



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ
(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN) (11) 
CỤC SỞ HỮU TRÍ TUỆ
1-0052527
(51)^{2021.01} H04N 7/12; H04N 19/174; H04N 19/136; H04N 19/172 (13) B

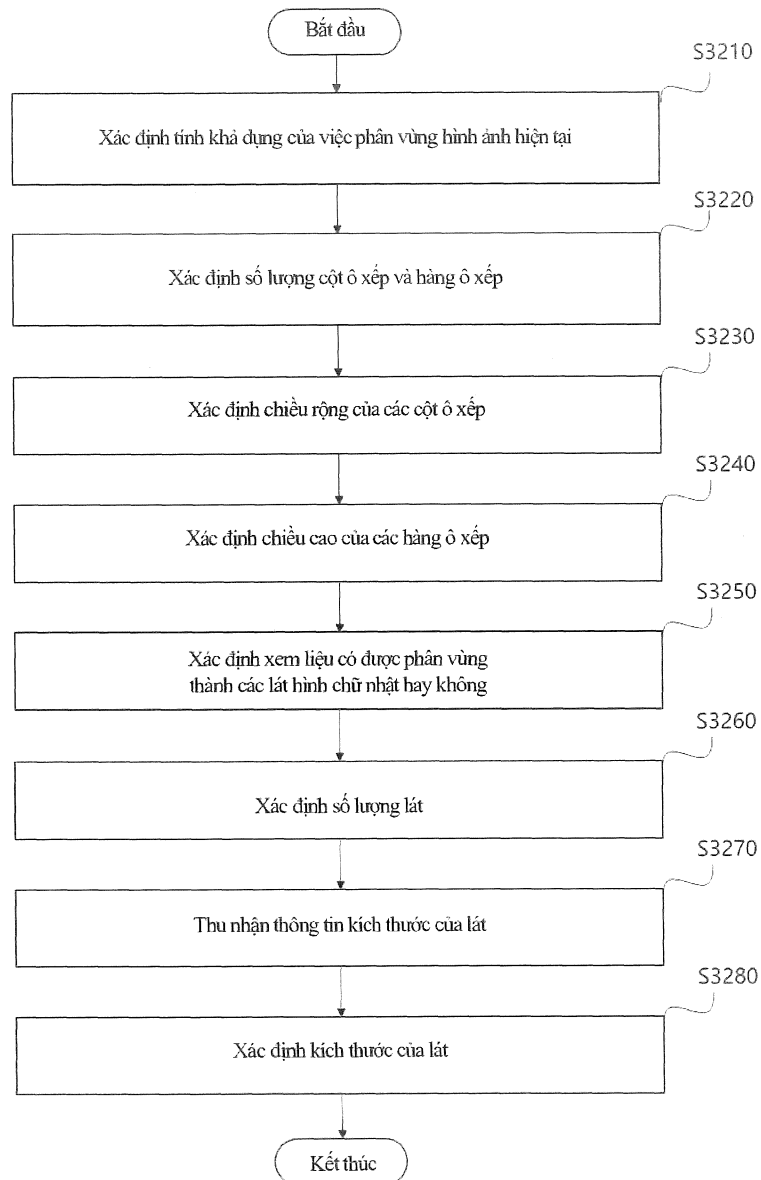
(21) 1-2022-06425 (22) 08/03/2021
(86) PCT/KR2021/002822 08/03/2021 (87) WO 2021/182816 16/09/2021
(30) 62/987,336 09/03/2020 US
(45) 27/10/2025 451 (43) 26/12/2022 417A
(73) LG Electronics Inc. (KR)
128, Yeoui-daero, Yeongdeungpo-gu Seoul 07336, Republic of Korea
(72) HENDRY, Hendry (ID); KIM, Seung Hwan (KR); PALURI, Seethal (IN).
(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP MÃ HÓA VÀ GIẢI MÃ ẢNH, PHƯƠNG PHÁP TRUYỀN
LUỒNG BIT VÀ PHƯƠNG TIỆN PHI CHUYỂN TIẾP ĐỌC ĐƯỢC BẰNG MÁY
TÍNH

(21) 1-2022-06425

(57) Sáng chế đề cập đến phương pháp và thiết bị mã hóa/giải mã ảnh. Phương pháp giải mã ảnh được thực hiện bởi thiết bị mã hóa ảnh có thể bao gồm bước thu nhận thông tin kích thước chỉ ra kích thước của lát hiện tại tương ứng với ít nhất một phần của hình ảnh hiện tại từ luồng bit và xác định kích thước của lát hiện tại dựa trên thông tin kích thước này.

FIG.32



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp và thiết bị mã hóa/giải mã ảnh, và cụ thể hơn là, đề cập đến phương pháp và thiết bị mã hóa và giải mã ảnh để mã hóa thông tin kích thước của lát theo cách có chọn lọc, và phương pháp truyền luồng bit được tạo ra bởi phương pháp/thiết bị mã hóa ảnh của sáng chế.

Tình trạng kỹ thuật của sáng chế

Gần đây, nhu cầu đối với các ảnh độ phân giải cao và chất lượng cao, chẳng hạn như các ảnh độ phân giải cao (High Definition, HD) và các ảnh độ phân giải siêu cao (Ultra High Definition, UHD), đang ngày càng gia tăng trong các lĩnh vực khác nhau. Khi độ phân giải và chất lượng của dữ liệu ảnh được cải thiện, lượng thông tin hoặc bit được truyền tăng tương đối so với dữ liệu ảnh hiện có. Sự gia tăng về lượng thông tin hoặc bit được truyền làm gia tăng chi phí truyền và chi phí lưu trữ.

Theo đó, cần có công nghệ nén ảnh hiệu quả cao để truyền, lưu trữ, và tái tạo một cách hiệu quả thông tin về các ảnh độ phân giải cao và chất lượng cao.

Bản chất kỹ thuật của sáng chế**Vấn đề kỹ thuật**

Một mục đích của sáng chế là đề xuất phương pháp và thiết bị mã hóa/giải mã ảnh có hiệu suất mã hóa/giải mã được cải thiện.

Một mục đích khác của sáng chế là đề xuất phương pháp và thiết bị mã hóa/giải mã ảnh để cải thiện hiệu suất mã hóa/giải mã bằng cách mã hóa thông tin kích thước của lát theo cách có chọn lọc.

Một mục đích khác của sáng chế là để đề xuất phương pháp truyền luồng bit được tạo ra bởi phương pháp hoặc thiết bị mã hóa ảnh theo sáng chế.

Một mục đích khác của sáng chế là để đề xuất phương tiện ghi lưu trữ luồng bit được tạo ra bởi phương pháp hoặc thiết bị mã hóa ảnh theo sáng chế.

Một mục đích khác của sáng chế là để đề xuất phương tiện ghi lưu trữ luồng bit được nhận, được giải mã và được sử dụng để xây dựng lại ảnh bằng thiết bị giải mã ảnh theo sáng chế.

Các vấn đề kỹ thuật được giải quyết bởi sáng chế này không bị giới hạn vào các vấn đề kỹ thuật trên đây và các vấn đề kỹ thuật khác không được mô tả ở đây sẽ trở nên rõ ràng với những người có hiểu biết trong lĩnh vực tương ứng nhờ phần mô tả sau đây.

