình ảnh chuẩn lưu trữ dữ liệu video chuẩn dùng để mã hóa dữ liệu video của bộ mã hóa dữ liệu video 20, ví dụ, trong các chế độ mã hóa nội cấu trúc hoặc liên cấu trúc. Bộ nhớ dữ liệu video 101 và bộ đệm hình ảnh giải mã 116 có thể được tạo ra bởi bất kỳ trong số các thiết bị nhớ, như bộ nhớ truy cập ngẫu nhiên động (dynamic random access memory - DRAM), bao gồm DRAM đồng bộ (synchronous DRAM - SDRAM), RAM từ trở (magnetoresistive RAM - MRAM), RAM điện trở (resistive RAM - RRAM), hoặc các loại thiết bị nhớ khác. Bộ nhớ dữ liệu video 101 và bộ đệm hình ảnh giải mã 116 có thể được tạo ra bởi cùng một thiết bị nhớ hoặc các thiết bị nhớ riêng. Theo các ví dụ khác nhau, bộ nhớ dữ liệu video 101 có thể nằm trên chip cùng các bộ phận khác của bộ mã hóa dữ liệu video 20, hoặc ngoài chip so với các bộ phận này.

Bộ mã hóa dữ liệu video 20 có thể nhận dữ liệu video. Bộ mã hóa dữ liệu video 20 có thể mã hóa mỗi CTU trong một lát của hình ảnh của dữ liệu video. Mỗi CTU có thể được kết hợp với các khối cây mã hóa (coding tree block - CTB) độ chói có kích thước bằng nhau và các CTB tương ứng của hình ảnh. Là một phần của quá trình mã hóa CTU, đơn vị mã hóa khối 100 có thể thực hiện phân chia cây tứ phân để chia các CTB của CTU thành các khối nhỏ dần. Khối nhỏ hơn này có thể là các khối mã hóa của các CU. Ví dụ, đơn vị mã hóa khối 100 có thể phân chia CTB gắn với CTU thành bốn khối con có kích thước bằng nhau, phân chia một hoặc nhiều khối con thành bốn khối con phụ có kích thước bằng nhau, và cứ tiếp tục như vậy.

Bộ mã hóa dữ liệu video 20 có thể mã hóa các CU của CTU để tạo ra các dạng biểu diễn mã hóa của các CU (tức là, các CU mã hóa). Là một phần trong quá trình mã hóa CU, đơn vị mã hóa khối 100 có thể phân chia các khối mã hóa gắn với CU trong một hoặc nhiều PU của CU này. Do đó, mỗi PU có thể được kết hợp với khối dự báo độ chói và các khối dự báo màu tương ứng. Bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể hỗ trợ các PU có các kích thước khác nhau. Như được biểu thị ở trên, kích thước của CU có thể chỉ kích thước khối mã hóa độ chói của CU và kích thước của PU có thể chỉ kích thước khối dự báo độ chói của PU. Giả sử kích thước của một CU cụ thể là $2N \times 2N$, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể hỗ trợ các kích thước PU $2N \times 2N$ hoặc $N \times N$ đối với dự báo nội cấu trúc, và các kích thước PU đối xứng $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, hoặc tương tự đối với dự báo

liên cấu trúc. Bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 cũng có thể hỗ trợ phân chia không đối xứng các kích thước PU $2N \times nU$, $2N \times nD$, $nL \times 2N$, và $nR \times 2N$ đối với dự báo liên cấu trúc.

Đơn vị xử lý dự báo liên cấu trúc 120 có thể tạo ra dữ liệu dự báo cho PU bằng cách thực hiện dự báo liên cấu trúc trên mỗi PU của CU. Dữ liệu dự báo cho PU có thể bao gồm các khối dự báo của PU và thông tin chuyển động cho PU này. Đơn vị dự báo liên cấu trúc 121 có thể thực hiện các thao tác khác nhau cho PU của CU tùy thuộc vào việc PU ở trong lát I, lát P, hay lát B. Trong lát I, tất cả PU đều được dự báo nội cấu trúc. Do đó, nếu PU ở trong lát I, thì đơn vị dự báo liên cấu trúc 121 không thực hiện dự báo liên cấu trúc trên PU. Do đó, đối với các khối được mã hóa trong chế độ I, khối dự báo được tạo ra bằng cách sử dụng dự báo không gian từ các khối lân cận được mã hóa trước đó trong cùng một khung.

Nếu PU ở trong lát P, thì đơn vị đánh giá chuyển động của đơn vị xử lý dự báo liên cấu trúc 120 có thể tìm kiếm hình ảnh chuẩn trong danh sách các hình ảnh chuẩn (ví dụ, “RefPicList0”) cho vùng chuẩn cho PU. Vùng chuẩn cho PU có thể là vùng trong hình ảnh chuẩn mà chứa các khối mẫu tương ứng gần nhất với các khối mẫu của PU. Đơn vị đánh giá chuyển động của đơn vị xử lý dự báo liên cấu trúc 120 có thể tạo ra chỉ số chuẩn mà biểu thị vị trí trong RefPicList0 của hình ảnh chuẩn chứa vùng chuẩn cho PU. Ngoài ra, đơn vị đánh giá chuyển động có thể tạo ra MV mà biểu thị sự dịch chuyển không gian giữa khối mã hóa của PU và vị trí chuẩn gắn với vùng chuẩn. Ví dụ, MV có thể là vectơ hai chiều cung cấp độ dịch từ các tọa độ trong hình được giải mã hiện thời đến các tọa độ trong hình ảnh chuẩn. Đơn vị đánh giá chuyển động có thể xuất ra chỉ số chuẩn và MV dưới dạng thông tin chuyển động của PU. Đơn vị bù chuyển động của đơn vị xử lý dự báo liên cấu trúc 120 có thể tạo ra các khối dự báo của PU dựa trên các mẫu thực hoặc được nội suy tại vị trí chuẩn được chỉ báo bởi vectơ chuyển động của PU.

Nếu PU ở trong lát B, thì đơn vị đánh giá chuyển động có thể thực hiện dự báo đơn hoặc dự báo đôi cho PU. Để thực hiện dự báo đơn cho PU, đơn vị đánh giá chuyển động có thể tìm kiếm các hình ảnh chuẩn của RefPicList0 hoặc danh sách hình ảnh chuẩn thứ hai (“RefPicList1”) cho vùng chuẩn cho PU. Đơn vị đánh giá chuyển động có thể xuất ra, dưới dạng thông tin chuyển động của PU, chỉ số chuẩn mà biểu thị vị trí trong RefPicList0 hoặc RefPicList1 của hình ảnh chuẩn chứa vùng chuẩn, MV

biểu thị sự dịch chuyển không gian giữa khối dự báo của PU và vị trí chuẩn gắn với vùng chuẩn, và một hoặc nhiều bộ chỉ báo hướng dự báo biểu thị liệu hình ảnh chuẩn ở trong RefPicList0 hay RefPicList1. Đơn vị bù chuyển động của đơn vị xử lý dự báo liên cấu trúc 120 có thể tạo ra các khối dự báo của PU dựa ít nhất một phần vào các mẫu thực hoặc được nội suy tại vùng chuẩn được chỉ báo bởi vector chuyển động của PU.

Để thực hiện dự báo liên cấu trúc hai chiều cho PU, đơn vị đánh giá chuyển động có thể tìm kiếm các hình ảnh chuẩn trong RefPicList0 đối với một vùng chuẩn cho PU và cũng có thể tìm kiếm các hình ảnh chuẩn trong RefPicList1 đối với vùng chuẩn khác cho PU này. Đơn vị đánh giá chuyển động có thể tạo ra các chỉ số hình ảnh chuẩn biểu thị vị trí trong RefPicList0 và RefPicList1 của các hình ảnh chuẩn mà chứa các vùng chuẩn. Ngoài ra, đơn vị đánh giá chuyển động có thể tạo ra các MV mà biểu thị sự dịch chuyển không gian giữa vị trí chuẩn mà được kết hợp với các vùng chuẩn và khối mẫu của PU. Thông tin chuyển động của PU có thể bao gồm các chỉ số chuẩn và các MV của PU. Đơn vị bù chuyển động có thể tạo ra các khối dự báo của PU dựa ít nhất một phần vào các mẫu thực hoặc được nội suy tại các vùng chuẩn được chỉ báo bởi các vector chuyển động của PU.

Theo các ví dụ khác nhau của sáng chế, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để thực hiện mã hóa dựa trên bảng màu. Ví dụ, đối với khung HEVC, các kỹ thuật mã hóa dựa trên bảng màu có thể được tạo cấu hình để được sử dụng ở mức CU. Trong các ví dụ khác, các kỹ thuật mã hóa dữ liệu video dựa trên bảng màu có thể được tạo cấu hình để được sử dụng ở mức PU. Trong các ví dụ khác, các kỹ thuật mã hóa dựa trên bảng màu có thể được tạo cấu hình để được sử dụng ở mức đơn vị dự báo con (PU con) (ví dụ, khối con của đơn vị dự báo). Do đó, tất cả quy trình bậc lộ được mô tả trong bản mô tả này (trong toàn bộ sáng chế), trong ngữ cảnh mức của CU có thể, theo cách khác hoặc ngoài ra, áp dụng cho mức PU hoặc mức PU con. Tuy nhiên, các ví dụ dựa trên HEVC này không nên được coi là thu hẹp hoặc hạn chế các kỹ thuật mã hóa dữ liệu video dựa trên bảng màu được mô tả trong bản mô tả này, vì các kỹ thuật đó có thể được áp dụng để làm việc độc lập hoặc như là một phần của các hệ thống/chuẩn hiện có hoặc sắp được phát triển. Trong các trường hợp này, đơn vị để mã hóa bảng màu có thể là các khối vuông, các khối hình chữ nhật hoặc thậm chí cả các vùng có hình dạng không phải hình chữ nhật.

Ví dụ, đơn vị mã hóa dựa trên bảng màu 122 có thể thực hiện giải mã dựa trên bảng màu khi chế độ giải mã dựa trên bảng màu được chọn, chẳng hạn, cho CU hoặc PU. Ví dụ, đơn vị mã hóa dựa trên bảng màu 122 có thể được tạo cấu hình để tạo ra bảng màu có các mục nhập biểu thị các giá trị điểm ảnh, chọn các giá trị điểm ảnh trong bảng màu để biểu diễn các giá trị điểm ảnh của ít nhất một số vị trí trong khối dữ liệu video, và báo hiệu thông tin kết hợp ít nhất một số trong số các vị trí của khối dữ liệu video với các mục nhập trong bảng màu tương ứng lần lượt với các giá trị điểm ảnh đã chọn. Mặc dù các chức năng khác nhau được mô tả là được thực hiện bởi đơn vị mã hóa dựa trên bảng màu 122, nhưng một số hoặc toàn bộ các chức năng này có thể được thực hiện bởi các đơn vị xử lý khác, hoặc tổ hợp các đơn vị xử lý khác nhau.

Theo các khía cạnh của sáng chế, đơn vị mã hóa dựa trên bảng màu 122 có thể được tạo cấu hình để thực hiện sự kết hợp bất kỳ của các kỹ thuật mã hóa bảng màu được mô tả ở đây.

Đơn vị xử lý dự báo nội cấu trúc 126 có thể tạo ra các dữ liệu dự báo cho PU bằng cách thực hiện dự báo nội cấu trúc trên PU này. Dữ liệu dự báo cho PU có thể bao gồm các khối dự báo cho PU và các phần tử cú pháp khác nhau. Đơn vị xử lý dự báo nội cấu trúc 126 có thể thực hiện dự báo nội cấu trúc trên các PU trong các lát I, lát P và lát B.

Để thực hiện dự báo nội cấu trúc trên PU, đơn vị xử lý dự báo nội cấu trúc 126 có thể sử dụng nhiều chế độ dự báo nội cấu trúc để tạo ra nhiều tập hợp dữ liệu dự báo cho PU. Đơn vị xử lý dự báo nội cấu trúc 126 có thể sử dụng mẫu từ các khối mẫu của các PU lân cận để tạo ra khối dự báo cho PU. Các PU lân cận có thể ở trên, ở trên về phía phải, ở trên về phía trái, hoặc ở bên trái PU, giả sử thứ tự mã hóa từ trái sang phải, từ trên xuống dưới, cho các PU, CU và CTU. Đơn vị xử lý dự báo nội cấu trúc 126 có thể sử dụng số lượng chế độ dự báo nội cấu trúc khác nhau, ví dụ, 33 chế độ dự báo nội cấu trúc có hướng. Trong một số ví dụ, số lượng chế độ dự báo nội cấu trúc có thể phụ thuộc vào kích thước của vùng được kết hợp với PU đó.

Đơn vị mã hóa khối 100 có thể chọn dữ liệu dự báo cho các PU của CU từ dữ liệu dự báo được tạo ra bởi đơn vị xử lý dự báo liên cấu trúc 120 cho các PU hoặc dữ liệu dự báo được tạo ra bởi đơn vị xử lý dự báo nội cấu trúc 126 cho các PU. Trong một số ví dụ, đơn vị mã hóa khối 100 chọn dữ liệu dự báo cho các PU của CU dựa trên các giá trị đo tốc độ/méo của các tập hợp dữ liệu dự báo. Các khối dự báo của dữ liệu

dự báo được chọn có thể được gọi ở đây là các khối dự báo được chọn.

Đơn vị tạo ra phần dư 102 có thể tạo ra, dựa vào khối mã hóa độ chói, Cb và Cr của CU và các khối dự báo độ chói, Cb và Cr được chọn của các PU trong CU, các khối dư độ chói, Cb và Cr của CU. Ví dụ, đơn vị tạo ra phần dư 102 có thể tạo ra các khối dư của CU sao cho mỗi mẫu trong các khối dư này có giá trị bằng với hiệu số giữa mẫu trong khối mã hóa của CU và mẫu tương ứng trong khối dự báo được chọn tương ứng của PU trong CU.

Đơn vị xử lý biến đổi 104 có thể thực hiện phân chia cây tứ phân để phân chia các khối dư gắn với CU thành các khối biến đổi gắn với các TU của CU. Do đó, trong một số ví dụ, TU có thể được kết hợp với khối biến đổi độ chói và hai khối biến đổi màu. Kích thước và vị trí của các khối biến đổi độ chói và màu của các TU trong CU có thể dựa vào hoặc có thể không dựa vào kích thước và vị trí của các khối dự báo của các PU trong CU. Cấu trúc cây tứ phân, được biết đến là “cây tứ phân dư” (residual quad-tree - RQT) có thể bao gồm các nút gắn với mỗi trong số các vùng. Các TU của CU có thể tương ứng với các nút lá của RQT.

Đơn vị xử lý biến đổi 104 có thể tạo ra các khối hệ số biến đổi cho mỗi TU của CU bằng cách áp dụng một hoặc nhiều phép biến đổi cho các khối biến đổi của TU. Đơn vị xử lý biến đổi 104 có thể áp dụng các phép biến đổi khác nhau cho khối biến đổi gắn với TU. Ví dụ, đơn vị xử lý biến đổi 104 có thể áp dụng phép biến đổi cosin rời rạc (discrete cosine transform - DCT), biến đổi có hướng, hoặc biến đổi tương tự về mặt khái niệm cho khối biến đổi. Trong một số ví dụ, đơn vị xử lý biến đổi 104 không áp dụng phép biến đổi cho khối biến đổi. Trong các ví dụ này, khối biến đổi có thể được xử lý dưới dạng khối hệ số biến đổi.

Đơn vị lượng tử hóa 106 có thể lượng tử hóa các hệ số biến đổi trong khối hệ số. Quy trình lượng tử hóa có thể làm giảm độ sâu bit gắn với một số hoặc tất cả các hệ số biến đổi. Ví dụ, hệ số biến đổi n bit có thể được làm tròn xuống hệ số biến đổi m bit trong quá trình lượng tử hóa, trong đó n lớn hơn m . Đơn vị lượng tử hóa 106 có thể lượng tử hóa khối hệ số gắn với TU trong CU dựa trên giá trị tham số lượng tử hóa (quantization parameter - QP) gắn với CU này. Bộ mã hóa dữ liệu video 20 có thể điều chỉnh mức độ lượng tử hóa áp dụng cho các khối hệ số gắn với CU bằng cách điều chỉnh giá trị QP gắn với CU này. Việc lượng tử hóa có thể gây ra tổn hao thông tin, do đó các hệ số biến đổi lượng tử hóa có thể có độ chính xác thấp hơn các hệ số biến đổi

gốc.

Đơn vị lượng tử hóa ngược 108 và đơn vị xử lý biến đổi ngược 110 có thể áp dụng lần lượt phép biến đổi lượng tử hóa ngược và biến đổi ngược cho khối hệ số để tái tạo khối dư từ khối hệ số. Đơn vị tái tạo 112 có thể bổ sung khối dư tái tạo cho các mẫu tương ứng từ một hoặc nhiều khối dự báo được tạo ra bởi đơn vị mã hóa khối 100 để tạo ra khối biến đổi tái tạo gắn với TU. Bằng cách tái tạo các khối biến đổi cho từng TU của CU theo cách này, bộ mã hóa dữ liệu video 20 có thể tái tạo các khối mã hóa của CU này.

Đơn vị lọc 114 có thể thực hiện một hoặc nhiều thao tác tách khối để giảm các thành phần lạ dạng khối trong các khối mã hóa gắn với CU. Đơn vị lọc 114 có thể thực hiện các thao tác lọc khác, bao gồm lọc bù thích nghi mẫu (sample adaptive offset - SAO) và/hoặc lọc vòng thích nghi (adaptive loop filtering - ALF). Bộ đệm hình ảnh giải mã 116 có thể lưu trữ các khối mã hóa tái tạo sau khi đơn vị lọc 114 thực hiện một hoặc nhiều thao tác tách khối trên các khối mã hóa tái tạo. Đơn vị xử lý dự báo liên cấu trúc 120 có thể sử dụng hình ảnh chuẩn chứa các khối mã hóa tái tạo để thực hiện dự báo liên cấu trúc trên các PU của các hình ảnh khác. Ngoài ra, đơn vị xử lý dự báo nội cấu trúc 126 có thể sử dụng các khối mã hóa tái tạo trong bộ đệm hình ảnh giải mã 116 để thực hiện dự báo nội cấu trúc trên các PU khác trong cùng hình ảnh với CU.