Giải pháp kỹ thuật của sáng chế

Phương pháp giải mã ảnh được thực hiện bởi thiết bị giải mã ảnh theo một khía cạnh của sáng chế có thể bao gồm bước thu nhận thông tin kích thước chỉ ra kích thước của lát hiện tại tương ứng với ít nhất một phần của hình ảnh hiện tại từ luồng bit và xác định kích thước của lát hiện tại dựa trên thông tin kích thước này. Ở đây, thông tin kích thước có thể bao gồm thông tin chiều rộng chỉ ra chiều rộng của lát hiện tại theo các đơn vị là các cột ô xếp và thông tin chiều cao chỉ ra chiều cao của lát hiện tại theo các đơn vị là các hàng ô xếp, và bước thu nhận thông tin kích thước từ luồng bit có thể được thực hiện dựa trên việc liệu lát hiện tại thuộc về cột ô xếp cuối cùng hay hàng ô xếp cuối cùng của hình ảnh hiện tại.

Thêm vào đó, thiết bị giải mã ảnh theo một khía cạnh của sáng chế có thể bao gồm bộ nhớ và ít nhất một bộ xử lý. Ít nhất một bộ xử lý này có thể thu nhận thông tin kích thước chỉ ra kích thước của lát hiện tại tương ứng với ít nhất một phần của hình ảnh hiện tại từ luồng bit; và xác định kích thước của lát hiện tại dựa trên thông tin kích thước này. Ở đây, thông tin kích thước có thể bao gồm thông tin chiều rộng chỉ ra chiều rộng của lát hiện tại theo các đơn vị là các cột ô xếp và thông tin chiều cao chỉ ra chiều cao của lát hiện tại theo các đơn vị là các hàng

ô xếp, và thông tin kích thước có thể được thu nhận dựa trên việc liệu lát hiện tại thuộc về cột ô xếp cuối cùng hay hàng ô xếp cuối cùng của hình ảnh hiện tại.

Phương pháp mã hóa ảnh được thực hiện bởi thiết bị mã hóa ảnh theo một khía cạnh khác của sáng chế có thể bao gồm bước xác định lát hiện tại tương ứng với ít nhất một phần của hình ảnh hiện tại và tạo ra luồng bit bao gồm thông tin kích thước của lát hiện tại. Ở đây, thông tin kích thước có thể bao gồm thông tin chiều rộng chỉ ra chiều rộng của lát hiện tại theo các đơn vị là các cột ô xếp và thông tin chiều cao chỉ ra chiều cao của lát hiện tại theo các đơn vị là các hàng ô xếp, và bước tạo ra luồng bit bao gồm thông tin kích thước của lát hiện tại có thể được thực hiện dựa trên việc liệu lát hiện tại thuộc về cột ô xếp cuối cùng hay hàng ô xếp cuối cùng của hình ảnh hiện tại.

Thêm vào đó, phương pháp truyền theo một khía cạnh khác của sáng chế có thể truyền luồng bit được tạo ra bởi thiết bị mã hóa ảnh hoặc phương pháp mã hóa ảnh của sáng chế này.

Thêm vào đó, phương tiện ghi có thể đọc được bằng máy tính theo một khía cạnh khác của sáng chế có thể lưu trữ luồng bit được tạo ra bởi thiết bị mã hóa ảnh hoặc phương pháp mã hóa ảnh của sáng chế.

Các dấu hiệu được tóm tắt ngắn gọn trên đây đối với sáng chế này chỉ đơn thuần là các khía cạnh làm ví dụ của phần mô tả chi tiết dưới đây của sáng chế, và không làm giới hạn phạm vi bảo hộ của sáng chế.

Hiệu quả của sáng chế

Theo sáng chế, có thể đề xuất phương pháp và thiết bị mã hóa/giải mã ảnh với hiệu suất mã hóa/giải mã được cải thiện.

Ngoài ra, theo sáng chế, có thể đề xuất phương pháp và thiết bị mã hóa/giải mã ảnh để cải thiện hiệu suất mã hóa/giải mã bằng cách mã hóa thông tin kích thước của lát theo cách có chọn lọc.

Ngoài ra, theo sáng chế, có thể đề xuất phương pháp truyền luồng bit được tạo ra bởi

phương pháp hoặc thiết bị mã hóa ảnh theo sáng chế.

Ngoài ra, theo sáng chế, có thể đề xuất phương tiện ghi lưu trữ luồng bit được tạo ra bởi phương pháp hoặc thiết bị mã hóa ảnh theo sáng chế.

Ngoài ra, theo sáng chế, có thể đề xuất phương tiện ghi lưu trữ luồng bit được nhận, được giải mã và được sử dụng để xây dựng lại ảnh bởi thiết bị giải mã ảnh theo sáng chế.

Những người có hiểu biết trong lĩnh vực kỹ thuật tương ứng sẽ hiểu rằng các hiệu quả có thể đạt được thông qua sáng chế này không bị giới hạn vào những gì đã được mô tả một cách cụ thể trên đây và các ưu điểm khác của sáng chế sẽ được hiểu rõ ràng hơn từ phần mô tả chi tiết.

Mô tả vắn tắt các hình vẽ

Fig.1 là hình thể hiện sơ lược hệ thống lập mã video, mà phương án của sáng chế có thể áp dụng được với nó.

Fig.2 là hình thể hiện sơ lược thiết bị mã hóa ảnh, mà phương án của sáng chế có thể áp dụng được với nó.

Fig.3 là hình thể hiện sơ lược thiết bị giải mã ảnh, mà phương án của sáng chế có thể áp dụng được với nó.

Fig.4 là hình thể hiện cấu trúc phân vùng của ảnh theo một phương án.

Fig.5 là hình thể hiện phương án về loại phân vùng của khối theo cấu trúc cây đa loại.

Fig.6 là hình thể hiện cơ chế báo hiệu thông tin chia tách khối trong cây tứ phân với cấu trúc cây đa loại lồng nhau theo sáng chế.

Fig.7 là hình thể hiện phương án trong đó CTU được phân vùng thành nhiều CU.

Fig.8 là hình thể hiện các mẫu tham chiếu lân cận theo một phương án.

Fig.9 đến Fig.10 là các hình minh họa việc dự đoán nội theo một phương án.

Fig.11 là hình minh họa phương pháp mã hóa sử dụng việc dự đoán liên theo một phương án.

Fig.12 là hình minh họa phương pháp giải mã sử dụng việc dự đoán liên theo một phương án.

Fig.13 là sơ đồ khối của CABAC theo một phương án để mã hóa một phần từ cú pháp.

Fig.14 đến Fig.17 là các hình minh họa việc mã hóa và giải mã entropy theo một phương án.

Fig.18 và Fig.19 là các hình thể hiện ví dụ về thủ tục giải mã và mã hóa hình ảnh theo một phương án.

Fig.20 là hình thể hiện cấu trúc lớp cho ảnh được lập mã theo một phương án.

Fig.21 đến Fig.24 là các hình minh họa phương án về việc phân vùng hình ảnh nhờ sử dụng ô xếp, lát và hình ảnh con.

Fig.25 là hình thể hiện một phương án về cú pháp cho tập tham số trình tự.

Fig.26 là hình thể hiện một phương án về cú pháp của tập tham số hình ảnh.

Fig.27 là hình thể hiện phương án về cú pháp của tiêu đề lát.

Fig.28 và Fig.29 là các hình thể hiện phương án về phương pháp mã hóa và phương pháp giải mã.

Fig.30 và Fig.31 là các hình thể hiện một phương án khác về tập tham số hình ảnh.

Fig.32 là hình minh họa phương án về phương pháp giải mã.

Fig.33 và Fig.34 là các hình thể hiện thuật toán để xác định SliceTopLeftTileIdx.

Fig.35 là hình thể hiện phương án về phương pháp mã hóa.

Fig.36 là hình thể hiện hệ thống tạo luồng nội dung, mà phương án của sáng chế có thể áp dụng được với nó.

Mô tả chi tiết sáng chế

Ở dưới đây, các phương án của sáng chế sẽ được mô tả chi tiết với tham chiếu đến các hình vẽ kèm theo để dễ dàng được thi hành bởi những người có hiểu biết trong lĩnh vực tương ứng. Tuy nhiên, sáng chế có thể được thi hành dưới nhiều dạng khác nhau, và không bị giới

hạn ở các phương án được mô tả ở đây.

Khi mô tả sáng chế này, nếu xác định được rằng phần mô tả chi tiết của chức năng hoặc cấu trúc đã biết có liên quan khiến cho phạm vi của sáng chế này trở nên mơ hồ một cách không cần thiết thì phần mô tả chi tiết của chúng sẽ được lược bỏ. Trên các hình vẽ, các phần không liên quan đến phần mô tả của sáng chế được lược bỏ, và các số tham chiếu tương tự nhau được gán với các phần tương tự nhau.

Trong sáng chế, khi một thành phần được “kết nối”, “ghép nối” hoặc “liên kết” với thành phần khác, điều này có thể bao gồm không chỉ mối quan hệ kết nối trực tiếp mà cả mối quan hệ kết nối gián tiếp trong đó có mặt thành phần xen giữa. Thêm vào đó, khi một thành phần “bao gồm” hoặc “có” các thành phần khác, điều này có nghĩa là các thành phần khác có thể được đưa vào thêm, chứ không phải loại trừ các thành phần khác, trừ khi được nêu rõ khác đi.

Trong sáng chế, các thuật ngữ thứ nhất, thứ hai, v.v. có thể được sử dụng chỉ nhằm mục đích phân biệt một thành phần với các thành phần khác, và không làm giới hạn thứ tự hoặc tầm quan trọng của các thành phần trừ khi được nêu rõ khác đi. Theo đó, trong phạm vi của sáng chế, thành phần thứ nhất trong một phương án có thể được gọi là thành phần thứ hai trong phương án khác, và tương tự, thành phần thứ hai trong một phương án có thể được gọi là thành phần thứ nhất trong phương án khác.

Trong sáng chế, các thành phần mà chúng được phân biệt với nhau được dự tính để mô tả rõ ràng mỗi đặc điểm, và không có nghĩa là các thành phần này nhất thiết phải tách rời nhau. Tức là, nhiều thành phần có thể được hợp nhất và được thi hành trong một đơn vị phần cứng hoặc phần mềm, hoặc một thành phần có thể được phân tán và được thi hành trong nhiều đơn vị phần cứng hoặc phần mềm. Do đó, kể cả nếu không được nêu rõ khác đi thì các phương án như vậy, mà trong đó các thành phần được hợp nhất hoặc thành phần được phân tán, cũng được bao gồm trong phạm vi của sáng chế.

Trong sáng chế, các thành phần được mô tả trong các phương án khác nhau không nhất thiết là để chỉ các thành phần thiết yếu, và một số thành phần có thể là các thành phần tùy chọn.

Theo đó, phương án bao gồm tập con của các thành phần được mô tả trong một phương án cũng được bao gồm trong phạm vi của sáng chế. Thêm vào đó, các phương án bao gồm các thành phần khác, ngoài các thành phần được mô tả trong các phương án khác nhau, được bao gồm trong phạm vi của sáng chế.

Sáng chế đề cập đến việc mã hóa và giải mã ảnh, và các thuật ngữ được sử dụng trong sáng chế này có thể có nghĩa chung được sử dụng phổ biến trong lĩnh vực kỹ thuật, mà sáng chế thuộc về lĩnh vực kỹ thuật này, trừ khi được định nghĩa mới trong sáng chế.

Trong sáng chế này, “video” có thể nghĩa là tập các ảnh theo thời gian. Hình ảnh nói chung đề cập đến đơn vị thể hiện một ảnh tại thời điểm cụ thể, và lát/ô xếp là loại đơn vị lập mã cấu thành nên một phần của hình ảnh trong quy trình lập mã. Một hình ảnh có thể gồm một hoặc nhiều lát/ô xếp. Thêm vào đó, lát/ô xếp có thể bao gồm một hoặc nhiều đơn vị cây lập mã (coding tree unit, CTU). Một hình ảnh có thể gồm một hoặc nhiều lát/ô xếp. Một hình ảnh có thể bao gồm một hoặc nhiều nhóm ô xếp. Một nhóm ô xếp có thể bao gồm một hoặc nhiều ô xếp. Viên gạch có thể đại diện cho vùng hình chữ nhật gồm các hàng CTU trong ô xếp trong hình ảnh. Một ô xếp có thể bao gồm một hoặc nhiều viên gạch. Viên gạch có thể đại diện cho vùng hình chữ nhật gồm các hàng CTU trong ô xếp. Một ô xếp có thể được phân vùng thành nhiều viên gạch và mỗi viên gạch có thể bao gồm một hoặc nhiều hàng CTU thuộc về một ô xếp. Ô xếp không được phân vùng thành nhiều viên gạch cũng có thể được xử lý như là một viên gạch.

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

Trong sáng chế, “đơn vị” có thể biểu diễn đơn vị cơ sở của việc xử lý ảnh. Đơn vị có thể bao gồm ít nhất một thành phần trong số vùng cụ thể của hình ảnh và thông tin liên quan đến

vùng này. Một đơn vị có thể bao gồm một khối độ chói và hai khối sắc độ (ví dụ, Cb và Cr). Đơn vị có thể được sử dụng thay thế được cho nhau với các thuật ngữ chẳng hạn như “mảng mẫu”, “khối” hoặc “khu vực” trong một số trường hợp. Trong trường hợp chung, khối $M \times N$ có thể bao gồm các mẫu (hoặc các mảng mẫu) hoặc tập hợp (hoặc mảng) các hệ số biến đổi gồm M cột và N hàng.

Trong sáng chế, “khối hiện tại” có thể nghĩa là một thành phần trong số “khối lập mã hiện tại”, “đơn vị lập mã hiện tại”, “khối mục tiêu lập mã”, “khối mục tiêu giải mã” hoặc “khối mục tiêu xử lý”. Khi việc dự đoán được thực hiện, “khối hiện tại” có thể nghĩa là “khối dự đoán hiện tại” hoặc “khối mục tiêu dự đoán”. Khi việc biến đổi (biến đổi ngược)/lượng tử hóa (giải lượng tử hóa) được thực hiện, “khối hiện tại” có thể nghĩa là “khối biến đổi hiện tại” hoặc “khối mục tiêu biến đổi”. Khi việc lọc được thực hiện, “khối hiện tại” có thể nghĩa là “khối mục tiêu lọc”.

Thêm vào đó, trong sáng chế này, “khối hiện tại” có thể nghĩa là “khối độ chói của khối hiện tại” trừ khi được nêu tường minh là khối sắc độ. “Khối sắc độ của khối hiện tại” có thể được trình bày bằng cách đưa vào phần mô tả tường minh về khối sắc độ, chẳng hạn như “khối sắc độ” hoặc “khối sắc độ hiện tại”.