Đơn vị mã hóa entropy 118 có thể nhận dữ liệu từ các thành phần chức năng khác của bộ mã hóa dữ liệu video 20. Ví dụ, đơn vị mã hóa entropy 118 có thể nhận các khối hệ số từ đơn vị lượng tử hóa 106 và có thể nhận các phần tử cú pháp từ đơn vị mã hóa khối 100. Đơn vị mã hóa entropy 118 có thể thực hiện một hoặc nhiều thao tác mã hóa entropy trên dữ liệu để tạo ra dữ liệu được mã hóa entropy. Ví dụ, đơn vị mã hóa entropy 118 có thể thực hiện thao tác mã hóa thích ứng ngữ cảnh, ví dụ như thao tác mã hóa CABAC, thao tác mã hóa độ dài thay đổi thích ứng theo ngữ cảnh (context-adaptive variable length coding - CAVLC), thao tác mã hóa độ dài biến - biến (variable-to-variable - V2V), thao tác mã hóa số học nhị phân thích ứng theo ngữ cảnh dựa trên cú pháp (syntax-based context-adaptive binary arithmetic coding - SBAC), thao tác mã hóa entropy phân chia khoảng xác suất (Probability Interval Partitioning Entropy - PIPE), thao tác mã hóa Golomb-hàm mũ, hoặc kiểu thao tác mã hóa entropy khác trên dữ liệu. Bộ mã hóa dữ liệu video 20 có thể xuất ra dòng bit bao gồm dữ liệu được mã hóa entropy tạo ra bởi đơn vị mã hóa entropy 118. Ví dụ, dòng bit này có thể

chứa dữ liệu biểu diễn RQT cho CU.

Trong một số ví dụ, thao tác mã hóa dư không được thực hiện với thao tác mã hóa bảng màu. Theo đó, bộ mã hóa dữ liệu video 20 có thể không thực hiện biến đổi hoặc lượng tử hóa khi thao tác mã hóa sử dụng chế độ mã hóa bảng màu. Ngoài ra, bộ mã hóa dữ liệu video 20 có thể mã hóa entropy dữ liệu tạo ra bằng cách sử dụng chế độ mã hóa bảng màu riêng biệt với dữ liệu dư.

Theo một hoặc nhiều kỹ thuật của sáng chế, bộ mã hóa dữ liệu video 20, và cụ thể là đơn vị mã hóa dựa trên bảng màu 122, có thể thực hiện mã hóa dữ liệu video dựa trên bảng màu đối với các khối video được dự báo. Như được mô tả ở trên, bảng màu được tạo ra bởi bộ mã hóa dữ liệu video 20 có thể được mã hóa rõ ràng và được gửi cho bộ giải mã dữ liệu video 30, được dự báo từ các mục nhập bảng màu trước đó, được dự báo từ các giá trị điểm ảnh trước đó, hoặc tổ hợp của chúng.

Theo một hoặc nhiều kỹ thuật của sáng chế, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để xác định rằng khối hiện thời được mã hóa trong chế độ bảng màu, mã hóa trong chế độ bỏ qua các thể hiện của phần tử cú pháp thứ nhất để tái tạo khối hiện thời, và sau khi mã hóa trong chế độ bỏ qua các thể hiện của phần tử cú pháp thứ nhất, mã hóa trong chế độ ngữ cảnh các thể hiện của phần tử cú pháp thứ hai để tái tạo khối hiện thời, ví dụ, bằng cách sử dụng quy trình mã hóa CABAC. Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa trong chế độ bỏ qua hai thể hiện bất kỳ trong số các thể hiện của phần tử cú pháp thứ nhất, ví dụ, bằng cách sử dụng chế độ bỏ qua của quy trình mã hóa CABAC, mà không đan xen với việc mã hóa trong chế độ ngữ cảnh một thể hiện bất kỳ trong số các thể hiện của phần tử cú pháp thứ hai. Trong một ví dụ, phần tử cú pháp thứ nhất bao gồm một trong phần tử cú pháp `palette_index_idc` hoặc phần tử cú pháp `palette_escape_val`, và phần tử cú pháp thứ hai bao gồm phần tử cú pháp `palette_run_msb_id_plus1`. Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa bỏ qua các thể hiện của phần tử cú pháp thứ nhất ở trước đoạn mã hóa khối chỉ số cho khối hiện thời.

Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa phần tử cú pháp thứ ba biểu thị số lượng thể hiện của phần tử cú pháp thứ nhất, trong đó bước mã hóa trong chế độ bỏ qua các thể hiện của phần tử cú pháp thứ nhất bao gồm mã hóa trong chế độ bỏ qua các thể hiện của phần tử cú pháp thứ nhất dựa vào phần tử cú pháp thứ ba. Bộ mã hóa dữ liệu video 20 có thể mã hóa phần tử cú pháp thứ ba bằng cách sử dụng

dụng một trong số mã Golomb Rice, mã Golomb hàm mũ, mã Rice rút gọn, mã đơn phân, hoặc dạng ghép nối của mã Golomb Rice và mã Golomb hàm mũ, hoặc phiên bản rút gọn của bất kỳ trong số các mã trên.

Fig.3 là sơ đồ khối thể hiện bộ giải mã dữ liệu video 30 làm ví dụ được tạo cấu hình để thực hiện các kỹ thuật của sáng chế. Fig.3 được đưa ra nhằm mục đích giải thích và không hạn chế các kỹ thuật được lấy làm ví dụ và mô tả chung trong bản mô tả này. Nhằm mục đích giải thích, sáng chế mô tả bộ giải mã dữ liệu video 30 trong ngữ cảnh mã hóa HEVC. Tuy nhiên, các kỹ thuật trong bản mô tả này có thể áp dụng được cho các phương pháp hoặc chuẩn mã hóa khác.

Chi tiết về mã hóa bảng màu được mô tả trên đây đối với bộ mã hóa 20 không được lặp lại ở đây đối với bộ giải mã 30, nhưng cần hiểu rằng bộ giải mã 30 có thể thực hiện quy trình giải mã ngược lại so với quy trình mã hóa bất kỳ được mô tả trong đây đối với bộ mã hóa 20.

Bộ giải mã dữ liệu video 30 là ví dụ về thiết bị mà có thể được tạo cấu hình để thực hiện các kỹ thuật mã hóa dựa trên bảng màu và mã hóa entropy (ví dụ, CABAC) theo các ví dụ khác nhau được mô tả trong sáng chế.

Trong ví dụ trên Fig.3, bộ giải mã dữ liệu video 30 bao gồm đơn vị giải mã entropy 150, bộ nhớ dữ liệu video, đơn vị giải mã khối 152, đơn vị lượng tử hóa ngược 154, đơn vị xử lý biến đổi ngược 156, đơn vị tái tạo 158, đơn vị lọc 160, và bộ đệm hình ảnh giải mã 162. Đơn vị giải mã khối 152 bao gồm đơn vị bù chuyển động 164 và đơn vị xử lý dự báo nội cấu trúc 166. Bộ giải mã dữ liệu video 30 còn bao gồm đơn vị giải mã dựa trên bảng màu 165 được tạo cấu hình để thực hiện các khía cạnh khác nhau của các kỹ thuật mã hóa dựa trên bảng màu được mô tả theo sáng chế. Trong các ví dụ khác, bộ giải mã dữ liệu video 30 có thể bao gồm nhiều hơn, ít hơn, hoặc các thành phần chức năng khác.

Bộ nhớ dữ liệu video 151 có thể lưu trữ dữ liệu video, như dòng bit video mã hóa, cần được giải mã bởi các thành phần của bộ giải mã dữ liệu video 30. Dữ liệu video được lưu trữ trong bộ nhớ dữ liệu video 151 có thể được thu, ví dụ, từ vật ghi đọc được bằng máy tính 16, ví dụ, từ nguồn video cục bộ, như máy quay, thông qua việc truyền thông dữ liệu video qua mạng có dây hoặc không dây, hoặc bằng cách truy cập vật ghi dữ liệu vật lý. Bộ nhớ dữ liệu video 151 có thể tạo ra bộ đệm hình ảnh mã hóa (CPB - coded picture buffer) mà lưu trữ dữ liệu video mã hóa từ dòng bit video mã

hóa. Bộ đệm hình ảnh giải mã 162 có thể là bộ nhớ hình ảnh chuẩn lưu trữ dữ liệu video chuẩn để dùng để giải mã dữ liệu video bởi bộ giải mã dữ liệu video 30, ví dụ, trong các chế độ mã hóa nội cấu trúc hoặc liên cấu trúc. Bộ nhớ dữ liệu video 151 và bộ đệm hình ảnh giải mã 162 có thể được tạo ra bởi thiết bị bất kỳ trong số các thiết bị nhớ, như bộ nhớ truy cập ngẫu nhiên động (dynamic random access memory - DRAM), bao gồm DRAM đồng bộ (synchronous DRAM - SDRAM), RAM từ trở (magnetoresistive RAM - MRAM), RAM điện trở (resistive RAM - RRAM), hoặc các loại thiết bị nhớ khác. Bộ nhớ dữ liệu video 151 và bộ đệm hình ảnh giải mã 162 có thể được tạo ra bởi cùng một thiết bị nhớ hoặc các thiết bị nhớ riêng. Trong các ví dụ khác nhau, bộ nhớ dữ liệu video 151 có thể nằm trên chip cùng các bộ phận khác của bộ giải mã dữ liệu video 30, hoặc ngoài chip so với các bộ phận đó.

Bộ đệm hình ảnh mã hóa (coded picture buffer-CPB), bộ đệm này có thể được tạo ra bởi bộ nhớ dữ liệu video 151, có thể nhận và lưu trữ dữ liệu video mã hóa (ví dụ, các đơn vị NAL) của dòng bit. Đơn vị giải mã entropy 150 có thể nhận được dữ liệu video mã hóa (ví dụ, các đơn vị NAL) từ CPB và phân tích cú pháp các đơn vị NAL để giải mã các phần tử cú pháp. Đơn vị giải mã entropy 150 có thể giải mã entropy các phần tử cú pháp được mã hóa entropy trong các đơn vị NAL. Đơn vị giải mã khối 152, đơn vị lượng tử hóa ngược 154, đơn vị xử lý biến đổi ngược 156, đơn vị tái tạo 158, và đơn vị lọc 160 có thể tạo ra dữ liệu video giải mã dựa trên các phần tử cú pháp trích ra từ dòng bit.

Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để thực hiện quy trình thường ngược với quy trình của bộ mã hóa dữ liệu video 20 mô tả ở đây. Tương tự, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để thực hiện quy trình thường ngược với quy trình của bộ giải mã dữ liệu video 30 mô tả ở đây. Ví dụ, giả sử rằng bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã phần tử cú pháp được mã hóa trong dòng bit cũng như giả sử rằng bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa phần tử cú pháp trong dòng bit.

Theo ví dụ khác, đơn vị giải mã entropy 150 có thể được tạo cấu hình để thực hiện quy trình thường ngược với quy trình của đơn vị mã hóa entropy 118 mô tả ở đây. Theo các khía cạnh của sáng chế, đơn vị giải mã entropy 150 có thể được tạo cấu hình để giải mã entropy mọi từ mã được tạo ra bởi đơn vị mã hóa entropy 118. Ví dụ, đơn vị giải mã entropy 150 có thể được tạo cấu hình để giải mã entropy các giá trị được mã

hóa Exp-Golomb rút gọn (truncated Exp-Golomb - TEGk) bậc thứ k đều và không đều, ví dụ như vector dự báo bảng màu nhị phân và/hoặc ánh xạ bảng màu cho CU. Theo ví dụ khác, đơn vị giải mã entropy 150 có thể được tạo cấu hình để giải mã entropy từ mã Exp-Golomb (EGk) bậc thứ k, từ mã Exp-Golomb rút gọn (TEGk) bậc thứ k, từ mã Exp-Golomb rút gọn (TEGk) không đều bậc thứ k, hoặc tổ hợp bất kỳ của chúng.

Các đơn vị NAL của dòng bit có thể bao gồm các đơn vị NAL lát mã hóa. Để giải mã dòng bit, đơn vị giải mã entropy 150 có thể trích và giải mã entropy các phần tử cú pháp từ các đơn vị NAL lát mã hóa. Mỗi lát mã hóa có thể bao gồm phần đầu lát và dữ liệu lát. Phần đầu lát có thể chứa các phần tử cú pháp gắn liền với lát. Các phần tử cú pháp trong phần đầu lát có thể bao gồm phần tử cú pháp nhận dạng PPS gắn với hình ảnh chứa lát này.

Ngoài việc giải mã các phần tử cú pháp từ dòng bit, bộ giải mã dữ liệu video 30 có thể thực hiện thao tác tái tạo trên CU không phân chia. Để thực hiện thao tác tái tạo trên CU không phân chia, bộ giải mã dữ liệu video 30 có thể thực hiện thao tác tái tạo trên từng TU của CU này. Bằng cách thực hiện thao tác tái tạo trên từng TU của CU, bộ giải mã dữ liệu video 30 có thể tái tạo các khối dư của CU.

Là một phần của việc thực hiện thao tác tái tạo trên TU của CU, đơn vị lượng tử hóa ngược 154 có thể lượng tử hóa ngược, tức là, khử lượng tử hóa, các khối hệ số gắn với TU này. Đơn vị lượng tử hóa ngược 154 có thể sử dụng giá trị QP gắn với CU của TU để xác định mức lượng tử hóa và, tương tự, mức lượng tử hóa ngược cho đơn vị lượng tử hóa ngược 154 áp dụng. Tức là, tỷ lệ nén, ví dụ, tỷ lệ số lượng bit được sử dụng để biểu diễn chuỗi gốc và chuỗi được nén, có thể được kiểm soát bằng cách điều chỉnh giá trị của QP được sử dụng khi lượng tử hóa các hệ số biến đổi. Tỷ lệ nén này cũng có thể phụ thuộc vào phương pháp mã hóa entropy được dùng.

Sau khi đơn vị lượng tử hóa ngược 154 lượng tử hóa ngược khối hệ số, đơn vị xử lý biến đổi ngược 156 có thể áp dụng một hoặc nhiều phép biến đổi ngược cho khối hệ số để tạo ra khối dư gắn với TU. Ví dụ, đơn vị xử lý biến đổi ngược 156 có thể áp dụng phép DCT ngược, biến đổi số nguyên ngược, biến đổi Karhunen-Loeve ngược (Karhunen-Loeve transform - KLT), biến đổi quay ngược, biến đổi hướng ngược, hoặc phép biến đổi ngược khác cho khối hệ số.

Nếu PU được mã hóa bằng cách sử dụng phương pháp dự báo nội cấu trúc, thì

đơn vị xử lý dự báo nội cấu trúc 166 có thể thực hiện dự báo nội cấu trúc để tạo ra các khối dự báo cho PU này. Đơn vị xử lý dự báo nội cấu trúc 166 có thể sử dụng chế độ dự báo nội cấu trúc để tạo ra các khối dự báo độ chói, Cb và Cr cho PU dựa trên các khối dự báo của các PU lân cận về mặt không gian. Đơn vị xử lý dự báo nội cấu trúc 166 có thể xác định chế độ dự báo nội cấu trúc cho PU dựa trên một hoặc nhiều phần tử cú pháp được giải mã từ dòng bit.

Đơn vị giải mã khối 152 có thể xây dựng danh sách hình ảnh chuẩn thứ nhất (RefPicList0) và danh sách hình ảnh chuẩn thứ hai (RefPicList1) dựa trên các phần tử cú pháp trích ra từ dòng bit. Hơn thế nữa, nếu PU được mã hóa bằng cách sử dụng phương pháp dự báo liên cấu trúc, thì đơn vị giải mã entropy 150 có thể trích ra thông tin chuyển động cho PU. Đơn vị bù chuyển động 164 có thể xác định một hoặc nhiều vùng chuẩn cho PU dựa trên thông tin chuyển động của PU này. Đơn vị bù chuyển động 164 có thể tạo ra các khối dự báo độ chói, Cb và Cr cho PU dựa trên các khối mẫu tại một hoặc nhiều khối chuẩn cho PU.

Đơn vị tái tạo 158 có thể sử dụng các khối biến đổi độ chói, Cb và Cr gắn với các TU của CU và các khối dự báo độ chói, Cb và Cr của các PU trong CU, tức là dữ liệu dự báo nội cấu trúc hoặc dữ liệu dự báo liên cấu trúc, tùy trường hợp, để tái tạo các khối mã hóa độ chói, Cb và Cr của CU. Ví dụ, đơn vị tái tạo 158 có thể bổ sung các mẫu của các khối biến đổi độ chói, Cb và Cr vào các mẫu tương ứng của các khối dự báo độ chói, Cb và Cr để tái tạo các khối mã hóa độ chói, Cb và Cr của CU.

Đơn vị lọc 160 có thể thực hiện thao tác tách khối để giảm các thành phần lạ dạng khối gắn với các khối mã hóa độ chói, Cb và Cr của CU. Bộ giải mã dữ liệu video 30 có thể lưu trữ các khối mã hóa độ chói, Cb và Cr của CU vào bộ đệm hình ảnh giải mã 162. Bộ đệm hình ảnh giải mã 162 có thể cung cấp các hình ảnh chuẩn cho thao tác bù chuyển động tiếp theo, dự báo nội cấu trúc, và trình diễn trên thiết bị hiển thị, chẳng hạn như thiết bị hiển thị 32 trên Fig.1. Ví dụ, bộ giải mã dữ liệu video 30 có thể thực hiện, dựa trên các khối độ chói, Cb và Cr trong bộ đệm hình ảnh giải mã 162, các thao tác dự báo nội cấu trúc hoặc dự báo liên cấu trúc trên các PU của các CU khác.