Trong sáng chế này, thuật ngữ “/” và “,” cần được hiểu là để thể hiện “và/hoặc.” Ví dụ, cách trình bày “A/B” và “A, B” có thể có nghĩa là “A và/hoặc B.” Hơn nữa, “A/B/C” và “A/B/C” có thể có nghĩa là “ít nhất một thành phần trong số A, B, và/hoặc C.”

Trong sáng chế này, thuật ngữ “hoặc” cần được diễn giải là để thể hiện “và/hoặc.” Ví dụ, cách trình bày “A hoặc B” có thể bao gồm 1) chỉ “A”, 2) chỉ “B”, và/hoặc 3) cả “A và B”. Nói cách khác, trong sáng chế này, thuật ngữ “hoặc” cần được diễn giải là để thể hiện “thêm vào đó hoặc thay vào đó.”

Tổng quan về hệ thống lập mã video

Fig.1 là hình thể hiện hệ thống lập mã video theo sáng chế.

Hệ thống lập mã video theo một phương án có thể bao gồm thiết bị nguồn 10 và thiết bị nhận 20. Thiết bị nguồn 10 có thể phân phát thông tin hoặc dữ liệu video và/hoặc ảnh được mã

hóa đến thiết bị nhận 20 dưới dạng tệp tin hoặc tạo luồng qua phương tiện lưu trữ kỹ thuật số hoặc mạng.

Thiết bị nguồn 10 theo một phương án có thể bao gồm bộ tạo ra nguồn video 11, thiết bị mã hóa 12 và bộ truyền 13. Thiết bị nhận 20 theo một phương án có thể bao gồm bộ nhận 21, thiết bị giải mã 22 và bộ kết xuất 23. Thiết bị mã hóa 12 có thể được gọi là thiết bị mã hóa video/ảnh, và thiết bị giải mã 22 có thể được gọi là thiết bị giải mã video/ảnh. Bộ truyền 13 có thể được chứa trong thiết bị mã hóa 12. Bộ nhận 21 có thể được chứa trong thiết bị giải mã 22. Bộ kết xuất 23 có thể bao gồm bộ hiển thị và bộ hiển thị này có thể được tạo cấu hình như thiết bị tách biệt hoặc thành phần bên ngoài.

Bộ tạo ra nguồn video 11 có thể thu nhận video/ảnh thông qua quy trình chụp, tổng hợp hoặc tạo ra video/ảnh. Bộ tạo ra nguồn video 11 có thể bao gồm thiết bị chụp video/ảnh và/hoặc thiết bị tạo ra video/ảnh. Thiết bị chụp video/ảnh có thể bao gồm, ví dụ, một hoặc nhiều camera, các phần lưu trữ video/ảnh chứa các video/ảnh đã được chụp trước đó, và dạng tương tự. Thiết bị tạo ra video/ảnh có thể bao gồm, ví dụ, máy tính, máy tính bảng, và điện thoại thông minh, và có thể tạo ra các video/ảnh (theo cách điện tử). Ví dụ, video/ảnh ảo có thể được tạo ra qua máy tính hoặc dạng tương tự. Trong trường hợp này, quy trình chụp video/ảnh có thể được thay thế bởi quy trình tạo ra dữ liệu liên quan.

Thiết bị mã hóa 12 có thể mã hóa video/ảnh đầu vào. Thiết bị mã hóa 12 có thể thực hiện chuỗi các thủ tục như dự đoán, biến đổi, và lượng tử hóa để đạt được hiệu quả nén và lập mã. Thiết bị mã hóa 12 có thể xuất ra dữ liệu được mã hóa (thông tin video/ảnh được mã hóa) dưới dạng luồng bit.

Bộ truyền 13 có thể truyền thông tin hoặc dữ liệu video/ảnh được mã hóa được xuất ra dưới dạng luồng bit đến bộ nhận 21 của thiết bị nhận 20 thông qua phương tiện lưu trữ kỹ thuật số hoặc mạng dưới dạng tệp tin hoặc tạo luồng. Phương tiện lưu trữ kỹ thuật số có thể bao gồm các phương tiện lưu trữ khác nhau như USB, SD, CD, DVD, Blu-ray, HDD, SSD, và loại tương tự. Bộ truyền 13 có thể bao gồm phần tử để tạo ra tệp tin phương tiện thông qua định

dạng tệp tin được xác định trước và có thể bao gồm phần tử để truyền qua mạng phát quảng bá/truyền thông. Bộ nhận 21 có thể trích xuất/nhận luồng bit từ phương tiện lưu trữ hoặc mạng và truyền luồng bit đến thiết bị giải mã 22.

Thiết bị giải mã 22 có thể giải mã video/ảnh bằng cách thực hiện chuỗi các thủ tục như giải lượng tử hóa, biến đổi ngược, và dự đoán tương ứng với hoạt động của thiết bị mã hóa 12.

Bộ kết xuất 23 có thể kết xuất video/ảnh được giải mã. Video/ảnh được kết xuất có thể được hiển thị qua bộ hiển thị.

Tổng quan về thiết bị mã hóa ảnh

Fig.2 là hình thể hiện sơ lược thiết bị mã hóa ảnh, mà phương án của sáng chế có thể áp dụng được với nó.

Như được thể hiện trên Fig.2, thiết bị nguồn ảnh 100 có thể bao gồm bộ phân vùng ảnh 110, bộ trừ 115, bộ biến đổi 120, bộ lượng tử hóa 130, bộ giải lượng tử hóa 140, bộ biến đổi ngược 150, bộ cộng 155, bộ lọc 160, bộ nhớ 170, bộ dự đoán liên 180, bộ dự đoán nội 185 và bộ mã hóa entropy 190. Bộ dự đoán liên 180 và bộ dự đoán nội 185 có thể được gọi chung là “bộ dự đoán”. Bộ biến đổi 120, bộ lượng tử hóa 130, bộ giải lượng tử hóa 140 và bộ biến đổi ngược 150 có thể được bao gồm trong bộ xử lý phần dư. Bộ xử lý phần dư có thể còn bao gồm bộ trừ 115.

Tất cả hoặc ít nhất một số thành phần trong số nhiều thành phần tạo cấu hình thiết bị nguồn ảnh 100 có thể được tạo cấu hình bởi một thành phần phần cứng (ví dụ, bộ mã hóa hoặc bộ xử lý) trong một số phương án. Thêm vào đó, bộ nhớ 170 có thể bao gồm bộ đệm hình ảnh được giải mã (decoded picture buffer, DPB) và có thể được tạo cấu hình bằng phương tiện lưu trữ kỹ thuật số.

Bộ phân vùng ảnh 110 có thể phân vùng ảnh (hoặc hình ảnh hoặc khung) đầu vào được nhập vào thiết bị nguồn ảnh 100 thành một hoặc nhiều đơn vị xử lý. Ví dụ, đơn vị xử lý có thể được gọi là đơn vị lập mã (coding unit, CU). Đơn vị lập mã có thể được thu nhận bằng cách phân vùng theo cách đệ quy đơn vị cây lập mã (coding tree unit, CTU) hoặc đơn vị lập mã lớn

nhất (largest coding unit, LCU) theo cấu trúc cây tứ phân cây nhị phân cây tam phân (quad-tree binary-tree ternary-tree, QT/BT/TT). Ví dụ, một đơn vị lập mã có thể được phân vùng thành nhiều đơn vị lập mã có độ sâu sâu hơn dựa trên cấu trúc cây tứ phân, cấu trúc cây nhị phân, và/hoặc cấu trúc tam phân. Để phân vùng đơn vị lập mã, cấu trúc cây tứ phân có thể được áp dụng đầu tiên và cấu trúc cây nhị phân và/hoặc cấu trúc tam phân có thể được áp dụng sau. Thủ tục lập mã theo sáng chế có thể được thực hiện dựa trên đơn vị lập mã cuối cùng mà nó không được phân vùng thêm nữa. Đơn vị lập mã lớn nhất có thể được sử dụng làm đơn vị lập mã cuối cùng, hoặc đơn vị lập mã có độ sâu sâu hơn thu được bằng cách phân vùng đơn vị lập mã lớn nhất có thể được sử dụng làm đơn vị lập mã cuối cùng. Ở đây, thủ tục lập mã có thể bao gồm thủ tục dự đoán, biến đổi, và xây dựng lại, mà chúng sẽ được mô tả sau. Như là một ví dụ khác, đơn vị xử lý của thủ tục lập mã có thể là đơn vị dự đoán (prediction unit, PU) hoặc đơn vị biến đổi (transform unit, TU). Đơn vị dự đoán và đơn vị biến đổi có thể được chia tách hoặc phân vùng từ đơn vị lập mã cuối cùng. Đơn vị dự đoán có thể là đơn vị của việc dự đoán mẫu, và đơn vị biến đổi có thể là đơn vị để dẫn ra hệ số biến đổi và/hoặc đơn vị để dẫn ra tín hiệu phần dư từ hệ số biến đổi này.

Bộ dự đoán (bộ dự đoán liên 180 hoặc bộ dự đoán nội 185) có thể thực hiện việc dự đoán trên khối cần được xử lý (khối hiện tại) và tạo ra khối được dự đoán bao gồm các mẫu dự đoán cho khối hiện tại. Bộ dự đoán có thể xác định xem liệu việc dự đoán nội hay việc dự đoán liên được áp dụng trên cơ sở khối hiện tại hay CU. Bộ dự đoán có thể tạo ra thông tin khác nhau liên quan đến việc dự đoán khối hiện tại và truyền thông tin được tạo ra đến bộ mã hóa entropy 190. Thông tin về việc dự đoán có thể được mã hóa trong bộ mã hóa entropy 190 và được xuất ra dưới dạng luồng bit.

Bộ dự đoán nội 185 có thể dự đoán khối hiện tại bằng cách tham chiếu đến các mẫu trong hình ảnh hiện tại. Các mẫu được tham chiếu có thể được đặt trong vùng lân cận của khối hiện tại hoặc có thể được đặt tách ra tùy theo chế độ dự đoán nội và/hoặc kỹ thuật dự đoán nội. Các chế độ dự đoán nội có thể bao gồm nhiều chế độ không hướng và nhiều chế độ có hướng. Chế

độ không hướng có thể bao gồm, ví dụ, chế độ DC và chế độ phẳng. Chế độ có hướng có thể bao gồm, ví dụ, 33 chế độ dự đoán có hướng hoặc 65 chế độ dự đoán có hướng tùy theo mức độ chi tiết của hướng dự đoán. Tuy nhiên, điều này chỉ đơn thuần là ví dụ, nhiều hoặc ít chế độ dự đoán có hướng hơn có thể được sử dụng phụ thuộc vào việc thiết lập. Bộ dự đoán nội 185 có thể xác định chế độ dự đoán được áp dụng cho khối hiện tại bằng cách sử dụng chế độ dự đoán được áp dụng cho khối lân cận.

Bộ dự đoán liên 180 có thể dẫn ra khối được dự đoán cho khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được chỉ rõ bởi vector chuyển động trên hình ảnh tham chiếu. Trong trường hợp này, để làm giảm lượng thông tin chuyển động được truyền trong chế độ dự đoán liên, thông tin chuyển động có thể được dự đoán theo các đơn vị là các khối, các khối con, hoặc các mẫu dựa trên sự tương quan của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể bao gồm vector chuyển động và chỉ số hình ảnh tham chiếu. Thông tin chuyển động có thể còn bao gồm thông tin hướng dự đoán liên (dự đoán L0, dự đoán L1, song dự đoán (Bi prediction), v.v.). Trong trường hợp dự đoán liên, khối lân cận có thể bao gồm khối lân cận theo không gian có mặt trong hình ảnh hiện tại và khối lân cận theo thời gian có mặt trong hình ảnh tham chiếu. Hình ảnh tham chiếu chứa khối tham chiếu và hình ảnh tham chiếu chứa khối lân cận theo thời gian có thể là giống nhau hoặc khác nhau. Khối lân cận theo thời gian có thể được gọi là khối tham chiếu được đặt cùng vị trí, CU được đặt cùng chỗ (co-located CU, colCU), và dạng tương tự. Hình ảnh tham chiếu chứa khối lân cận theo thời gian có thể được gọi là hình ảnh được đặt cùng vị trí (collocated picture, colPic). Ví dụ, bộ dự đoán liên 180 có thể tạo cấu trúc danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận và tạo ra thông tin chỉ ra ứng viên nào được sử dụng để dẫn ra vector chuyển động và/hoặc chỉ số hình ảnh tham chiếu của khối hiện tại. Việc dự đoán liên có thể được thực hiện dựa trên các chế độ dự đoán khác nhau. Ví dụ, trong trường hợp chế độ bỏ qua (skip) và chế độ hợp nhất (merge), bộ dự đoán liên 180 có thể sử dụng thông tin chuyển động của khối lân cận làm thông tin chuyển động của khối hiện tại. Trong trường hợp chế độ bỏ qua, không giống chế độ

hợp nhất, tín hiệu phần dư có thể không được truyền. Trong trường hợp chế độ dự đoán vector chuyển động (motion vector prediction, MVP), vector chuyển động của khối lân cận có thể được sử dụng làm bộ dự đoán vector chuyển động, và vector chuyển động của khối hiện tại có thể được báo hiệu bằng cách mã hóa chênh lệch vector chuyển động và chỉ dấu cho bộ dự đoán vector chuyển động. Chênh lệch vector chuyển động có thể nghĩa là chênh lệch giữa vector chuyển động của khối hiện tại và bộ dự đoán vector chuyển động.