Theo các ví dụ khác nhau trong bản mô tả này, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để thực hiện mã hóa dựa trên bảng màu. Đơn vị giải mã dựa trên bảng màu 165, ví dụ, có thể thực hiện giải mã dựa trên bảng màu khi chế độ giải mã

dựa trên bảng màu được chọn, ví dụ cho CU hoặc PU. Ví dụ, đơn vị giải mã dựa trên bảng màu 165 có thể được tạo cấu hình để tạo ra bảng màu có các mục nhập biểu thị các giá trị điểm ảnh, nhận thông tin kết hợp ít nhất một số vị trí điểm ảnh trong khối dữ liệu video với các mục nhập trong bảng màu, chọn các giá trị điểm ảnh trong bảng màu dựa trên thông tin, và tái tạo các giá trị điểm ảnh của khối dựa trên các giá trị điểm ảnh được chọn trong bảng màu. Mặc dù các chức năng khác nhau được mô tả là được thực hiện bởi đơn vị giải mã dựa trên bảng màu 165, nhưng một số hoặc toàn bộ chức năng này có thể được thực hiện bởi các đơn vị xử lý khác, hoặc tổ hợp các đơn vị xử lý khác nhau.

Đơn vị giải mã dựa trên bảng màu 165 có thể nhận thông tin chế độ mã hóa bảng màu, và thực hiện các thao tác ở trên khi thông tin chế độ mã hóa bảng màu biểu thị rằng chế độ mã hóa bảng màu áp dụng cho khối. Khi thông tin chế độ mã hóa bảng màu biểu thị rằng chế độ mã hóa bảng màu không áp dụng cho khối, hoặc khi thông tin chế độ khác biểu thị việc sử dụng chế độ khác, đơn vị giải mã dựa trên bảng màu 165 giải mã khối dữ liệu video bằng cách sử dụng chế độ mã hóa không dựa trên bảng màu, chẳng hạn như chế độ mã hóa dự báo nội cấu trúc hoặc dự báo liên cấu trúc HEVC. Khối dữ liệu video này có thể là, ví dụ, CU hoặc PU được tạo ra theo quy trình mã hóa HEVC. Chế độ mã hóa dựa trên bảng màu có thể bao gồm một trong nhiều chế độ mã hóa dựa trên bảng màu khác nhau, hoặc có thể có một chế độ mã hóa dựa trên bảng màu.

Theo các khía cạnh của sáng chế, đơn vị giải mã dựa trên bảng màu 165 có thể được tạo cấu hình để thực hiện sự kết hợp bất kỳ của các kỹ thuật mã hóa bảng màu được mô tả ở đây. Chi tiết về mã hóa bảng màu được mô tả trên đây đối với bộ mã hóa 20 không được lặp lại ở đây đối với bộ giải mã 30, nhưng cần hiểu rằng bộ giải mã 30 có thể thực hiện quy trình giải mã dựa trên bảng màu ngược với quy trình mã hóa dựa trên bảng màu bất kỳ được mô tả ở đây đối với bộ mã hóa 20.

Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định rằng khối hiện thời được mã hóa trong chế độ bảng màu, giải mã trong chế độ bỏ qua các thể hiện của phần tử cú pháp thứ nhất để tái tạo khối hiện thời, ví dụ, bằng cách sử dụng chế độ bỏ qua của quy trình mã hóa CABAC, và sau khi giải mã trong chế độ bỏ qua các thể hiện của phần tử cú pháp thứ nhất, giải mã trong chế độ ngữ cảnh các thể hiện của phần tử cú pháp thứ hai để tái tạo khối hiện thời, ví dụ, bằng cách sử dụng quy trình mã hóa

CABAC. Bộ giải mã dữ liệu video 30 có thể giải mã trong chế độ bỏ qua hai thể hiện bất kỳ trong số các thể hiện của phần tử cú pháp thứ nhất mà không đan xen với việc giải mã trong chế độ ngữ cảnh một thể hiện bất kỳ trong số các thể hiện của phần tử cú pháp thứ hai. Trong một số ví dụ, phần tử cú pháp thứ nhất bao gồm một trong phần tử cú pháp `palette_index_idc` hoặc phần tử cú pháp `palette_escape_val`, và phần tử cú pháp thứ hai bao gồm phần tử cú pháp `palette_run_msb_id_plus1`. Bộ giải mã dữ liệu video 30 có thể giải mã bỏ qua các thể hiện của phần tử cú pháp thứ nhất ở trước đoạn mã hóa khối chỉ số cho khối hiện thời.

Bộ giải mã dữ liệu video 30 có thể giải mã phần tử cú pháp thứ ba biểu thị số lượng thể hiện của phần tử cú pháp thứ nhất, trong đó bước giải mã trong chế độ bỏ qua các thể hiện của phần tử cú pháp thứ nhất bao gồm bước giải mã trong chế độ bỏ qua các thể hiện của phần tử cú pháp thứ nhất dựa vào phần tử cú pháp thứ ba. Bộ giải mã dữ liệu video 30 có thể giải mã phần tử cú pháp thứ ba bằng cách sử dụng một trong số mã Golomb Rice, mã Golomb hàm mũ, mã Rice rút gọn, mã đơn phân, hoặc dạng ghép nối của mã Golomb Rice và mã Golomb hàm mũ, hoặc phiên bản rút gọn của mã bất kỳ trong số các mã trên.

Fig.4 là sơ đồ khái niệm thể hiện ví dụ về việc xác định bảng màu để mã hóa dữ liệu video, theo các kỹ thuật của sáng chế. Ví dụ trên Fig.4 bao gồm hình ảnh 178 có đơn vị mã hóa (CU) PAL (bảng màu) thứ nhất 188 được kết hợp với các bảng màu thứ nhất 184 và CU PAL thứ hai 188 được kết hợp với các bảng màu thứ hai 192. Như được mô tả chi tiết hơn dưới đây và theo các kỹ thuật của sáng chế, các bảng màu thứ hai 192 được dựa trên các bảng màu thứ nhất 184. Hình ảnh 178 cũng bao gồm khối 196 được mã hóa theo chế độ mã hóa dự báo nội cấu trúc và khối 200 được mã hóa theo chế độ mã hóa dự báo liên cấu trúc.

Các kỹ thuật trên Fig.4 được mô tả trong ngữ cảnh của bộ mã hóa dữ liệu video 20 (Fig.1 và Fig.2) và bộ giải mã dữ liệu video 30 (Fig.1 và Fig.3) và theo chuẩn mã hóa dữ liệu video HEVC nhằm mục đích giải thích. Tuy nhiên, cần hiểu rằng các kỹ thuật trong bản mô tả này không bị giới hạn theo cách này, và có thể được áp dụng bởi các bộ xử lý mã hóa dữ liệu video khác và/hoặc các thiết bị trong các quy trình và/hoặc các chuẩn mã hóa dữ liệu video khác.

Nói chung, bảng màu chỉ số lượng giá trị điểm ảnh trội nhất và/hoặc đại diện cho CU hiện đang được mã hóa, CU 188 theo ví dụ trên Fig.4. Các bảng màu thứ nhất

184 (mà còn có thể được gọi là các chỉ số 184) và các bảng màu thứ hai 192 (mà còn có thể được gọi là các chỉ số 192) được thể hiện là bao gồm nhiều bảng màu (mà còn có thể được gọi là nhiều chỉ số). Trong một số ví dụ, theo các khía cạnh trong bản mô tả, bộ mã hóa dữ liệu video (chẳng hạn như bộ mã hóa dữ liệu video 20 hoặc bộ giải mã dữ liệu video 30) có thể mã hóa các bảng màu (ví dụ, các chỉ số) riêng rẽ cho từng thành phần màu sắc của CU. Ví dụ, bộ mã hóa dữ liệu video 20 có thể mã hóa một bảng màu cho thành phần độ chói (Y) của CU, bảng màu khác cho thành phần màu (U) của CU, và bảng màu khác nữa cho thành phần màu (V) của CU. Trong ví dụ này, các mục nhập của bảng màu Y có thể biểu diễn các giá trị điểm ảnh Y của CU, các mục nhập của bảng màu U có thể biểu diễn các giá trị điểm ảnh U của CU, và các mục nhập của bảng màu V có thể biểu diễn các giá trị điểm ảnh V của CU.

Theo các ví dụ khác, bộ mã hóa dữ liệu video 20 có thể mã hóa một bảng màu cho tất cả thành phần màu sắc của CU. Trong ví dụ này, bộ mã hóa dữ liệu video 20 có thể mã hóa bảng màu có mục nhập thứ i là giá trị bội ba, bao gồm Y_i , U_i , và V_i . Trong trường hợp này, bảng màu bao gồm các giá trị cho mỗi thành phần của các điểm ảnh. Do đó, việc thể hiện các bảng màu 184 và 192 dưới dạng tập hợp các bảng màu gồm nhiều bảng màu riêng lẻ chỉ là một ví dụ minh họa chứ không dự định giới hạn phạm vi sáng chế.

Trong ví dụ trên Fig.4, các bảng màu thứ nhất 184 bao gồm ba mục nhập 202-206 lần lượt có giá trị chỉ số mục nhập 1, giá trị chỉ số mục nhập 2, và giá trị chỉ số mục nhập 3. Các bảng màu thứ nhất 184 liên kết các giá trị chỉ số (ví dụ, các giá trị chỉ số thể hiện ở cột trái của các bảng màu thứ nhất 184) với các giá trị điểm ảnh. Ví dụ, như được thể hiện trên Fig.4, một trong các bảng màu thứ nhất 184 lần lượt liên kết các giá trị chỉ số 1, 2, và 3 với các giá trị điểm ảnh A, B, và C. Như được mô tả ở đây, thay vì mã hóa các giá trị điểm ảnh thực của CU thứ nhất 180, bộ mã hóa dữ liệu video (chẳng hạn như bộ mã hóa dữ liệu video 20 hoặc bộ giải mã dữ liệu video 30) có thể sử dụng kỹ thuật mã hóa dựa trên bảng màu để mã hóa các điểm ảnh của khối bằng cách sử dụng các chỉ số từ 1 đến 3 (mà còn có thể được biểu diễn là các giá trị chỉ số từ 1 đến 3). Tức là, đối với mỗi vị trí điểm ảnh của CU thứ nhất 180, bộ mã hóa dữ liệu video 20 có thể mã hóa giá trị chỉ số cho điểm ảnh này, trong đó giá trị chỉ số được kết hợp với giá trị điểm ảnh trong một hoặc nhiều bảng màu thứ nhất 184. Bộ giải mã dữ liệu video 30 có thể thu được các giá trị chỉ số từ dòng bit và tái tạo các giá

trị điểm ảnh bằng cách sử dụng các giá trị chỉ số và một hoặc nhiều bảng màu thứ nhất 184. Do đó, các bảng màu thứ nhất 184 được truyền bởi bộ mã hóa dữ liệu video 20 trong dòng bit dữ liệu video mã hóa để sử dụng bởi bộ giải mã dữ liệu video 30 khi giải mã dựa trên bảng màu.

Theo một số ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể xác định các bảng màu thứ hai 192 dựa trên các bảng màu thứ nhất 184. Ví dụ, bộ mã hóa dữ liệu video 20 và/hoặc bộ giải mã dữ liệu video 30 có thể định vị một hoặc nhiều khối mà từ đó xác định được các bảng màu dự báo, trong ví dụ này là các bảng màu thứ nhất 184. Trong một số ví dụ, chẳng hạn như ví dụ được minh họa trên Fig.4, bộ mã hóa dữ liệu video 20 và/hoặc bộ giải mã dữ liệu video 30 có thể định vị CU được mã hóa trước đó chẳng hạn như CU liền kề bên trái (CU thứ nhất 180) khi xác định bảng màu dự báo cho CU thứ hai 188.

Trong ví dụ trên Fig.4, các bảng màu thứ hai 192 bao gồm ba mục nhập 208-212 lần lượt có giá trị chỉ số mục nhập 1, giá trị chỉ số mục nhập 2, và giá trị chỉ số mục nhập 3. Các bảng màu thứ hai 192 liên kết các giá trị chỉ số (ví dụ, các giá trị chỉ số thể hiện trong cột trái của các bảng màu thứ nhất 192) với các giá trị điểm ảnh. Ví dụ, như được thể hiện trên Fig.4, một trong các bảng màu thứ hai 192 lần lượt liên kết các giá trị chỉ số 1, 2, và 3 với các giá trị điểm ảnh A, B, và D. Trong ví dụ này, bộ mã hóa dữ liệu video 20 có thể mã hóa một hoặc nhiều phần tử cú pháp biểu thị mục nhập nào của các bảng màu thứ nhất 184 được bao gồm trong các bảng màu thứ hai 192. Trong ví dụ trên Fig.4, một hoặc nhiều phần tử cú pháp được minh họa là vector 216. Vector 216 có số lượng bin kết hợp (hoặc các bit), trong đó mỗi bin biểu thị liệu bộ dự báo bảng màu gắn với bin đó có được sử dụng để dự báo mục nhập của bảng màu hiện thời hay không. Ví dụ, vector 216 biểu thị rằng hai mục nhập đầu tiên của các bảng màu thứ nhất 184 (202 và 204) được bao gồm trong các bảng màu thứ hai 192 (giá trị “1” trong vector 216), trong khi mục nhập thứ ba của các bảng màu thứ nhất 184 không được bao gồm trong các bảng màu thứ hai 192 (giá trị “0” trong vector 216). Trong ví dụ trên Fig.4, vector là vector Boolean.

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể xác định danh sách bộ dự báo bảng màu (mà còn có thể được gọi là bảng dự báo bảng màu) khi thực hiện dự báo bảng màu. Danh sách bộ dự báo bảng màu có thể bao gồm các mục nhập từ các bảng màu của một hoặc nhiều khối lân cận mà được sử dụng

để dự báo một hoặc nhiều mục nhập của bảng màu để mã hóa khối hiện thời. Bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể xây dựng danh sách theo cùng một cách. Bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể mã hóa dữ liệu (ví dụ như vector 216) để biểu thị mục nhập nào của danh sách bộ dự báo bảng màu được bao gồm trong bảng màu để mã hóa khối hiện thời.

Fig.5 là sơ đồ khái niệm thể hiện ví dụ về việc xác định các chỉ số theo bảng màu cho khối điểm ảnh, theo các kỹ thuật của sáng chế. Ví dụ, Fig.5 bao gồm khối chỉ số 240 (mà còn có thể được gọi là giá trị ánh xạ 240 hoặc giá trị ánh xạ chỉ số 240) bao gồm các giá trị chỉ số (ví dụ, các giá trị chỉ số 1, 2, và 3) liên kết các vị trí tương ứng của các điểm ảnh gắn với các giá trị chỉ số với mục nhập trong các bảng màu 244.

Trong khi khối chỉ số 240 được minh họa trong ví dụ trên Fig.5 bao gồm giá trị chỉ số cho mỗi vị trí điểm ảnh, cần hiểu rằng trong các ví dụ khác, không phải tất cả các vị trí điểm ảnh đều có thể được kết hợp với giá trị chỉ số liên kết giá trị điểm ảnh với mục nhập của các bảng màu 244. Tức là, như được đề cập trên đây, theo một số ví dụ, bộ mã hóa dữ liệu video 20 có thể mã hóa (và bộ giải mã dữ liệu video 30 có thể thu được, từ dòng bit mã hóa) chỉ báo của giá trị điểm ảnh thực (hoặc phiên bản lượng tử hóa của nó) cho vị trí trong khối chỉ số 240 nếu giá trị điểm ảnh không được bao gồm trong các bảng màu 244.

Theo một số ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để mã hóa ánh xạ bổ sung biểu thị các vị trí điểm ảnh nào được kết hợp với các giá trị chỉ số nào. Ví dụ, giả sử rằng mục nhập (i, j) trong khối chỉ số 240 tương ứng với vị trí (i, j) của CU. Bộ mã hóa dữ liệu video 20 có thể mã hóa một hoặc nhiều phần tử cú pháp cho mỗi mục nhập của khối chỉ số (tức là, mỗi vị trí điểm ảnh) biểu thị liệu mục nhập có giá trị chỉ số kết hợp hay không. Ví dụ, bộ mã hóa dữ liệu video 20 có thể mã hóa cờ có giá trị bằng 1 để biểu thị rằng giá trị điểm ảnh tại vị trí (i, j) trong CU là một trong các giá trị trong các bảng màu 244.

Bộ mã hóa dữ liệu video 20 có thể, trong ví dụ này, cũng mã hóa bảng màu (thể hiện trong ví dụ trên Fig.5 là 244). Trong những trường hợp mà các bảng màu 244 bao gồm một mục nhập duy nhất và giá trị điểm ảnh kết hợp, bộ mã hóa dữ liệu video 20 có thể bỏ qua việc báo hiệu giá trị chỉ số. Bộ mã hóa dữ liệu video 20 có thể mã hóa cờ có giá trị bằng 0 để biểu thị rằng giá trị điểm ảnh tại vị trí (i, j) trong CU không phải là một trong các giá trị trong các bảng màu 244. Theo ví dụ này, bộ mã hóa dữ liệu video

20 cũng có thể mã hóa chỉ báo của giá trị điểm ảnh để bộ giải mã dữ liệu video 30 sử dụng trong việc tái tạo giá trị điểm ảnh. Theo một số ví dụ, giá trị điểm ảnh có thể được mã hóa theo cách bị tổn hao.

Giá trị của điểm ảnh trong một vị trí của CU có thể tạo ra chỉ báo về các giá trị của một hoặc nhiều điểm ảnh khác trong các vị trí khác của CU. Ví dụ, có thể có xác suất tương đối cao là các vị trí điểm ảnh lân cận của CU sẽ có cùng giá trị điểm ảnh hoặc có thể được ánh xạ đến cùng giá trị chỉ số (trong trường hợp mã hóa tổn hao, trong đó nhiều hơn một giá trị điểm ảnh được ánh xạ lên một giá trị chỉ số).

Do đó, bộ mã hóa dữ liệu video 20 có thể mã hóa một hoặc nhiều phần tử cú pháp biểu thị số điểm ảnh hoặc giá trị chỉ số liên tiếp theo trình tự quét cho trước mà có cùng giá trị điểm ảnh hoặc giá trị chỉ số. Như mô tả ở trên, dãy các giá trị chỉ số hoặc điểm ảnh có giá trị tương tự có thể được gọi là loạt trong bản mô tả này. Theo một ví dụ nhằm mục đích minh họa, nếu hai điểm ảnh hoặc chỉ số liên tiếp theo trình tự quét cho trước có giá trị khác nhau, thì loạt bằng 0. Nếu hai điểm ảnh hoặc chỉ số liên tiếp theo trình tự quét cho trước có cùng giá trị nhưng điểm ảnh hoặc chỉ số thứ ba theo trình tự quét có giá trị khác, thì loạt bằng 1. Đối với ba chỉ số hoặc điểm ảnh liên tiếp có cùng giá trị, loạt bằng 2, v.v.. Bộ giải mã dữ liệu video 30 có thể thu được các phần tử cú pháp biểu thị loạt từ dòng bit mã hóa và sử dụng dữ liệu để xác định số lượng vị trí liên tiếp có cùng giá trị điểm ảnh hoặc giá trị chỉ số.