Bộ dự đoán có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp dự đoán và các kỹ thuật dự đoán khác nhau được mô tả dưới đây. Ví dụ, bộ dự đoán có thể không chỉ áp dụng việc dự đoán nội hoặc dự đoán liên mà còn áp dụng đồng thời cả việc dự đoán nội và dự đoán liên để dự đoán khối hiện tại. Phương pháp dự đoán áp dụng đồng thời cả việc dự đoán nội và việc dự đoán liên cho việc dự đoán khối hiện tại có thể được gọi là việc dự đoán liên và nội kết hợp (combined inter and intra prediction, CIIP). Thêm vào đó, bộ dự đoán có thể thực hiện việc sao chép khối nội (intra block copy, IBC) cho việc dự đoán khối hiện tại. Việc sao chép khối nội có thể được sử dụng cho việc lập mã ảnh/video nội dung của trò chơi hoặc dạng tương tự, ví dụ, lập mã nội dung màn (screen content coding, SCC). IBC là phương pháp dự đoán hình ảnh hiện tại nhờ sử dụng khối tham chiếu được xây dựng lại trước đó trong hình ảnh hiện tại ở vị trí đặt cách khối khối hiện tại một khoảng cách được xác định trước. Khi IBC được áp dụng, vị trí của khối tham chiếu trong hình ảnh hiện tại có thể được mã hóa dưới dạng vector (vector khối) tương ứng với khoảng cách được xác định trước này. IBC về cơ bản thực hiện việc dự đoán trong hình ảnh hiện tại, nhưng có thể được thực hiện tương tự với việc dự đoán liên ở chỗ khối tham chiếu được dẫn ra bên trong hình ảnh hiện tại. Tức là, IBC có thể sử dụng ít nhất một trong các kỹ thuật dự đoán liên được mô tả trong sáng chế này.

Tín hiệu dự đoán được tạo ra bởi bộ dự đoán có thể được sử dụng để tạo ra tín hiệu được xây dựng lại hoặc để tạo ra tín hiệu phần dư. Bộ trừ 115 có thể tạo ra tín hiệu phần dư (khối phần dư hoặc mảng mẫu phần dư) bằng cách lấy tín hiệu ảnh đầu vào (khối gốc hoặc mảng mẫu gốc) trừ đi tín hiệu dự đoán (khối được dự đoán hoặc mảng mẫu dự đoán) được xuất ra từ

bộ dự đoán. Tín hiệu phần dư được tạo ra có thể được truyền đến bộ biến đổi 120.

Bộ biến đổi 120 có thể tạo ra các hệ số biến đổi bằng cách áp dụng kỹ thuật biến đổi cho tín hiệu phần dư. Ví dụ, kỹ thuật biến đổi có thể bao gồm ít nhất một biến đổi trong số biến đổi cosin rời rạc (discrete cosine transform, DCT), biến đổi sin rời rạc (discrete sine transform, DST), biến đổi Karhunen-Loève (Karhunen-Loève transform, KLT), biến đổi dựa trên đồ thị (graph-based transform, GBT), hoặc biến đổi phi tuyến có điều kiện (conditionally non-linear transform, CNT). Ở đây, GBT nghĩa là việc biến đổi thu được từ đồ thị khi thông tin mối quan hệ giữa các điểm ảnh được biểu diễn bằng đồ thị. CNT đề cập đến việc biến đổi thu được dựa trên tín hiệu dự đoán được tạo ra nhờ sử dụng tất cả các điểm ảnh được xây dựng lại trước đó. Thêm vào đó, quy trình biến đổi có thể được áp dụng cho các khối điểm ảnh hình vuông có kích thước giống nhau hoặc có thể được áp dụng cho các khối có kích thước biến thiên thay vì hình vuông.

Bộ lượng tử hóa 130 có thể lượng tử hóa các hệ số biến đổi và truyền chúng đến bộ mã hóa entropy 190. Bộ mã hóa entropy 190 có thể mã hóa tín hiệu được lượng tử hóa (thông tin về các hệ số biến đổi được lượng tử hóa) và xuất ra luồng bit. Thông tin về các hệ số biến đổi được lượng tử hóa có thể được gọi là thông tin phần dư. Bộ lượng tử hóa 130 có thể sắp xếp lại các hệ số biến đổi được lượng tử hóa ở loại khối thành dạng vector một chiều dựa trên thứ tự quét hệ số và tạo ra thông tin về các hệ số biến đổi được lượng tử hóa dựa trên các hệ số biến đổi được lượng tử hóa ở dạng vector một chiều này.

Bộ mã hóa entropy 190 có thể thực hiện các phương pháp mã hóa khác nhau ví dụ chẳng hạn như Golomb hàm số mũ, lập mã chiều dài biến đổi thích ứng ngữ cảnh (context-adaptive variable length coding, CAVLC), lập mã số học nhị phân thích ứng ngữ cảnh (context-adaptive binary arithmetic coding, CABAC), và dạng tương tự. Bộ mã hóa entropy 190 có thể mã hóa thông tin cần thiết cho việc xây dựng lại video/ảnh ngoài các hệ số biến đổi được lượng tử hóa (ví dụ, các giá trị của các phân tử cú pháp, v.v.) cùng nhau hoặc tách biệt. Thông tin được mã hóa (ví dụ, thông tin video/ảnh được mã hóa) có thể được truyền hoặc được lưu trữ theo các

đơn vị là các lớp trừu tượng mạng (network abstraction layer, NAL) dưới dạng luồng bit. Thông tin video/ảnh có thể còn bao gồm thông tin về các tập tham số khác nhau như tập tham số thích ứng (adaptation parameter set, APS), tập tham số hình ảnh (picture parameter set, PPS), tập tham số trình tự (sequence parameter set, SPS), hoặc tập tham số video (video parameter set, VPS). Thêm vào đó, thông tin video/ảnh có thể còn bao gồm thông tin ràng buộc chung. Thông tin được báo hiệu, thông tin và/hoặc các phần tử cú pháp được truyền được mô tả trong sáng chế này có thể được mã hóa qua thủ tục mã hóa được mô tả trên đây và được đưa vào trong luồng bit.

Luồng bit có thể được truyền qua mạng hoặc có thể được lưu trữ trong phương tiện lưu trữ kỹ thuật số. Mạng này có thể bao gồm mạng phát quảng bá và/hoặc mạng truyền thông, và phương tiện lưu trữ kỹ thuật số này có thể bao gồm các phương tiện lưu trữ khác nhau như USB, SD, CD, DVD, Blu-ray, HDD, SSD, và dạng tương tự. Bộ truyền (không được thể hiện) truyền tín hiệu được xuất ra từ bộ mã hóa entropy 190 và/hoặc đơn vị lưu trữ (không được thể hiện) lưu trữ tín hiệu này có thể được đưa vào làm phần tử bên trong/bên ngoài của thiết bị nguồn ảnh 100. Theo cách khác, bộ truyền có thể được bố trí như thành phần của bộ mã hóa entropy 190.

Các hệ số biến đổi được lượng tử hóa được xuất ra từ bộ lượng tử hóa 130 có thể được sử dụng để tạo ra tín hiệu phần dư. Ví dụ, tín hiệu phần dư (khối phần dư hoặc các mẫu phần dư) có thể được xây dựng lại bằng cách áp dụng việc giải lượng tử hóa và biến đổi ngược đối với các hệ số biến đổi được lượng tử hóa qua bộ giải lượng tử hóa 140 và bộ biến đổi ngược 150.