Trong một số ví dụ theo các kỹ thuật của sáng chế, đơn vị mã hóa entropy 118 và đơn vị giải mã entropy 150 có thể được tạo cấu hình để mã hóa entropy khối chỉ số 240. Ví dụ, đơn vị mã hóa entropy 118 và đơn vị giải mã entropy 150 có thể được tạo cấu hình để mã hóa entropy các độ dài loạt (ví dụ, giá trị hoặc mã độ dài loạt) và/hoặc vectơ dự báo bảng màu nhị phân liên quan đến khối chỉ số trong chế độ bảng màu.

Fig.6 là sơ đồ khái niệm thể hiện ví dụ về việc xác định độ dài loạt copyabove lớn nhất, ví dụ, giả sử ví dụ về trình tự quét mảnh, theo các kỹ thuật của sáng chế. Trong ví dụ trên Fig.6, nếu không có các điểm ảnh được bao bởi các đường nét đứt 280 được mã hóa dưới dạng mẫu thoát, độ dài loạt lớn nhất có thể là 35 (tức là số vị trí điểm ảnh không được đổ bóng mờ). Nếu một hoặc nhiều điểm ảnh bên trong các đường nét đứt 280 được mã hóa dưới dạng mẫu thoát, giả sử rằng điểm ảnh được đánh dấu là điểm ảnh thoát (vị trí điểm ảnh có dấu “X”) là điểm ảnh thoát thứ nhất trong các đường nét đứt 280 theo trình tự quét, thì độ dài loạt copyabove được mã hóa lớn nhất

có thể bằng năm.

Trong một số ví dụ, bộ giải mã dữ liệu video 30 chỉ có thể xác định chế độ loạt (ví dụ, chế độ bảng màu trong đó các điểm ảnh được mã hóa) cho các điểm ảnh trong các đường nét đứt 280. Do đó, trong trường hợp xấu nhất, bộ giải mã dữ liệu video 30 thực hiện xác định các điểm ảnh BlockWidth-1. Trong một số ví dụ, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để thực hiện các hạn chế nhất định liên quan đến số điểm ảnh lớn nhất mà chế độ loạt được kiểm tra. Ví dụ, bộ giải mã dữ liệu video 30 chỉ có thể kiểm tra các điểm ảnh bên trong các đường nét đứt 280 nếu các điểm ảnh này nằm trong cùng một hàng với điểm ảnh hiện thời. Bộ giải mã dữ liệu video 30 có thể suy ra rằng tất cả các điểm ảnh khác bên trong các đường nét đứt 280 không được mã hóa dưới dạng các mẫu thoát. Ví dụ trên Fig.6 giả sử trình tự quét mảnh. Tuy nhiên, các kỹ thuật này có thể được áp dụng cho các trình tự quét khác, ví dụ như quét theo chiều dọc, quét qua theo chiều ngang, và quét qua theo chiều dọc.

Theo ví dụ của sáng chế, nếu chế độ loạt hiện thời là 'copyabove', thì các ngữ cảnh của độ dài loạt cho điểm ảnh hiện thời có thể phụ thuộc vào giá trị chỉ số của chỉ số của điểm ảnh lân cận trên so với điểm ảnh hiện thời. Trong ví dụ này, nếu điểm ảnh lân cận trên so với điểm ảnh hiện thời nằm ngoài CU hiện thời, bộ giải mã dữ liệu video giả định rằng chỉ số tương ứng bằng hằng số định trước k . Trong một số ví dụ, $k = 0$.

Trong khi mã hóa/giải mã entropy, bộ mã hóa hoặc bộ giải mã entropy có thể đặt các bit của ký hiệu cần được mã hóa hoặc giải mã vào một hoặc nhiều bin. Các bin có thể biểu thị liệu giá trị của ký hiệu có bằng không hay không. Bộ mã hóa entropy hoặc bộ giải mã entropy có thể sử dụng các giá trị của các bin để điều chỉnh quy trình mã hóa entropy. Trong một số ví dụ, bộ mã hóa hoặc bộ giải mã entropy cũng có thể sử dụng các bin để biểu thị việc các giá trị có lớn hơn giá trị cụ thể hay không, ví dụ, lớn hơn không, lớn hơn một, v.v..

Trong một số ví dụ, nếu chế độ hiện thời là chế độ 'copyabove', thì bin thứ nhất của từ mã độ dài loạt lựa chọn một trong hai ngữ cảnh CABAC ứng viên dựa vào việc liệu mẫu lân cận trên (ví dụ, điểm ảnh) so với mẫu hiện thời (ví dụ, điểm ảnh) có bằng 0 hay không.

Theo ví dụ khác, nếu chế độ hiện thời là chế độ 'copyprevious', thì bin thứ nhất của từ mã độ dài loạt lựa chọn một trong bốn ngữ cảnh CABAC ứng viên dựa vào việc

giá trị chỉ số bằng 0, bằng 1, bằng 2, hoặc lớn hơn 2.

Fig.8 là lưu đồ thể hiện quy trình giải mã dữ liệu video làm ví dụ theo các kỹ thuật của sáng chế. Quy trình trên Fig.8 thường được mô tả là được thực hiện bởi bộ giải mã dữ liệu video 30 nhằm mục đích minh họa, mặc dù nhiều bộ xử lý khác cũng có thể thực hiện quy trình được thể hiện trên Fig.8 này. Trong một số ví dụ, đơn vị giải mã khối 152, đơn vị giải mã dựa trên bảng màu 165, và/hoặc đơn vị giải mã entropy 150 có thể thực hiện một hoặc nhiều quy trình thể hiện trên Fig.8.

Trong ví dụ trên Fig.8, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để nhận, từ dòng bit video mã hóa, khối dữ liệu video được mã hóa trong chế độ bảng màu của hình ảnh (800). Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để nhận, từ dòng bit video mã hóa, thông tin chế độ bảng màu được mã hóa cho khối dữ liệu video được mã hóa trong chế độ bảng màu (802). Trong một số ví dụ, thông tin chế độ bảng màu được mã hóa có thể bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất. Ví dụ, phần tử cú pháp thứ nhất có thể bao gồm `palette_index_idc` hoặc `palette_escape_val`, và các phần tử cú pháp khác với phần tử cú pháp thứ nhất có thể bao gồm phần tử cú pháp `palette_run_msb_id_plus1`. Theo ví dụ khác, phần tử cú pháp thứ nhất có thể là chỉ báo của chỉ số đối với mảng các mục nhập bảng màu hoặc phần tử cú pháp thứ nhất có thể chỉ rõ giá trị mẫu thoát mã hóa được lượng tử hóa cho thành phần màu sắc tương ứng với mẫu thoát. Các phần tử cú pháp khác với phần tử cú pháp thứ nhất có thể bao gồm phần tử cú pháp chỉ rõ chỉ số của bit quan trọng nhất trong dạng biểu diễn nhị phân của biến biểu diễn độ dài loạt và phần tử cú pháp chỉ rõ chế độ kiểu loạt.

Theo ví dụ khác, các phần tử cú pháp khác với phần tử cú pháp thứ nhất có thể là phần tử cú pháp bất kỳ và tất cả các phần tử cú pháp khác với phần tử cú pháp thứ nhất. Như được mô tả ở đây liên quan đến một số ví dụ, các phần tử cú pháp khác với phần tử cú pháp thứ nhất cũng có thể khác với các phần tử cú pháp thứ hai, thứ ba, và/hoặc thứ tư. Trong các ví dụ này, các phần tử cú pháp khác với các phần tử cú pháp thứ nhất, thứ hai, thứ ba và thứ tư có thể là phần tử cú pháp bất kỳ và tất cả các phần tử cú pháp khác với các phần tử cú pháp thứ hai, thứ ba, và/hoặc thứ tư. Trong một số ví dụ, các phần tử cú pháp khác với phần tử cú pháp thứ nhất có thể là phần tử cú pháp bất kỳ và tất cả các phần tử cú pháp không được giải mã trong chế độ bỏ qua và/hoặc không cần được giải mã trong chế độ bỏ qua.

Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã, bằng cách sử dụng chế độ bỏ qua ví dụ như chế độ bỏ qua của quy trình mã hóa CABAC, các thể hiện của phần tử cú pháp thứ nhất trước khi giải mã các phần tử cú pháp khác với phần tử cú pháp thứ nhất bằng cách sử dụng chế độ ngữ cảnh (804). Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã, bằng cách sử dụng chế độ ngữ cảnh ví dụ như chế độ CABAC thông thường (thay vì chế độ bỏ qua), các phần tử cú pháp khác với phần tử cú pháp thứ nhất sau khi giải mã các thể hiện của phần tử cú pháp thứ nhất bằng cách sử dụng chế độ bỏ qua (806). Trong một số ví dụ, các thể hiện của phần tử cú pháp thứ nhất bao gồm tất cả các thể hiện của phần tử cú pháp thứ nhất cho khối dữ liệu video được mã hóa trong chế độ bảng màu. Trong các ví dụ này, tất cả các thể hiện của phần tử cú pháp thứ nhất được giải mã bằng cách sử dụng chế độ bỏ qua trước khi giải mã dữ liệu tiếp theo bất kỳ, ví dụ như các phần tử cú pháp khác với phần tử cú pháp thứ nhất. Nói cách khác, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã, bằng cách sử dụng chế độ ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất sau khi giải mã tất cả các thể hiện của phần tử cú pháp thứ nhất cho khối dữ liệu video được mã hóa trong chế độ bảng màu bằng cách sử dụng chế độ bỏ qua.

Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã khối dữ liệu video được mã hóa trong chế độ bảng màu bằng cách sử dụng các thể hiện được giải mã của phần tử cú pháp thứ nhất và các phần tử cú pháp được giải mã khác với phần tử cú pháp thứ nhất (808). Trong một số ví dụ, các thể hiện của phần tử cú pháp thứ nhất được nhóm lại với nhau sao cho giảm bớt được việc chuyển giữa chế độ bỏ qua và chế độ ngữ cảnh trong khi giải mã khối dữ liệu video được mã hóa trong chế độ bảng màu.

Trong một số ví dụ, thông tin chế độ bảng màu được mã hóa có thể bao gồm phần tử cú pháp thứ hai biểu thị số lượng các thể hiện của phần tử cú pháp thứ nhất (ví dụ, biểu thị việc có bao nhiêu thể hiện của phần tử cú pháp thứ nhất được dùng cho khối dữ liệu video được mã hóa trong chế độ bảng màu). Các phần tử cú pháp khác với phần tử cú pháp thứ nhất cũng có thể khác với phần tử cú pháp thứ hai. Trong các ví dụ này, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã, bằng cách sử dụng chế độ bỏ qua, phần tử cú pháp thứ hai trước khi giải mã các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai. Trong một số ví dụ,

không có thể hiện nào của phần tử cú pháp thứ hai được đan xen giữa hai thể hiện bất kỳ của phần tử cú pháp thứ nhất cho khối dữ liệu video được mã hóa trong chế độ bảng màu. Trong một số ví dụ, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định rằng, sau khi giải mã số lượng thể hiện của phần tử cú pháp thứ nhất bằng số lượng được chỉ báo bởi phần tử cú pháp thứ hai, dữ liệu tiếp theo trong dòng bit video mã hóa sau số lượng thể hiện của phần tử cú pháp thứ nhất tương ứng với các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai. Trong một số ví dụ, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã phần tử cú pháp thứ hai bằng cách sử dụng dạng ghép nối của mã Rice rút gọn và mã Golomb hàm mũ.

Trong một số ví dụ, thông tin chế độ bảng màu được mã hóa có thể bao gồm phần tử cú pháp thứ ba và phần tử cú pháp thứ tư. Trong các ví dụ này, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã phần tử cú pháp thứ ba để xác định giá trị tương ứng với phần tử cú pháp thứ ba chỉ báo liệu khối dữ liệu video được mã hóa trong chế độ bảng màu có bao gồm điểm ảnh thoát hay không. Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã phần tử cú pháp thứ tư để xác định giá trị tương ứng với phần tử cú pháp thứ tư chỉ báo kích thước bảng màu. Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã, dựa vào các giá trị xác định được lần lượt tương ứng với các phần tử cú pháp thứ ba và thứ tư, các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai bằng cách sử dụng chế độ ngữ cảnh sau khi giải mã các thể hiện của phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai bằng cách sử dụng chế độ bỏ qua.

Trong một số ví dụ, thông tin chế độ bảng màu được mã hóa có thể bao gồm phần tử cú pháp khác, và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã phần tử cú pháp khác này để xác định giá trị tương ứng với phần tử cú pháp khác này mà chỉ rõ số lượng giá trị khác biệt mà chỉ số bảng màu có cho khối dữ liệu video được mã hóa trong chế độ bảng màu. Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã, dựa vào giá trị xác định được tương ứng với phần tử cú pháp khác này, các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai bằng cách sử dụng chế độ ngữ cảnh sau khi giải mã các thể hiện của phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai bằng cách sử dụng chế độ bỏ qua.

Theo một số ví dụ, thông tin chế độ bảng màu được mã hóa có thể bao gồm phần tử cú pháp khác, và bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải

mã phần tử cú pháp khác này để xác định giá trị tương ứng với phần tử cú pháp khác này chỉ báo thể hiện cuối cùng của phần tử cú pháp `palette_run_type_flag[ xC ][ yC ]` cho khối dữ liệu video được mã hóa trong chế độ bảng màu.

Trong một số ví dụ, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định khối dữ liệu video được mã hóa có một hoặc nhiều mẫu thoát. Trong các ví dụ này, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã mẫu thoát cuối cùng trong khối dữ liệu video được mã hóa trong số một hoặc nhiều mẫu thoát. Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để suy ra giá trị chỉ số áp dụng cho các mẫu của khối dữ liệu video được mã hóa sau mẫu thoát cuối cùng. Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để giải mã các mẫu của khối dữ liệu video được mã hóa sau mẫu thoát cuối cùng bằng cách sử dụng giá trị chỉ số được suy ra cho mỗi mẫu trong số các mẫu sau mẫu thoát cuối cùng.

Trong một số ví dụ, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định số lượng chỉ số bảng màu thu được. Trong các ví dụ này, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định số lượng chỉ số bảng màu dựa vào số lượng chỉ số bảng màu thu được và số lượng thể hiện của phần tử cú pháp thứ nhất. Bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định giá trị loạt lớn nhất có thể cho khối dữ liệu video được mã hóa dựa vào số lượng chỉ số bảng màu thu được và số lượng thể hiện của phần tử cú pháp thứ nhất. Trong một số ví dụ, bộ giải mã dữ liệu video 30 có thể được tạo cấu hình để xác định giá trị loạt lớn nhất có thể cho khối dữ liệu video được mã hóa theo: $nCbS * nCbS - scanPos - 1 - paletteIndicesLeft$, trong đó $nCbS$ chỉ rõ kích thước của khối dữ liệu video được mã hóa, $scanPos$ chỉ rõ vị trí quét, và $paletteIndicesLeft$ chỉ rõ số lượng chỉ số bảng màu còn lại.

Fig.9 là lưu đồ thể hiện quy trình mã hóa dữ liệu video làm ví dụ theo các kỹ thuật của sáng chế. Quy trình được mô tả trên Fig.9 nói chung được thực hiện bằng bộ mã hóa dữ liệu video 20 nhằm mục đích minh họa, mặc dù rất nhiều bộ xử lý khác cũng có thể thực hiện quy trình thể hiện trên Fig.9. Trong một số ví dụ, đơn vị giải mã khối 100, đơn vị mã hóa dựa trên bảng màu 122, và/hoặc đơn vị mã hóa entropy 118 có thể thực hiện một hoặc nhiều quy trình thể hiện trên Fig.9.

Trong ví dụ trên Fig.9, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để xác định rằng khối dữ liệu video cần được mã hóa trong chế độ bảng màu (900). Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa khối dữ liệu video bằng

cách sử dụng chế độ bảng màu trong dòng bit mã hóa (902). Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để tạo ra thông tin chế độ bảng màu cho khối dữ liệu video (904). Thông tin chế độ bảng màu có thể bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất. Ví dụ, phần tử cú pháp thứ nhất có thể bao gồm `palette_index_idc` hoặc `palette_escape_val`, và các phần tử cú pháp khác với phần tử cú pháp thứ nhất có thể bao gồm phần tử cú pháp `palette_run_msb_id_plus1`. Theo ví dụ khác, phần tử cú pháp thứ nhất có thể là chỉ báo của chỉ số đối với mảng các mục nhập bảng màu hoặc phần tử cú pháp thứ nhất có thể chỉ rõ giá trị mẫu thoát mã hóa được lượng tử hóa cho thành phần màu sắc tương ứng với mẫu thoát. Các phần tử cú pháp khác với phần tử cú pháp thứ nhất có thể bao gồm phần tử cú pháp chỉ rõ chỉ số của bit quan trọng nhất trong dạng biểu diễn nhị phân của biến biểu diễn độ dài loạt và phần tử cú pháp chỉ rõ chế độ kiểu loạt.

Theo ví dụ khác, các phần tử cú pháp khác với phần tử cú pháp thứ nhất có thể là phần tử cú pháp bất kỳ và tất cả các phần tử cú pháp khác với phần tử cú pháp thứ nhất. Như được mô tả ở đây liên quan đến một số ví dụ, các phần tử cú pháp khác với phần tử cú pháp thứ nhất cũng có thể khác với các phần tử cú pháp thứ hai, thứ ba, và/hoặc thứ tư. Trong các ví dụ này, các phần tử cú pháp khác với các phần tử cú pháp thứ nhất, thứ hai, thứ ba và thứ tư có thể là phần tử cú pháp bất kỳ và tất cả các phần tử cú pháp khác với các phần tử cú pháp thứ hai, thứ ba, và/hoặc thứ tư. Trong một số ví dụ, các phần tử cú pháp khác với phần tử cú pháp thứ nhất có thể là phần tử cú pháp bất kỳ và tất cả các phần tử cú pháp không được mã hóa trong chế độ bỏ qua và/hoặc không cần được mã hóa trong chế độ bỏ qua.

Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa, bằng cách sử dụng chế độ bỏ qua ví dụ như chế độ bỏ qua của quy trình mã hóa CABAC, các thể hiện của phần tử cú pháp thứ nhất trong dòng bit mã hóa trước khi mã hóa các phần tử cú pháp khác với phần tử cú pháp thứ nhất trong dòng bit mã hóa bằng cách sử dụng chế độ ngưỡng cảnh (906). Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa, bằng cách sử dụng chế độ ngưỡng cảnh ví dụ như chế độ dựa trên ngưỡng cảnh CABAC thông thường, các phần tử cú pháp khác với phần tử cú pháp thứ nhất trong dòng bit mã hóa sau khi mã hóa các thể hiện của phần tử cú pháp thứ nhất bằng cách sử dụng chế độ bỏ qua trong dòng bit mã hóa (908). Trong một số ví dụ, các thể hiện của phần

tử cú pháp thứ nhất được nhóm với nhau để giảm bớt việc chuyển giữa chế độ bỏ qua và chế độ ngữ cảnh trong khi mã hóa khối dữ liệu video được mã hóa trong chế độ bảng màu.

Trong một số ví dụ, các thể hiện của phần tử cú pháp thứ nhất bao gồm tất cả các thể hiện của phần tử cú pháp thứ nhất cho khối dữ liệu video. Trong các ví dụ này, tất cả các thể hiện của phần tử cú pháp thứ nhất được mã hóa bằng cách sử dụng chế độ bỏ qua trước khi mã hóa dữ liệu tiếp theo bất kỳ, ví dụ như các phần tử cú pháp khác với phần tử cú pháp thứ nhất. Nói cách khác, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa, bằng cách sử dụng chế độ ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất sau khi mã hóa tất cả các thể hiện của phần tử cú pháp thứ nhất cho khối dữ liệu video bằng cách sử dụng chế độ bỏ qua.

Trong một số ví dụ, thông tin chế độ bảng màu có thể bao gồm phần tử cú pháp thứ hai biểu thị số lượng thể hiện của phần tử cú pháp thứ nhất (ví dụ, biểu thị việc có bao nhiêu thể hiện của phần tử cú pháp thứ nhất được dùng cho khối dữ liệu video). Các phần tử cú pháp khác với phần tử cú pháp thứ nhất cũng có thể khác với phần tử cú pháp thứ hai. Trong các ví dụ này, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa, bằng cách sử dụng chế độ bỏ qua, phần tử cú pháp thứ hai trong dòng bit mã hóa trước khi mã hóa các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai. Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa các thể hiện của phần tử cú pháp thứ nhất sao cho không có thể hiện nào của phần tử cú pháp thứ hai được đan xen giữa hai thể hiện bất kỳ của phần tử cú pháp thứ nhất cho khối dữ liệu video được mã hóa trong chế độ bảng màu trong dòng bit mã hóa. Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa phần tử cú pháp thứ hai trong dòng bit mã hóa sau các thể hiện được mã hóa của phần tử cú pháp thứ nhất trong dòng bit mã hóa. Ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa trước tiên tất cả các thể hiện của phần tử cú pháp thứ nhất, và sau đó mã hóa phần tử cú pháp thứ hai trong dòng bit mã hóa. Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa phần tử cú pháp thứ hai bằng cách sử dụng dạng ghép nối của mã Rice rút gọn và mã Golomb hàm mũ.

Trong một số ví dụ, thông tin chế độ bảng màu có thể bao gồm phần tử cú pháp thứ ba và phần tử cú pháp thứ tư. Trong các ví dụ này, bộ mã hóa dữ liệu video 20 có

thể được tạo cấu hình để mã hóa giá trị tương ứng với phần tử cú pháp thứ ba chỉ báo liệu khối dữ liệu video có bao gồm điểm ảnh thoát trong dòng bit mã hóa hay không. Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa giá trị tương ứng với phần tử cú pháp thứ tư chỉ báo kích thước bảng màu trong dòng bit mã hóa. Trong một số ví dụ, thông tin chế độ bảng màu có thể bao gồm phần tử cú pháp khác, và bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa giá trị tương ứng với phần tử cú pháp khác này mà chỉ rõ số lượng giá trị khác biệt mà chỉ số bảng màu có cho khối dữ liệu video trong dòng bit mã hóa.

Trong một số ví dụ, thông tin được mã hóa trong chế độ bảng màu có thể chứa phần tử cú pháp khác, và bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa giá trị tương ứng với phần tử cú pháp khác này chỉ báo thể hiện cuối cùng của phần tử cú pháp `palette_run_type_flag[ xC ][ yC ]` cho khối dữ liệu video.

Trong một số ví dụ, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa mẫu thoát cuối cùng trong khối dữ liệu video trong số một hoặc nhiều mẫu thoát. Trong các ví dụ này, bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để suy ra giá trị chỉ số áp dụng cho các mẫu của khối dữ liệu video sau mẫu thoát cuối cùng. Bộ mã hóa dữ liệu video 20 có thể được tạo cấu hình để mã hóa các mẫu của khối dữ liệu video sau mẫu thoát cuối cùng bằng cách sử dụng giá trị chỉ số được suy ra cho mỗi mẫu trong số các mẫu sau mẫu thoát cuối cùng.

Cần hiểu rằng tất cả các kỹ thuật được mô tả ở đây có thể được sử dụng riêng hoặc kết hợp. Ví dụ, bộ mã hóa dữ liệu video 20 và/hoặc một hoặc nhiều thành phần của bộ mã hóa và bộ giải mã dữ liệu video 30 và/hoặc một hoặc nhiều thành phần của bộ giải mã có thể thực hiện các kỹ thuật được mô tả trong bản mô tả này ở dạng kết hợp bất kỳ.

Cần hiểu rằng tùy thuộc vào phương án làm ví dụ, một số thao tác hoặc sự kiện của kỹ thuật bất kỳ trong số các kỹ thuật nêu trong sáng chế được mô tả ở đây có thể được thực hiện theo một trình tự khác, có thể được bổ sung vào, được kết hợp với nhau hoặc được loại bỏ bớt (ví dụ, không phải tất cả các thao tác hoặc sự kiện được mô tả đều cần thiết để thực hiện các kỹ thuật này). Ngoài ra, trong một số ví dụ, các thao tác hoặc sự kiện có thể được thực hiện đồng thời, ví dụ, bằng quy trình xử lý đa chuỗi, xử lý ngắt, hoặc đa xử lý, chứ không phải chỉ thực hiện tuần tự. Ngoài ra, mặc dù một số khía cạnh nhất định của sáng chế được mô tả là được thực hiện bởi một

mô đun hoặc bộ phận duy nhất để cho dễ hiểu, nhưng cần phải hiểu rằng các kỹ thuật trong bản mô tả này có thể được thực hiện bởi tổ hợp của các bộ phận hoặc mô đun kết hợp với bộ mã hóa dữ liệu video.

Các khía cạnh nhất định của sáng chế đã được mô tả, có liên quan đến chuẩn HEVC đang phát triển nhằm mục đích minh họa. Tuy nhiên, các kỹ thuật được mô tả trong bản mô tả này có thể hữu ích cho các quy trình mã hóa dữ liệu video khác, bao gồm các quy trình mã hóa dữ liệu video tiêu chuẩn hoặc độc quyền vẫn chưa được phát triển.

Các kỹ thuật được mô tả ở trên có thể được thực hiện bởi bộ mã hóa dữ liệu video 20 (Fig.1 và Fig.2) và/hoặc bộ giải mã dữ liệu video 30 (các Fig.1 và Fig.3), cả hai được gọi chung là bộ mã hóa dữ liệu video. Tương tự, mã hóa dữ liệu video có thể chỉ mã hóa dữ liệu video hoặc giải mã dữ liệu video, tùy trường hợp.

Theo bản mô tả này, thuật ngữ "hoặc" có thể được hiểu là "và/hoặc" nếu ngữ cảnh không làm hiểu khác. Ngoài ra, trong khi các cụm từ như "một hoặc nhiều" hoặc "ít nhất một" hoặc tương tự có thể được sử dụng cho một số dấu hiệu được mô tả ở đây nhưng không phải là các dấu hiệu khác; các dấu hiệu trong đó ngôn ngữ như vậy không được sử dụng có thể được hiểu có nghĩa bao hàm nếu ngữ cảnh không làm hiểu khác.

Mặc dù các kết hợp cụ thể theo các khía cạnh khác nhau của các kỹ thuật được mô tả trên đây, nhưng các kết hợp này được đưa ra chỉ để minh họa các ví dụ của các kỹ thuật được mô tả trong bản mô tả này. Do đó, các kỹ thuật của sáng chế không nên giới hạn ở các kết hợp làm ví dụ này và có thể bao gồm kết hợp có thể suy đoán bất kỳ của các kỹ thuật theo các khía cạnh khác nhau được mô tả trong bản mô tả này.

Theo một hoặc nhiều phương án làm ví dụ, các chức năng được mô tả có thể được thực hiện bằng phần cứng, phần mềm, phần sụn, hoặc tổ hợp bất kỳ của chúng. Nếu được thực hiện bằng phần mềm, thì các chức năng này có thể được lưu trữ trên hoặc truyền qua, dưới dạng một hoặc nhiều lệnh hoặc mã, vật ghi đọc được bởi máy tính và được thực thi bởi bộ phận xử lý dựa trên phần cứng. Vật ghi đọc được bằng máy tính có thể là phương tiện lưu trữ đọc được bằng máy tính, tương ứng với phương tiện lưu trữ hữu hình chẳng hạn như phương tiện lưu trữ dữ liệu, hoặc phương tiện truyền thông bao gồm mọi phương tiện tạo điều kiện thuận lợi cho việc truyền chương trình máy tính từ nơi này sang nơi khác, ví dụ theo giao thức truyền thông. Theo cách

này, vật ghi đọc được bằng máy tính thông thường có thể tương ứng với (1) phương tiện lưu trữ hữu hình đọc được bằng máy tính thuộc loại không khả biến hoặc (2) phương tiện truyền thông như tín hiệu hoặc sóng mang. Phương tiện lưu trữ dữ liệu có thể là phương tiện khả dụng bất kỳ mà có thể được truy cập bởi một hoặc nhiều máy tính hoặc một hoặc nhiều bộ xử lý để truy hồi các lệnh, mã và/hoặc các cấu trúc dữ liệu để thực hiện các kỹ thuật được mô tả trong sáng chế. Sản phẩm chương trình máy tính có thể bao gồm vật ghi đọc được bằng máy tính.

Ví dụ, và không giới hạn, các phương tiện lưu trữ đọc được bởi máy tính như vậy có thể bao gồm RAM, ROM, EEPROM, CD-ROM hoặc bộ nhớ đĩa quang, bộ nhớ đĩa từ khác, hoặc các thiết bị lưu trữ từ tính khác, bộ nhớ nhanh, hoặc phương tiện khác bất kỳ mà có thể được sử dụng để lưu trữ mã chương trình mong muốn ở dạng các lệnh hoặc cấu trúc dữ liệu và có thể được truy cập bởi máy tính. Hơn nữa, kết nối bất kỳ được gọi đúng là vật ghi đọc được bởi máy tính. Ví dụ, nếu các lệnh được truyền từ trang web, máy chủ hoặc nguồn từ xa khác bằng cách sử dụng cáp đồng trục, cáp sợi quang, cáp dây xoắn, đường thuê bao số (DSL - digital subscriber line), hoặc công nghệ không dây như hồng ngoại, vô tuyến và vi sóng, thì cáp đồng trục, cáp sợi quang, cáp dây xoắn, DSL, hoặc công nghệ không dây như hồng ngoại, vô tuyến và vi sóng cũng nằm trong định nghĩa về vật ghi. Tuy nhiên, cần hiểu rằng phương tiện lưu trữ đọc được bởi máy tính và phương tiện lưu trữ dữ liệu không bao gồm các kết nối, sóng mang, tín hiệu, hoặc các phương tiện chuyển tiếp khác, nhưng thay vào đó là hướng đến các phương tiện lưu trữ hữu hình, không chuyển tiếp. Đĩa từ và đĩa quang, như được sử dụng ở đây, bao gồm đĩa nén (Compact Disc - CD), đĩa laze, đĩa quang, đĩa kỹ thuật số đa năng (Digital Versatile Disc - DVD), đĩa mềm và đĩa Blu-ray, trong đó đĩa từ thường sao chép dữ liệu theo cách từ tính, trong khi đó đĩa quang sao chép dữ liệu theo cách quang học bằng các tia laze. Kết hợp của cả hai loại này cũng được bao gồm trong phạm vi của vật ghi đọc được bằng máy tính.

Các lệnh có thể được thực thi bởi một hoặc nhiều bộ xử lý, chẳng hạn như một hoặc nhiều bộ xử lý tín hiệu số (digital signal processor - DSP), các bộ vi xử lý đa năng, mạch tích hợp chuyên dụng (application specific integrated circuit - ASIC), mảng cổng lập trình được dạng trường (field programmable gate array- FPGA), hoặc các hệ mạch logic tích hợp hoặc rời rạc tương đương khác. Do vậy, thuật ngữ “bộ xử lý” như được sử dụng ở đây có thể chỉ cấu trúc bất kỳ trong cấu trúc nêu trên hoặc cấu

trúc khác bất kỳ phù hợp cho việc thực hiện các kỹ thuật được mô tả ở đây. Ngoài ra, theo một số khía cạnh, chức năng được mô tả ở đây có thể được tạo ra trong các môđun phần cứng và/hoặc phần mềm chuyên dụng được tạo cấu hình để mã hóa và giải mã, hoặc được đưa vào trong bộ mã hóa-giải mã kết hợp. Ngoài ra, các kỹ thuật này có thể được thực hiện đầy đủ trong một hoặc nhiều mạch hoặc phần tử logic.

Các kỹ thuật của sáng chế có thể được thực hiện trong nhiều loại cơ cấu hoặc thiết bị khác nhau, bao gồm máy cầm tay không dây, mạch tích hợp (integrated circuit - IC) hoặc bộ IC (ví dụ, bộ chip). Các thành phần, môđun hoặc đơn vị khác nhau được mô tả trong bản mô tả này để nhấn mạnh các khía cạnh chức năng của thiết bị được tạo cấu hình để thực hiện các kỹ thuật được bộc lộ, nhưng không nhất thiết đòi hỏi việc thực hiện bởi các đơn vị phần cứng khác nhau. Thay vào đó, như được mô tả trên đây, các đơn vị khác nhau có thể được kết hợp trong đơn vị phần cứng mã hóa-giải mã hoặc được tạo ra bởi tập hợp các đơn vị phần cứng liên kết hoạt động, bao gồm một hoặc nhiều bộ xử lý như được mô tả trên đây, kết hợp với phần mềm và/hoặc phần sụn thích hợp.

Các ví dụ khác nhau đã được mô tả ở đây. Sự kết hợp bất kỳ của các hệ thống, thao tác, chức năng hoặc ví dụ mô tả trên đây đều được dự tính. Các ví dụ này và các ví dụ khác đều nằm trong phạm vi của yêu cầu bảo hộ dưới đây.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã dữ liệu video, phương pháp này bao gồm các bước:

nhận, từ dòng bit video mã hóa, khối dữ liệu video được mã hóa trong chế độ bảng màu của hình ảnh;

nhận, từ dòng bit video mã hóa, thông tin chế độ bảng màu được mã hóa cho khối dữ liệu video được mã hóa trong chế độ bảng màu, trong đó thông tin chế độ bảng màu được mã hóa bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất, trong đó phần tử cú pháp thứ nhất là chỉ báo của chỉ số đối với mảng các mục nhập bảng màu hoặc chỉ rõ giá trị mẫu thoát mã hóa được lượng tử hóa cho thành phần màu sắc tương ứng với mẫu thoát, và trong đó các phần tử cú pháp khác với phần tử cú pháp thứ nhất bao gồm phần tử cú pháp mà chỉ rõ chỉ số của bit quan trọng nhất trong dạng biểu diễn nhị phân của biến biểu diễn độ dài loạt và phần tử cú pháp chỉ rõ chế độ kiểu loạt;

phân tích cú pháp các thể hiện của phần tử cú pháp thứ nhất từ dòng bit video mã hóa trước khi phân tích cú pháp các phần tử cú pháp khác với phần tử cú pháp thứ nhất từ dòng bit video mã hóa;

phân tích cú pháp các phần tử cú pháp khác với phần tử cú pháp thứ nhất từ dòng bit video mã hóa sau khi phân tích cú pháp các thể hiện của phần tử cú pháp thứ nhất từ dòng bit video mã hóa;

giải mã, bằng cách sử dụng chế độ bỏ qua mã hóa số học nhị phân thích ứng theo ngữ cảnh, các thể hiện của phần tử cú pháp thứ nhất;

giải mã, bằng cách sử dụng chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất; và

giải mã khối dữ liệu video được mã hóa trong chế độ bảng màu bằng cách sử dụng các thể hiện được giải mã của phần tử cú pháp thứ nhất và các phần tử cú pháp được giải mã khác với phần tử cú pháp thứ nhất.

2. Phương pháp theo điểm 1, trong đó các thể hiện của phần tử cú pháp thứ nhất bao gồm tất cả các thể hiện của phần tử cú pháp thứ nhất cho khối dữ liệu video được mã hóa trong chế độ bảng màu.

3. Phương pháp theo điểm 1, trong đó phần tử cú pháp thứ nhất là phần tử cú pháp

palette_index_idc biểu diễn chỉ số của mục nhập bảng màu tương ứng của bảng màu hoặc phần tử cú pháp palette_escape_val biểu diễn giá trị chỉ số dành riêng cho các mẫu thoát mà được biểu diễn bằng thông tin màu không được biểu diễn bằng mục nhập bất kỳ trong số các mục nhập bảng màu của bảng màu, và trong đó các phần tử cú pháp khác với phần tử cú pháp thứ nhất bao gồm phần tử cú pháp palette_run_msb_id_plus1 chỉ báo độ dài loạt của tập con các chỉ số bảng màu của bảng màu.