Bộ cộng 155 cộng tín hiệu phần dư được xây dựng lại vào tín hiệu dự đoán được xuất ra từ bộ dự đoán liên 180 hoặc bộ dự đoán nội 185 để tạo ra tín hiệu được xây dựng lại (hình ảnh được xây dựng lại, khối được xây dựng lại, mảng mẫu được xây dựng lại). Nếu không có phần dư nào cho khối cần được xử lý, như trường hợp mà trong đó chế độ bỏ qua được áp dụng, thì khối được dự đoán có thể được sử dụng làm khối được xây dựng lại. Bộ cộng 155 có thể được

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

Bộ lọc 160 có thể cải thiện chất lượng ảnh chủ quan/khách quan bằng cách áp dụng việc lọc cho tín hiệu được xây dựng lại. Ví dụ, bộ lọc 160 có thể tạo ra hình ảnh được xây dựng lại được cải biến bằng cách áp dụng các phương pháp lọc khác nhau đối với hình ảnh được xây dựng lại và lưu trữ hình ảnh được xây dựng lại được cải biến trong bộ nhớ 170, cụ thể là, DPB của bộ nhớ 170. Các phương pháp lọc khác nhau này có thể bao gồm, ví dụ, lọc khử khối, phân bù thích ứng mẫu, bộ lọc vòng thích ứng, bộ lọc song phương, và dạng tương tự. Bộ lọc 160 có thể tạo ra thông tin khác nhau liên quan đến việc lọc và truyền thông tin được tạo ra đến bộ mã hóa entropy 190 như sẽ được mô tả sau trong phần mô tả về mỗi phương pháp lọc. Thông tin liên quan đến việc lọc có thể được mã hóa bởi bộ mã hóa entropy 190 và xuất ra dưới dạng luồng bit.

Hình ảnh được xây dựng lại được cải biến được truyền đến bộ nhớ 170 có thể được sử dụng làm hình ảnh tham chiếu trong bộ dự đoán liên 180. Khi việc dự đoán liên được áp dụng thông qua thiết bị nguồn ảnh 100, việc không khớp dự đoán giữa thiết bị nguồn ảnh 100 và thiết bị giải mã ảnh có thể tránh được và hiệu suất mã hóa có thể được cải thiện.

DPB của bộ nhớ 170 có thể lưu trữ hình ảnh được xây dựng lại được cải biến để sử dụng làm hình ảnh tham chiếu trong bộ dự đoán liên 180. Bộ nhớ 170 có thể lưu trữ thông tin chuyển động của khối mà từ đó thông tin chuyển động trong hình ảnh hiện tại được dẫn ra (hoặc được mã hóa) và/hoặc thông tin chuyển động của các khối trong hình ảnh mà chúng đã được xây dựng lại. Thông tin chuyển động được lưu trữ có thể được truyền đến bộ dự đoán liên 180 và được sử dụng làm thông tin chuyển động của khối lân cận theo không gian hoặc thông tin chuyển động của khối lân cận theo thời gian. Bộ nhớ 170 có thể lưu trữ các mẫu được xây dựng lại của các khối được xây dựng lại trong hình ảnh hiện tại và có thể chuyển các mẫu được xây

dựng lại đến bộ dự đoán nội 185.

Tổng quan về thiết bị giải mã ảnh

Fig.3 là hình thể hiện sơ lược thiết bị giải mã ảnh, mà phương án của sáng chế có thể áp dụng được với nó.

Như được thể hiện trên Fig.3, thiết bị nhận ảnh 200 có thể bao gồm bộ giải mã entropy 210, bộ giải lượng tử hóa 220, bộ biến đổi ngược 230, bộ cộng 235, bộ lọc 240, bộ nhớ 250, bộ dự đoán liên 260 và bộ dự đoán nội 265. Bộ dự đoán liên 260 và bộ dự đoán nội 265 có thể được gọi chung là “bộ dự đoán”. Bộ giải lượng tử hóa 220 và bộ biến đổi ngược 230 có thể được bao gồm trong bộ xử lý phần dư.

Tất cả hoặc ít nhất một số thành phần trong số nhiều thành phần tạo cấu hình thiết bị nhận ảnh 200 có thể được tạo cấu hình bằng thành phần phần cứng (ví dụ, bộ giải mã hoặc bộ xử lý) theo một phương án. Thêm vào đó, bộ nhớ 250 có thể bao gồm bộ đệm hình ảnh được giải mã (decoded picture buffer, DPB) hoặc có thể được tạo cấu hình bằng phương tiện lưu trữ kỹ thuật số.

Thiết bị nhận ảnh 200, mà nó đã nhận luồng bit bao gồm thông tin video/ảnh, có thể xây dựng lại ảnh bằng cách thực hiện quy trình tương ứng với quy trình được thực hiện bởi thiết bị nguồn ảnh 100 trên Fig.2. Ví dụ, thiết bị nhận ảnh 200 có thể thực hiện việc giải mã nhờ sử dụng đơn vị xử lý được áp dụng trong thiết bị mã hóa ảnh. Do đó, đơn vị xử lý của việc giải mã, ví dụ, có thể là đơn vị lập mã. Đơn vị lập mã này có thể được thu nhận bằng cách phân vùng đơn vị cây lập mã hoặc đơn vị lập mã lớn nhất. Tín hiệu ảnh được xây dựng lại được giải mã và được xuất ra qua thiết bị nhận ảnh 200 có thể được tái tạo qua thiết bị tái tạo (không được thể hiện).

Thiết bị nhận ảnh 200 có thể nhận tín hiệu được xuất ra từ thiết bị mã hóa ảnh trên Fig.2 dưới dạng luồng bit. Tín hiệu nhận được có thể được giải mã qua bộ giải mã entropy 210. Ví dụ, bộ giải mã entropy 210 có thể phân tích cú pháp luồng bit để dẫn ra thông tin (ví dụ, thông tin video/ảnh) cần thiết cho việc xây dựng lại ảnh (hoặc xây dựng lại hình ảnh). Thông tin

video/ảnh có thể còn bao gồm thông tin về các tập tham số khác nhau như tập tham số thích ứng (adaptation parameter set, APS), tập tham số hình ảnh (picture parameter set, PPS), tập tham số trình tự (sequence parameter set, SPS), hoặc tập tham số video (video parameter set, VPS). Thêm vào đó, thông tin video/ảnh có thể còn bao gồm thông tin ràng buộc chung. Thiết bị giải mã ảnh có thể còn giải mã hình ảnh dựa trên thông tin về tập tham số và/hoặc thông tin ràng buộc chung. Thông tin và/hoặc các phần tử cú pháp được báo hiệu/được nhận được mô tả trong sáng chế này có thể được giải mã qua thủ tục giải mã và được thu từ luồng bit. Ví dụ, bộ giải mã entropy 210 giải mã thông tin trong luồng bit dựa trên phương pháp lập mã như lập mã Golomb hàm số mũ, CAVLC, hoặc CABAC, và các giá trị được xuất ra của các phần tử cú pháp cần thiết cho việc xây dựng lại ảnh và các giá trị được lượng tử hóa của các hệ số biến đổi cho phần dư. Cụ thể hơn, phương pháp giải mã entropy CABAC có thể nhận ngăn (bin) tương ứng với mỗi phần tử cú pháp trong luồng bit, xác định mô hình ngữ cảnh nhờ sử dụng thông tin về phần tử cú pháp mục tiêu giải mã, thông tin giải mã của khối lân cận và khối mục tiêu giải mã hoặc thông tin về ký hiệu/ngăn được giải mã trong giai đoạn trước, và thực hiện việc giải mã số học trên ngăn này bằng cách dự đoán xác suất xuất hiện ngăn theo mô hình ngữ cảnh xác định được, và tạo ra ký hiệu tương ứng với giá trị của mỗi phần tử cú pháp. Trong trường hợp này, phương pháp giải mã entropy CABAC có thể cập nhật mô hình ngữ cảnh bằng cách sử dụng thông tin về ký hiệu/ngăn được giải mã cho mô hình ngữ cảnh của ký hiệu/ngăn tiếp theo sau khi xác định mô hình ngữ cảnh. Thông tin liên quan đến việc dự đoán trong số thông tin được giải mã bởi bộ giải mã entropy 210 có thể được cung cấp cho bộ dự đoán (bộ dự đoán liên 260 và bộ dự đoán nội 265), và giá trị phần dư mà việc giải mã entropy được thực hiện trên đó trong bộ giải mã entropy 210, tức là, các hệ số biến đổi được lượng tử hóa và thông tin tham số liên quan, có thể được cung cấp cho bộ giải lượng tử hóa 220. Thêm vào đó, thông tin về việc lọc trong số thông tin được giải mã bởi bộ giải mã entropy 210 có thể được cung cấp cho bộ lọc 240. Trong khi đó, bộ nhận (không được thể hiện) để nhận tín hiệu được xuất ra từ thiết bị mã hóa ảnh có thể còn được tạo cấu hình dưới dạng phần tử bên trong/bên ngoài của thiết bị

nhận ảnh 200, hoặc bộ nhận này có thể là thành phần của bộ giải mã entropy 210.