4. Phương pháp theo điểm 1, trong đó các thể hiện của phần tử cú pháp thứ nhất được nhóm với nhau ở phần đầu của dữ liệu biểu diễn khối được mã hóa trong chế độ bảng màu, sao cho việc chuyển từ giải mã bằng cách sử dụng chế độ bỏ qua sang giải mã bằng cách sử dụng chế độ ngữ cảnh diễn ra chính xác một lần để tái tạo khối được mã hóa trong chế độ bảng màu.

5. Phương pháp theo điểm 1, trong đó thông tin chế độ bảng màu được mã hóa bao gồm phần tử cú pháp thứ hai chỉ báo số lượng thể hiện của phần tử cú pháp thứ nhất trong dữ liệu biểu diễn khối được mã hóa trong chế độ bảng màu trong dòng bit video mã hóa, trong đó các phần tử cú pháp mà khác với phần tử cú pháp thứ nhất cũng khác với phần tử cú pháp thứ hai, và trong đó phương pháp này còn bao gồm các bước:

phân tích cú pháp phần tử cú pháp thứ hai từ dòng bit video mã hóa trước khi phân tích cú pháp các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai từ dòng bit video mã hóa; và

giải mã, bằng cách sử dụng chế độ bỏ qua mã hóa số học nhị phân thích ứng theo ngữ cảnh, phần tử cú pháp thứ hai.

6. Phương pháp theo điểm 5, trong đó không có thể hiện nào của phần tử cú pháp thứ hai được đan xen giữa hai thể hiện bất kỳ của phần tử cú pháp thứ nhất cho khối dữ liệu video được mã hóa trong chế độ bảng màu.

7. Phương pháp theo điểm 5, trong đó phương pháp này còn bao gồm bước:

xác định rằng, sau khi giải mã số lượng thể hiện của phần tử cú pháp thứ nhất bằng số lượng được chỉ báo bởi phần tử cú pháp thứ hai, dữ liệu tiếp theo trong dòng

bit video mã hóa sau số lượng thể hiện của phần tử cú pháp thứ nhất tương ứng với các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai.

8. Phương pháp theo điểm 5, trong đó thông tin chế độ bảng màu được mã hóa bao gồm phần tử cú pháp thứ ba và phần tử cú pháp thứ tư, trong đó phương pháp này còn bao gồm các bước:

giải mã phần tử cú pháp thứ ba để xác định giá trị tương ứng với phần tử cú pháp thứ ba chỉ báo liệu khối dữ liệu video được mã hóa trong chế độ bảng màu có bao gồm mẫu thoát hay không;

giải mã phần tử cú pháp thứ tư để xác định giá trị tương ứng với phần tử cú pháp thứ tư chỉ báo kích thước bảng màu; và

giải mã, dựa vào các giá trị xác định được lần lượt tương ứng với các phần tử cú pháp thứ ba và thứ tư, các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai bằng cách sử dụng chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh.

9. Phương pháp theo điểm 5, trong đó thông tin chế độ bảng màu được mã hóa bao gồm phần tử cú pháp thứ ba, trong đó phương pháp này còn bao gồm các bước:

giải mã phần tử cú pháp thứ ba để xác định giá trị tương ứng với phần tử cú pháp thứ ba chỉ rõ số lượng giá trị khác biệt mà chỉ số bảng màu có cho khối dữ liệu video được mã hóa trong chế độ bảng màu; và

giải mã, dựa vào giá trị xác định được tương ứng với phần tử cú pháp thứ ba, các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai bằng cách sử dụng chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh.

10. Phương pháp theo điểm 5, trong đó thông tin chế độ bảng màu được mã hóa bao gồm phần tử cú pháp thứ ba, trong đó phương pháp này còn bao gồm bước:

giải mã phần tử cú pháp thứ ba để xác định giá trị tương ứng với phần tử cú pháp thứ ba chỉ báo thể hiện cuối cùng của phần tử cú pháp `palette_run_type_flag[xC][yC]` cho khối dữ liệu video được mã hóa trong chế độ bảng màu.

11. Phương pháp theo điểm 5, trong đó phương pháp này còn bao gồm bước:

giải mã phần tử cú pháp thứ hai bằng cách sử dụng dạng ghép nối của mã Rice rút gọn và mã Golomb hàm mũ.

12. Phương pháp theo điểm 5, trong đó phương pháp này còn bao gồm các bước:

xác định số lượng chỉ số bảng màu nhận được;

xác định số lượng chỉ số bảng màu còn lại, dựa vào số lượng chỉ số bảng màu nhận được và số lượng thể hiện của phần tử cú pháp thứ nhất; và

xác định rằng giá trị cho biến biểu diễn độ dài loạt bằng giá trị loạt lớn nhất có thể cho khối dữ liệu video được mã hóa trong chế độ bảng màu dựa vào số lượng chỉ số bảng màu thu được và số lượng thể hiện của phần tử cú pháp thứ nhất.

13. Phương pháp theo điểm 12,

trong đó giá trị loạt lớn nhất có thể cho khối dữ liệu video được mã hóa trong chế độ bảng màu bằng: $nCbS * nCbS - scanPos - 1 - paletteIndicesLeft$, trong đó $nCbS$ chỉ rõ kích thước của khối dữ liệu video được mã hóa trong chế độ bảng màu, $scanPos$ chỉ rõ vị trí quét, và $paletteIndicesLeft$ chỉ rõ số lượng chỉ số bảng màu còn lại.

14. Phương pháp theo điểm 1, trong đó phương pháp này còn bao gồm các bước:

xác định rằng khối dữ liệu video được mã hóa trong chế độ bảng màu có một hoặc nhiều mẫu thoát;

giải mã mẫu thoát cuối cùng trong khối dữ liệu video được mã hóa trong chế độ bảng màu trong số một hoặc nhiều mẫu thoát;

suy ra giá trị chỉ số mà áp dụng cho các mẫu của khối dữ liệu video được mã hóa trong chế độ bảng màu sau mẫu thoát cuối cùng; và

giải mã các mẫu của khối dữ liệu video được mã hóa trong chế độ bảng màu sau mẫu thoát cuối cùng bằng cách sử dụng giá trị chỉ số được suy ra cho mỗi mẫu trong số các mẫu sau mẫu thoát cuối cùng.

15. Thiết bị giải mã dữ liệu video, thiết bị này bao gồm:

bộ nhớ được tạo cấu hình để lưu trữ dữ liệu video; và

bộ giải mã dữ liệu video truyền thông với bộ nhớ, bộ giải mã dữ liệu video này

được tạo cấu hình để:

nhận khối dữ liệu video được mã hóa trong chế độ bảng màu của hình ảnh từ bộ nhớ;

nhận thông tin chế độ bảng màu được mã hóa cho khối dữ liệu video được mã hóa trong chế độ bảng màu, trong đó thông tin chế độ bảng màu được mã hóa bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất, trong đó phần tử cú pháp thứ nhất là chỉ báo của chỉ số đối với mảng các mục nhập bảng màu hoặc chỉ rõ giá trị mẫu thoát mã hóa được lượng tử hóa cho thành phần màu sắc tương ứng với mẫu thoát, và trong đó các phần tử cú pháp khác với phần tử cú pháp thứ nhất bao gồm phần tử cú pháp mà chỉ rõ chỉ số của bit quan trọng nhất trong dạng biểu diễn nhị phân của biến số biểu diễn độ dài loạt và phần tử cú pháp chỉ rõ chế độ kiểu loạt;

phân tích cú pháp các thể hiện của phần tử cú pháp thứ nhất từ thông tin chế độ bảng màu được mã hóa trước khi phân tích cú pháp các phần tử cú pháp khác với phần tử cú pháp thứ nhất từ thông tin chế độ bảng màu được mã hóa;

phân tích cú pháp các phần tử cú pháp khác với phần tử cú pháp thứ nhất từ thông tin chế độ bảng màu được mã hóa sau khi phân tích cú pháp các thể hiện của phần tử cú pháp thứ nhất từ thông tin chế độ bảng màu được mã hóa;

giải mã, bằng cách sử dụng chế độ bỏ qua mã hóa số học nhị phân thích ứng theo ngữ cảnh, các thể hiện của phần tử cú pháp thứ nhất;

giải mã, bằng cách sử dụng chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất; và

giải mã khối dữ liệu video được mã hóa trong chế độ bảng màu bằng cách sử dụng các thể hiện được giải mã của phần tử cú pháp thứ nhất và các phần tử cú pháp được giải mã khác với phần tử cú pháp thứ nhất.

16. Thiết bị theo điểm 15, trong đó các thể hiện của phần tử cú pháp thứ nhất bao gồm tất cả các thể hiện của phần tử cú pháp thứ nhất cho khối dữ liệu video được mã hóa trong chế độ bảng màu.

17. Thiết bị theo điểm 15, trong đó phần tử cú pháp thứ nhất là `palette_index_idc` hoặc `palette_escape_val`, và trong đó các phần tử cú pháp khác với phần tử cú pháp thứ nhất bao gồm phần tử cú pháp `palette_run_msb_id_plus1`.

18. Thiết bị theo điểm 15, trong đó các thể hiện của phần tử cú pháp thứ nhất được nhóm với nhau để giảm bớt việc chuyển giữa chế độ bỏ qua mã hóa số học nhị phân thích ứng theo ngữ cảnh và chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh trong khi giải mã khối dữ liệu video được mã hóa trong chế độ bảng màu.

19. Thiết bị theo điểm 15, trong đó thông tin chế độ bảng màu được mã hóa bao gồm phần tử cú pháp thứ hai chỉ báo số lượng thể hiện của phần tử cú pháp thứ nhất, trong đó các phần tử cú pháp khác với phần tử cú pháp thứ nhất cũng khác phần tử cú pháp thứ hai, và trong đó bộ giải mã dữ liệu video còn được tạo cấu hình để:

phân tích cú pháp phần tử cú pháp thứ hai từ thông tin chế độ bảng màu được mã hóa trước khi phân tích cú pháp các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai từ thông tin chế độ bảng màu được mã hóa; và

giải mã, bằng cách sử dụng chế độ bỏ qua mã hóa số học nhị phân thích ứng theo ngữ cảnh, phần tử cú pháp thứ hai.

20. Thiết bị theo điểm 19, trong đó không có thể hiện nào của phần tử cú pháp thứ hai được đan xen giữa hai thể hiện bất kỳ của phần tử cú pháp thứ nhất cho khối dữ liệu video được mã hóa trong chế độ bảng màu.

21. Thiết bị theo điểm 19, trong đó bộ giải mã dữ liệu video còn được tạo cấu hình để xác định rằng, sau khi giải mã số lượng thể hiện của phần tử cú pháp thứ nhất bằng số lượng được chỉ báo bởi phần tử cú pháp thứ hai, dữ liệu tiếp theo trong dòng bit video mã hóa sau số lượng thể hiện của phần tử cú pháp thứ nhất tương ứng với các phần tử cú pháp mà khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai.

22. Thiết bị theo điểm 19, trong đó thông tin chế độ bảng màu được mã hóa bao gồm phần tử cú pháp thứ ba và phần tử cú pháp thứ tư, trong đó bộ giải mã dữ liệu video còn được tạo cấu hình để:

giải mã phần tử cú pháp thứ ba để xác định giá trị tương ứng với phần tử cú pháp thứ ba chỉ báo liệu khối dữ liệu video được mã hóa trong chế độ bảng màu có bao gồm mẫu thoát hay không.

giải mã phần tử cú pháp thứ tư để xác định giá trị tương ứng với phần tử cú pháp thứ tư chỉ báo kích thước bảng màu; và

giải mã, dựa vào các giá trị xác định được lần lượt tương ứng với các phần tử cú pháp thứ ba và thứ tư, các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai bằng cách sử dụng chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh.

23. Thiết bị theo điểm 19, trong đó thông tin chế độ bảng màu được mã hóa bao gồm phần tử cú pháp thứ ba, trong đó bộ giải mã dữ liệu video còn được tạo cấu hình để:

giải mã phần tử cú pháp thứ ba để xác định giá trị tương ứng với phần tử cú pháp thứ ba chỉ rõ số lượng giá trị khác biệt mà chỉ số bảng màu có cho khối dữ liệu video được mã hóa trong chế độ bảng màu; và

giải mã, dựa vào giá trị xác định được tương ứng với phần tử cú pháp thứ ba, các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai bằng cách sử dụng chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh.

24. Thiết bị theo điểm 19, trong đó thông tin chế độ bảng màu được mã hóa bao gồm phần tử cú pháp thứ ba, trong đó bộ giải mã dữ liệu video còn được tạo cấu hình để:

giải mã phần tử cú pháp thứ ba để xác định giá trị tương ứng với phần tử cú pháp thứ ba chỉ báo thể hiện cuối cùng của phần tử cú pháp `palette_run_type_flag[ xC ][ yC ]` cho khối dữ liệu video được mã hóa trong chế độ bảng màu.

25. Thiết bị theo điểm 19, trong đó bộ giải mã dữ liệu video còn được tạo cấu hình để:

giải mã phần tử cú pháp thứ hai bằng cách sử dụng dạng ghép nối của mã Rice rút gọn và mã Golomb hàm mũ.

26. Thiết bị theo điểm 19, trong đó bộ giải mã dữ liệu video còn được tạo cấu hình để:

xác định số lượng chỉ số bảng màu nhận được;

xác định số lượng chỉ số bảng màu còn lại dựa vào số lượng chỉ số bảng màu thu được và số lượng thể hiện của phần tử cú pháp thứ nhất; và

xác định rằng giá trị cho biến số biểu diễn độ dài loạt bằng giá trị loạt lớn nhất có thể cho khối dữ liệu video được mã hóa trong chế độ bảng màu dựa vào số lượng chỉ số bảng màu thu được và số lượng thể hiện của phần tử cú pháp thứ nhất.

27. Thiết bị theo điểm 26, trong đó giá trị loạt lớn nhất có thể cho khối dữ liệu video được mã hóa trong chế độ bảng màu bằng: $nCbS * nCbS - scanPos - 1 - paletteIndicesLeft$, trong đó $nCbS$ chỉ rõ kích thước của khối dữ liệu video được mã hóa trong chế độ bảng màu, $scanPos$ chỉ rõ vị trí quét, và $paletteIndicesLeft$ chỉ rõ số lượng chỉ số bảng màu còn lại.

28. Thiết bị theo điểm 15, trong đó bộ giải mã dữ liệu video còn được tạo cấu hình để:
xác định rằng khối dữ liệu video được mã hóa trong chế độ bảng màu có một hoặc nhiều mẫu thoát;

giải mã mẫu thoát cuối cùng trong khối dữ liệu video được mã hóa trong chế độ bảng màu trong số một hoặc nhiều mẫu thoát;

suy ra giá trị chỉ số áp dụng cho các mẫu của khối dữ liệu video được mã hóa trong chế độ bảng màu sau mẫu thoát cuối cùng; và

giải mã các mẫu của khối dữ liệu video được mã hóa trong chế độ bảng màu sau mẫu thoát cuối cùng bằng cách sử dụng giá trị chỉ số được suy ra cho mỗi mẫu trong số các mẫu sau mẫu thoát cuối cùng.

29. Vật ghi bất biến đọc được bằng máy tính lưu trữ trên đó các lệnh mà, khi được thực thi, khiến cho một hoặc nhiều bộ xử lý:

nhận khối dữ liệu video được mã hóa trong chế độ bảng màu của hình ảnh từ bộ nhớ;

nhận thông tin chế độ bảng màu được mã hóa cho khối dữ liệu video được mã hóa trong chế độ bảng màu, trong đó thông tin chế độ bảng màu được mã hóa bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất, trong đó phần tử cú pháp thứ nhất là chỉ báo của chỉ số đối với mảng các mục nhập bảng màu hoặc chỉ rõ giá trị mẫu thoát mã hóa được lượng tử hóa cho

thành phần màu sắc tương ứng với mẫu thoát, và trong đó các phần tử cú pháp khác với phần tử cú pháp thứ nhất bao gồm phần tử cú pháp mà chỉ rõ chỉ số của bit quan trọng nhất trong dạng biểu diễn nhị phân của biến biểu diễn độ dài loạt và phần tử cú pháp chỉ rõ chế độ kiểu loạt;

phân tích cú pháp các thể hiện của phần tử cú pháp thứ nhất từ thông tin chế độ bảng màu được mã hóa trước khi phân tích cú pháp các phần tử cú pháp khác với phần tử cú pháp thứ nhất từ thông tin chế độ bảng màu được mã hóa;

phân tích cú pháp các phần tử cú pháp khác với phần tử cú pháp thứ nhất từ thông tin chế độ bảng màu được mã hóa sau khi phân tích cú pháp các thể hiện của phần tử cú pháp thứ nhất từ thông tin chế độ bảng màu được mã hóa;

giải mã, bằng cách sử dụng chế độ bỏ qua mã hóa số học nhị phân thích ứng theo ngữ cảnh, các thể hiện của phần tử cú pháp thứ nhất;

giải mã, bằng cách sử dụng chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất; và

giải mã khối dữ liệu video được mã hóa trong chế độ bảng màu bằng cách sử dụng các thể hiện được giải mã của phần tử cú pháp thứ nhất và các phần tử cú pháp được giải mã khác với phần tử cú pháp thứ nhất.

30. Phương pháp mã hóa dữ liệu video, phương pháp này bao gồm các bước:

xác định rằng khối dữ liệu video cần được mã hóa trong chế độ bảng màu;

mã hóa khối dữ liệu video bằng cách sử dụng chế độ bảng màu trong dòng bit mã hóa, trong đó bước mã hóa khối dữ liệu video bằng cách sử dụng chế độ bảng màu bao gồm các bước:

tạo ra thông tin chế độ bảng màu cho khối dữ liệu video, trong đó thông tin chế độ bảng màu bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất, trong đó phần tử cú pháp thứ nhất là chỉ báo của chỉ số đối với mảng gồm các mục nhập bảng màu hoặc chỉ rõ giá trị mẫu thoát mã hóa được lượng tử hóa cho thành phần màu sắc tương ứng với mẫu thoát, và trong đó các phần tử cú pháp khác với phần tử cú pháp thứ nhất bao gồm phần tử cú pháp mà chỉ rõ chỉ số của bit quan trọng nhất trong dạng biểu diễn nhị phân của biến biểu diễn độ dài loạt và phần tử cú pháp chỉ rõ chế độ kiểu loạt;

mã hóa, bằng cách sử dụng chế độ bỏ qua mã hóa số học nhị phân thích ứng

theo ngữ cảnh, các thể hiện của phần tử cú pháp thứ nhất trong dòng bit mã hóa trước khi mã hóa các phần tử cú pháp khác với phần tử cú pháp thứ nhất trong dòng bit mã hóa bằng cách sử dụng chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh; và

mã hóa, bằng cách sử dụng chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất trong dòng bit mã hóa sau khi mã hóa các thể hiện của phần tử cú pháp thứ nhất bằng cách sử dụng chế độ bỏ qua mã hóa số học nhị phân thích ứng theo ngữ cảnh trong dòng bit mã hóa.

31. Phương pháp theo điểm 30, trong đó các thể hiện của phần tử cú pháp thứ nhất bao gồm tất cả các thể hiện của phần tử cú pháp thứ nhất cho khối dữ liệu video.

32. Phương pháp theo điểm 30, trong đó phần tử cú pháp thứ nhất là `palette_index_idc` hoặc `palette_escape_val`, và trong đó các phần tử cú pháp khác với phần tử cú pháp thứ nhất bao gồm phần tử cú pháp `palette_run_msb_id_plus1`.

33. Phương pháp theo điểm 30, trong đó các thể hiện của phần tử cú pháp thứ nhất được nhóm với nhau để giảm việc chuyển giữa chế độ bỏ qua mã hóa số học nhị phân thích ứng theo ngữ cảnh và chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh trong khi mã hóa khối dữ liệu video được mã hóa trong chế độ bảng màu.

34. Phương pháp theo điểm 30, trong đó thông tin chế độ bảng màu bao gồm phần tử cú pháp thứ hai chỉ báo số lượng thể hiện của phần tử cú pháp thứ nhất, trong đó các phần tử cú pháp khác với phần tử cú pháp thứ nhất cũng khác phần tử cú pháp thứ hai, và trong đó phương pháp này còn bao gồm bước:

mã hóa, bằng cách sử dụng chế độ bỏ qua mã hóa số học nhị phân thích ứng theo ngữ cảnh, phần tử cú pháp thứ hai trong dòng bit mã hóa trước khi mã hóa các phần tử cú pháp khác với phần tử cú pháp thứ nhất và phần tử cú pháp thứ hai.

35. Phương pháp theo điểm 34, trong đó không có thể hiện nào của phần tử cú pháp thứ hai được đan xen giữa hai thể hiện bất kỳ của phần tử cú pháp thứ nhất cho khối dữ liệu video.

36. Phương pháp theo điểm 34, phương pháp này còn bao gồm bước:

mã hóa phần tử cú pháp thứ hai trong dòng bit mã hóa sau các thể hiện được mã hóa của phần tử cú pháp thứ nhất trong dòng bit mã hóa.

37. Phương pháp theo điểm 34, trong đó thông tin chế độ bảng màu bao gồm phần tử cú pháp thứ ba và phần tử cú pháp thứ tư, trong đó phương pháp này còn bao gồm các bước:

mã hóa giá trị tương ứng với phần tử cú pháp thứ ba chỉ báo liệu khối dữ liệu video có bao gồm mẫu thoát trong dòng bit mã hóa hay không; và

mã hóa giá trị tương ứng với phần tử cú pháp thứ tư chỉ báo kích thước bảng màu trong dòng bit mã hóa.

38. Phương pháp theo điểm 34, trong đó thông tin chế độ bảng màu bao gồm phần tử cú pháp thứ ba, trong đó phương pháp này còn bao gồm bước:

mã hóa giá trị tương ứng với phần tử cú pháp thứ ba mà chỉ rõ số lượng giá trị khác biệt mà chỉ số bảng màu có cho khối dữ liệu video trong dòng bit mã hóa.

39. Phương pháp theo điểm 34, trong đó thông tin chế độ bảng màu bao gồm phần tử cú pháp thứ ba, trong đó phương pháp này còn bao gồm bước:

mã hóa giá trị tương ứng với phần tử cú pháp thứ ba chỉ báo thể hiện cuối cùng của phần tử cú pháp `palette_run_type_flag[xC][yC]` cho khối dữ liệu video.

40. Phương pháp theo điểm 34, phương pháp này còn bao gồm bước:

mã hóa phần tử cú pháp thứ hai bằng cách sử dụng dạng ghép nối của mã Rice rút gọn và mã Golomb hàm mũ.

41. Phương pháp theo điểm 30, phương pháp này còn bao gồm các bước:

mã hóa mẫu thoát cuối cùng trong khối dữ liệu video trong số một hoặc nhiều mẫu thoát;

suy ra giá trị chỉ số áp dụng cho các mẫu của khối dữ liệu video sau mẫu thoát cuối cùng; và

mã hóa các mẫu của khối dữ liệu video sau mẫu thoát cuối cùng bằng cách sử dụng giá trị chỉ số được suy ra cho mỗi mẫu trong số các mẫu sau mẫu thoát cuối cùng.

42. Thiết bị mã hóa dữ liệu video, thiết bị này bao gồm:

bộ nhớ được tạo cấu hình để lưu trữ dữ liệu video; và

bộ mã hóa dữ liệu video truyền thông với bộ nhớ, bộ mã hóa dữ liệu video này được tạo cấu hình để:

xác định rằng khối dữ liệu video lưu trữ trong bộ nhớ cần được mã hóa trong chế độ bảng màu;

mã hóa khối dữ liệu video bằng cách sử dụng chế độ bảng màu trong dòng bit mã hóa, trong đó bộ mã hóa dữ liệu video được tạo cấu hình để mã hóa khối dữ liệu video bằng cách sử dụng chế độ bảng màu bao gồm bộ mã hóa dữ liệu video được tạo cấu hình để:

tạo ra thông tin chế độ bảng màu cho khối dữ liệu video, trong đó thông tin chế độ bảng màu bao gồm các thể hiện của phần tử cú pháp thứ nhất và các phần tử cú pháp khác với phần tử cú pháp thứ nhất, trong đó phần tử cú pháp thứ nhất là chỉ báo của chỉ số đối với mảng gồm các mục nhập bảng màu hoặc chỉ rõ giá trị mẫu thoát mã hóa được lượng tử hóa cho thành phần màu sắc tương ứng với mẫu thoát, và trong đó các phần tử cú pháp khác với phần tử cú pháp thứ nhất bao gồm phần tử cú pháp mà chỉ rõ chỉ số của bit quan trọng nhất trong dạng biểu diễn nhị phân của biến biểu diễn độ dài loạt và phần tử cú pháp chỉ rõ chế độ kiểu loạt;

mã hóa, bằng cách sử dụng chế độ bỏ qua mã hóa số học nhị phân thích ứng theo ngữ cảnh, các thể hiện của phần tử cú pháp thứ nhất trong dòng bit mã hóa trước khi mã hóa các phần tử cú pháp khác với phần tử cú pháp thứ nhất trong dòng bit mã hóa bằng cách sử dụng chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh; và

mã hóa, bằng cách sử dụng chế độ ngữ cảnh mã hóa số học nhị phân thích ứng theo ngữ cảnh, các phần tử cú pháp khác với phần tử cú pháp thứ nhất trong dòng bit mã hóa sau khi mã hóa các thể hiện của phần tử cú pháp thứ nhất bằng cách sử dụng chế độ bỏ qua mã hóa số học nhị phân thích ứng theo ngữ cảnh trong dòng bit mã hóa.

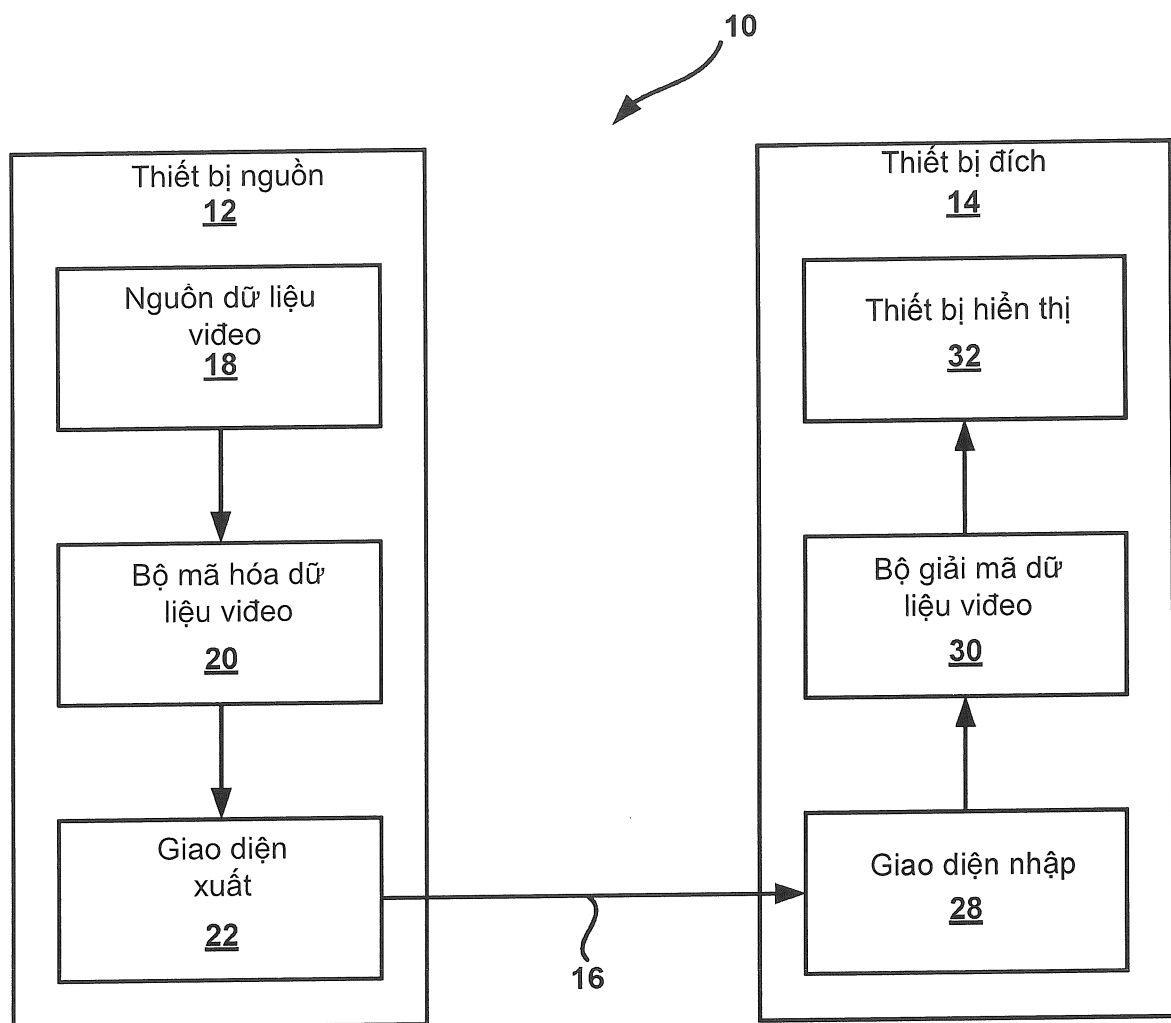


Fig.1

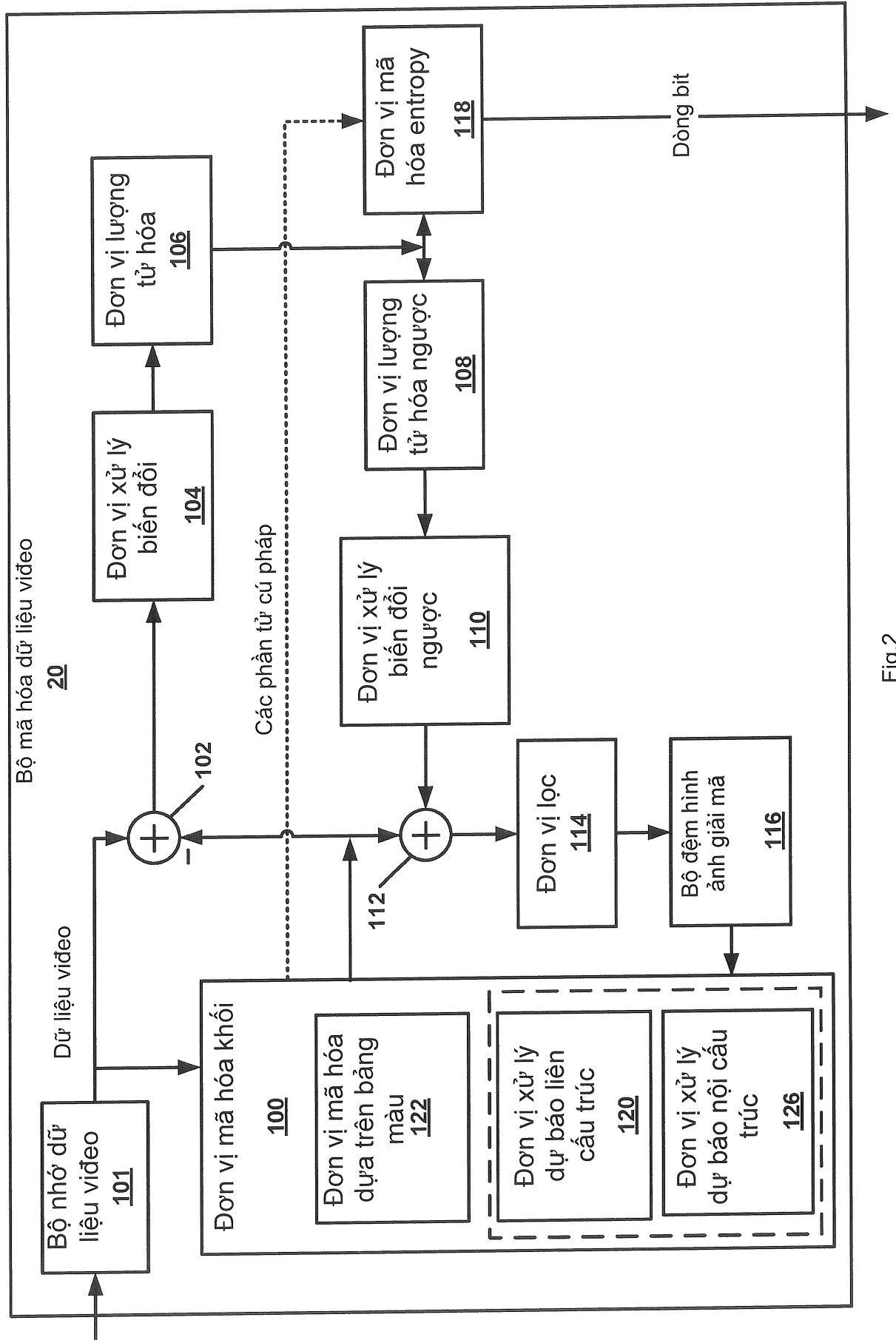


Fig.2

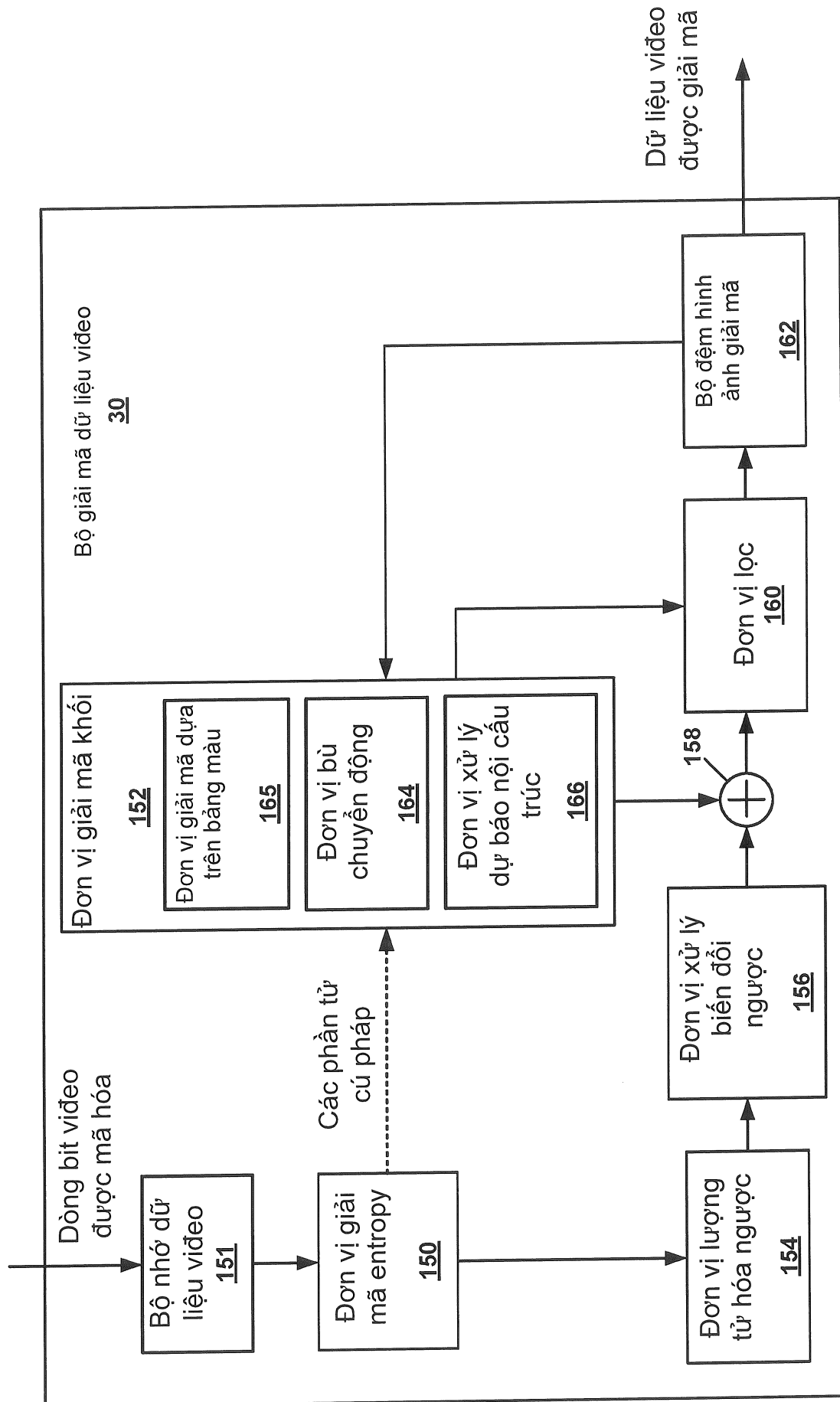


Fig.3

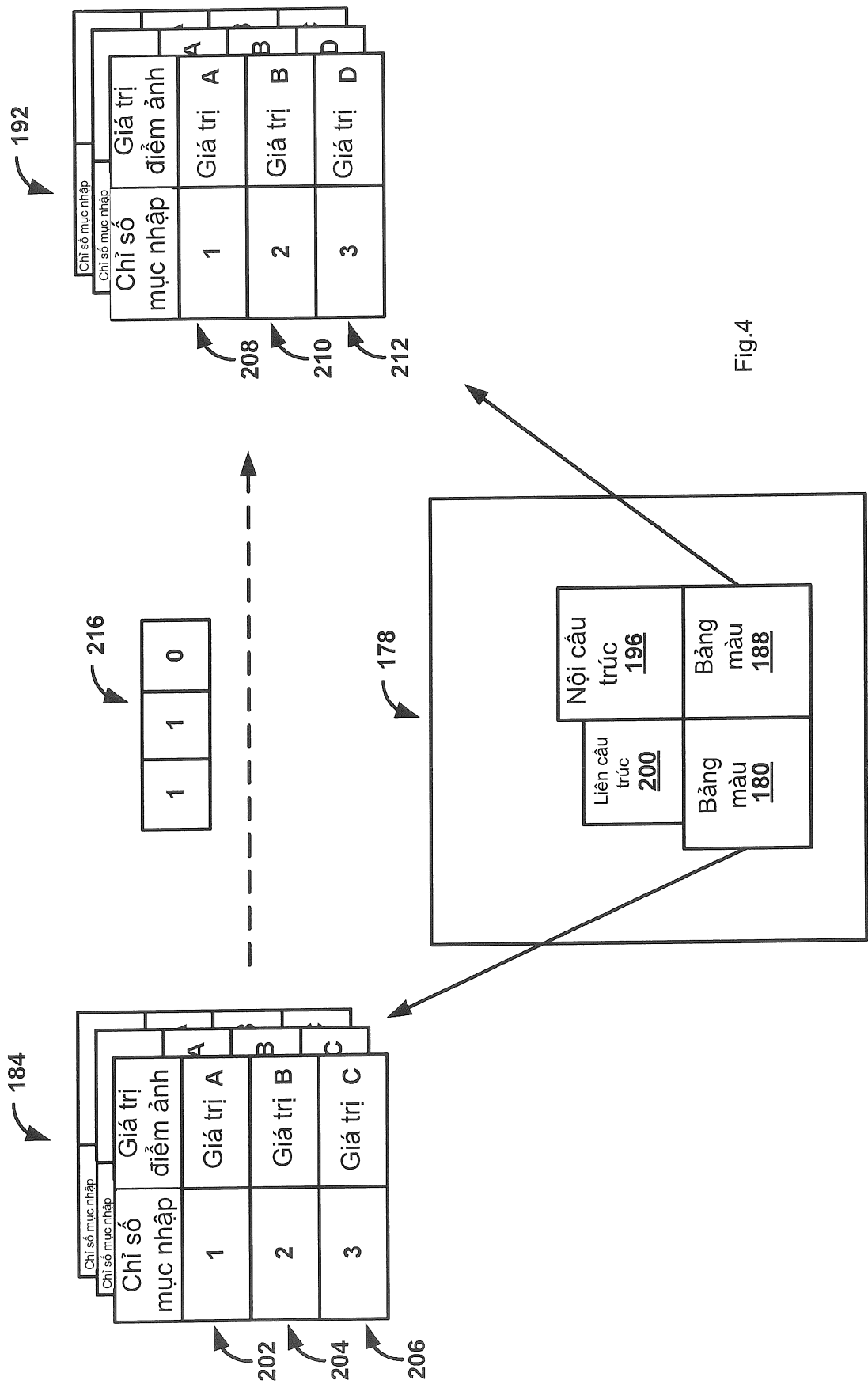


Fig.4

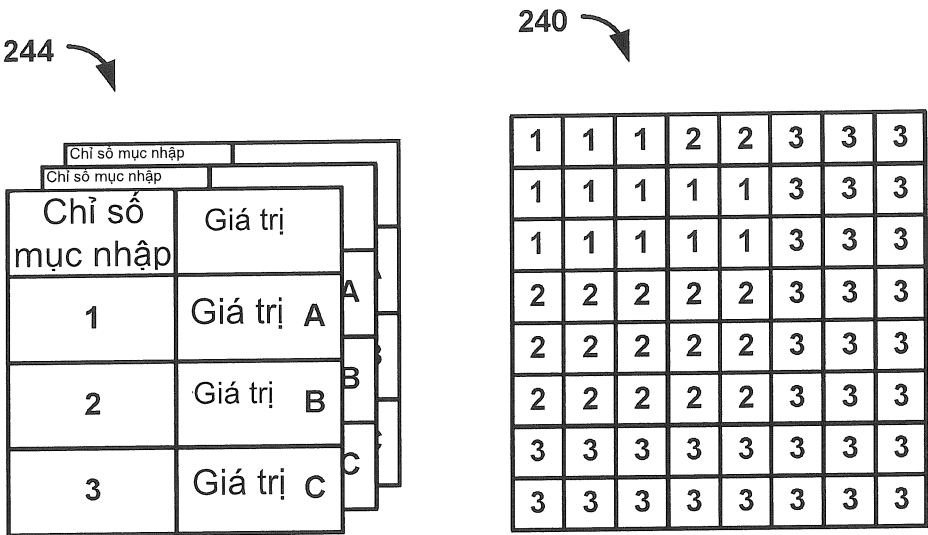


Fig.5

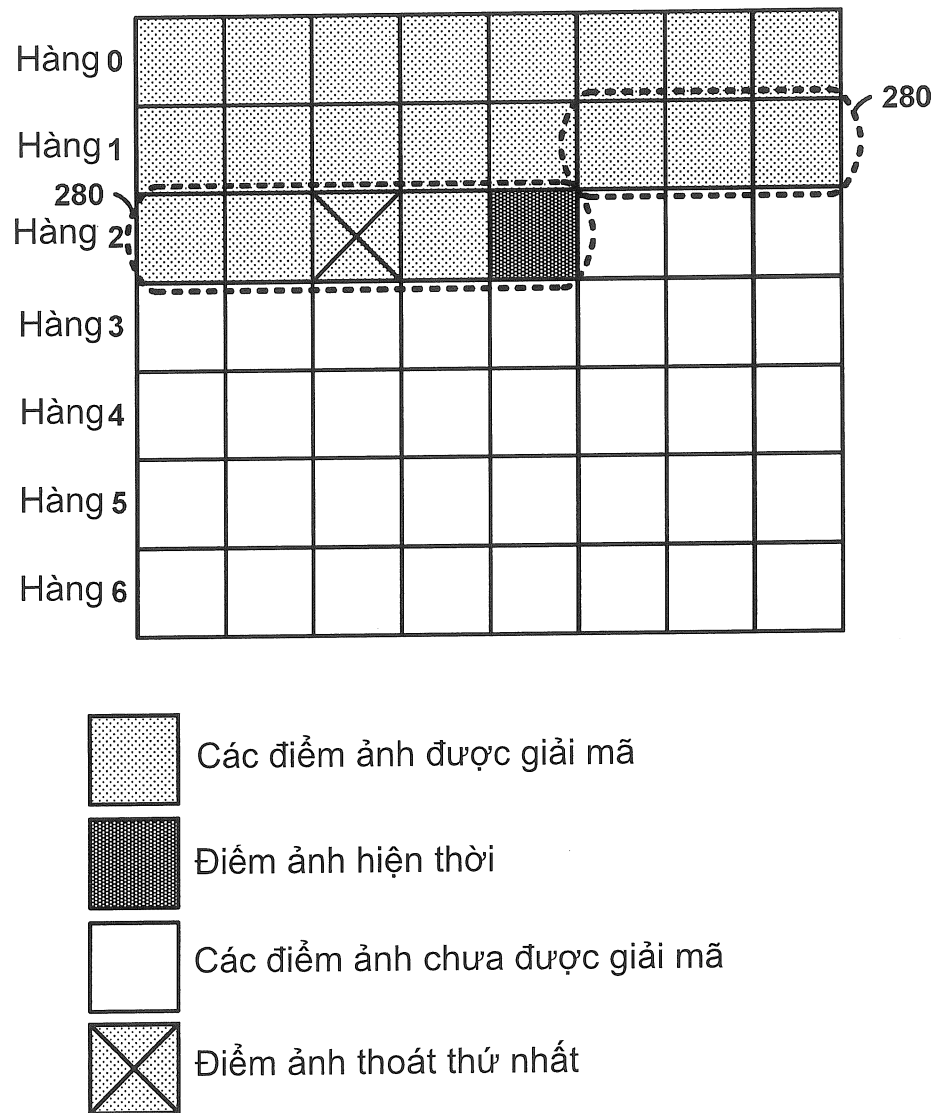


Fig.6

...	
if(currentPaletteSize != 0)	
palette_escape_val_present_flag	ae(v)
indices_idc_coding() (Relocate all occurrence of palette_index_idc to here)	
if(palette_escape_val_present_flag) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_palette_abs	ae(v)
if(cu_qp_delta_palette_abs)	
cu_qp_delta_palette_sign_flag	ae(v)
}	
if(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_palette_offset_flag	ae(v)
if(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	
cu_chroma_qp_palette_offset_idx	ae(v)
}	
}	
if(indexMax > 0)	
palette_transpose_flag	ae(v)
scanPos = 0	
while(scanPos < nCbS * nCbS) {	
...	
if(palette_run_type_flag[xC][yC] == COPY_INDEX_MODE && adjustedIndexMax > 0)	
palette_index_idc	ae(v)
if(indexMax > 0) {	
maxPaletteRun = nCbS * nCbS - scanPos - 1	
if(maxPaletteRun > 0) {	
palette_run_msb_id_plus1	ae(v)
if(palette_run_msb_id_plus1 > 1)	
palette_run_refinement_bits	ae(v)
}	
} else	
...	

Fig.7

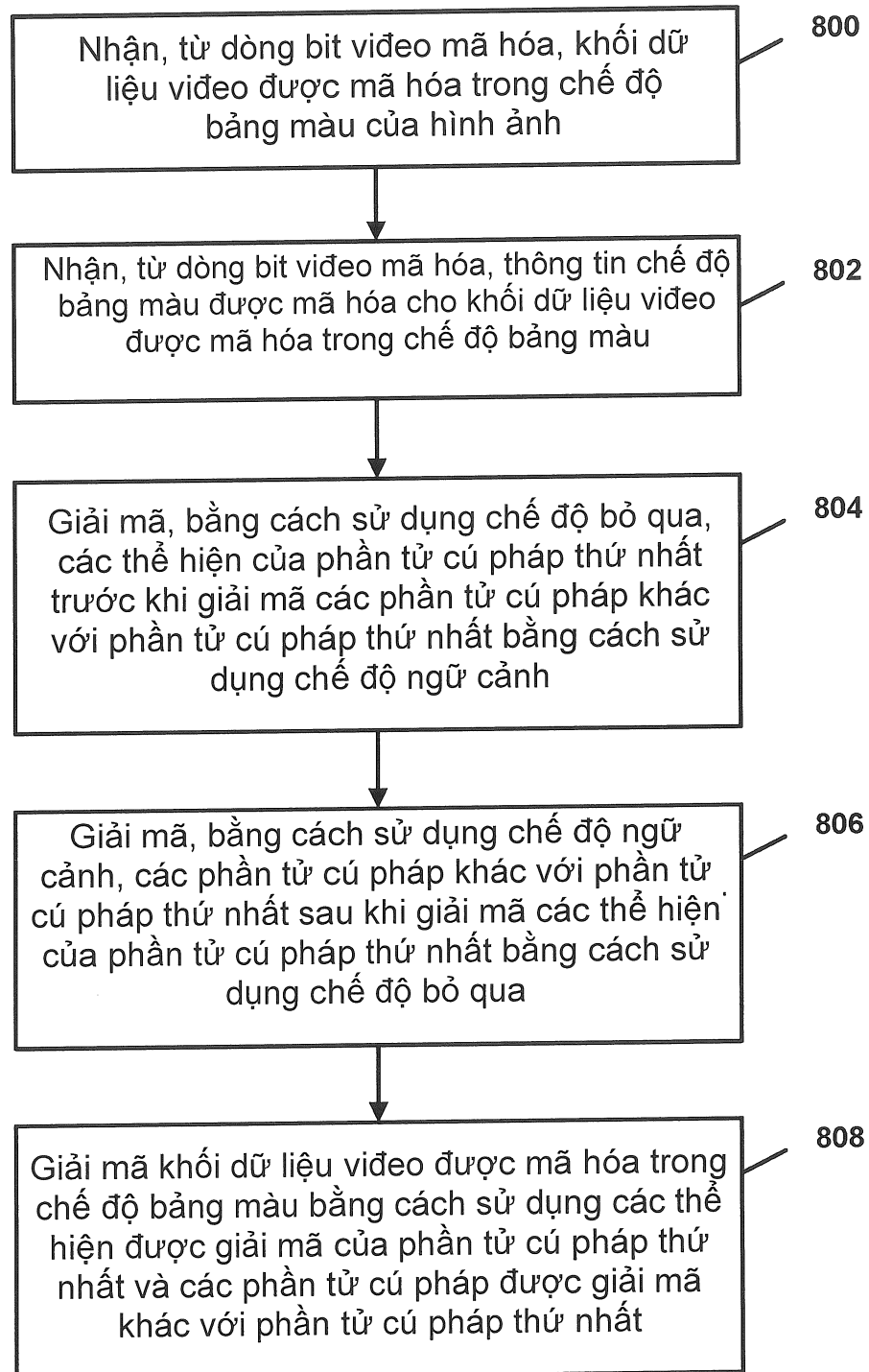


Fig.8

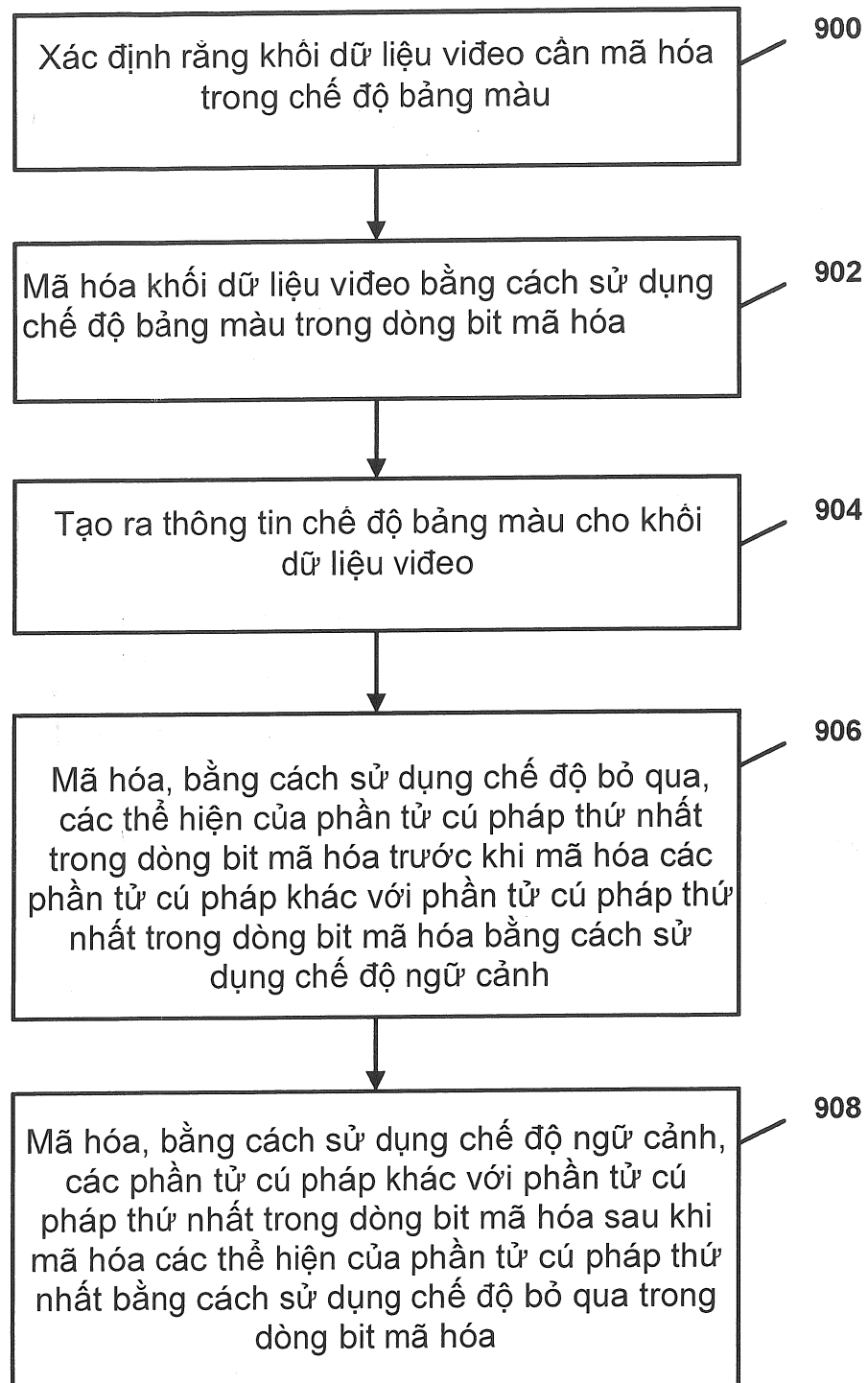


Fig.9