Trong khi đó, thiết bị giải mã ảnh theo sáng chế có thể được gọi là thiết bị giải mã video/ảnh/hình ảnh. Thiết bị giải mã ảnh có thể được phân loại thành bộ giải mã thông tin (bộ giải mã thông tin video/ảnh/hình ảnh) và bộ giải mã mẫu (bộ giải mã mẫu video/ảnh/hình ảnh). Bộ giải mã thông tin có thể bao gồm bộ giải mã entropy 210. Bộ giải mã mẫu có thể bao gồm ít nhất một thành phần trong số bộ giải lượng tử hóa 220, bộ biến đổi ngược 230, bộ cộng 235, bộ lọc 240, bộ nhớ 250, bộ dự đoán liên 160 hoặc bộ dự đoán nội 265.

Bộ giải lượng tử hóa 220 có thể giải lượng tử hóa các hệ số biến đổi được lượng tử hóa và xuất ra các hệ số biến đổi. Bộ giải lượng tử hóa 220 có thể sắp xếp lại các hệ số biến đổi được lượng tử hóa dưới dạng khối hai chiều. Trong trường hợp này, việc sắp xếp lại có thể được thực hiện dựa trên thứ tự quét hệ số được thực hiện trong thiết bị mã hóa ảnh. Bộ giải lượng tử hóa 220 có thể thực hiện việc giải lượng tử hóa trên các hệ số biến đổi được lượng tử hóa bằng cách sử dụng tham số lượng tử hóa (ví dụ, thông tin kích thước bước lượng tử hóa) và thu các hệ số biến đổi.

Bộ biến đổi ngược 230 có thể biến đổi ngược các hệ số biến đổi để thu tín hiệu phần dư (khối phần dư, mảng mẫu phần dư).

Bộ dự đoán có thể thực hiện việc dự đoán trên khối hiện tại và tạo ra khối được dự đoán bao gồm các mẫu dự đoán cho khối hiện tại. Bộ dự đoán có thể xác định xem liệu việc dự đoán nội hay việc dự đoán liên được áp dụng cho khối hiện tại dựa trên thông tin về việc dự đoán được xuất ra từ bộ giải mã entropy 210 và có thể xác định chế độ dự đoán (kỹ thuật dự đoán) nội/liên cụ thể.

Điều này là giống như được mô tả trong bộ dự đoán của thiết bị nguồn ảnh 100 ở chỗ bộ dự đoán có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp (các kỹ thuật) dự đoán khác nhau mà chúng sẽ được mô tả sau.

Bộ dự đoán nội 265 có thể dự đoán khối hiện tại bằng cách tham chiếu đến các mẫu trong hình ảnh hiện tại. Phần mô tả của bộ dự đoán nội 185 được áp dụng tương đương với bộ dự

đoán nội 265.

Bộ dự đoán liên 260 có thể dẫn ra khối được dự đoán cho khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được chỉ rõ bởi vectơ chuyển động trên hình ảnh tham chiếu. Trong trường hợp này, để làm giảm lượng thông tin chuyển động được truyền trong chế độ dự đoán liên, thông tin chuyển động có thể được dự đoán theo các đơn vị là các khối, các khối con, hoặc các mẫu dựa trên sự tương quan của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể bao gồm vectơ chuyển động và chỉ số hình ảnh tham chiếu. Thông tin chuyển động có thể còn bao gồm thông tin hướng dự đoán liên (dự đoán L0, dự đoán L1, song dự đoán (Bi prediction), v.v.). Trong trường hợp dự đoán liên, khối lân cận có thể bao gồm khối lân cận theo không gian có mặt trong hình ảnh hiện tại và khối lân cận theo thời gian có mặt trong hình ảnh tham chiếu. Ví dụ, bộ dự đoán liên 260 có thể tạo cấu hình danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận và dẫn ra vectơ chuyển động của khối hiện tại và/hoặc chỉ số hình ảnh tham chiếu dựa trên thông tin chọn ứng viên nhận được. Việc dự đoán liên có thể được thực hiện dựa trên các chế độ dự đoán khác nhau, và thông tin về việc dự đoán có thể bao gồm thông tin chỉ ra chế độ dự đoán liên cho khối hiện tại.

Bộ cộng 235 có thể tạo ra tín hiệu được xây dựng lại (hình ảnh được xây dựng lại, khối được xây dựng lại, mảng mẫu được xây dựng lại) bằng cách cộng tín hiệu phần dư thu được vào tín hiệu dự đoán (khối được dự đoán, mảng mẫu được dự đoán) được xuất ra từ bộ dự đoán (bao gồm bộ dự đoán liên 260 và/hoặc bộ dự đoán nội 265). Nếu không có phần dư nào cho khối cần được xử lý, như khi chế độ bỏ qua được áp dụng, thì khối được dự đoán có thể được sử dụng làm khối được xây dựng lại. Phần mô tả của bộ cộng 155 là có thể áp dụng tương đương được với bộ cộng 235. Bộ cộng 235 có thể được gọi là bộ xây dựng lại hoặc bộ tạo ra khối được xây dựng lại. Tín hiệu được xây dựng lại được tạo ra có thể được sử dụng cho việc dự đoán nội khối tiếp theo cần được xử lý trong hình ảnh hiện tại và có thể được sử dụng cho việc dự đoán liên hình ảnh tiếp theo qua việc lọc như được mô tả dưới đây.

Bộ lọc 240 có thể cải thiện chất lượng ảnh chủ quan/khách quan bằng cách áp dụng việc

lọc cho tín hiệu được xây dựng lại. Ví dụ, bộ lọc 240 có thể tạo ra hình ảnh được xây dựng lại được cải biến bằng cách áp dụng các phương pháp lọc khác nhau đối với hình ảnh được xây dựng lại và lưu trữ hình ảnh được xây dựng lại được cải biến trong bộ nhớ 250, cụ thể là, DPB của bộ nhớ 250. Các phương pháp lọc khác nhau này có thể bao gồm, ví dụ, lọc khử khối, phân bù thích ứng mẫu, bộ lọc vòng thích ứng, bộ lọc song phương, và dạng tương tự.

Hình ảnh được xây dựng lại (được cải biến) được lưu trữ trong DPB của bộ nhớ 250 có thể được sử dụng làm hình ảnh tham chiếu trong bộ dự đoán liên 260. Bộ nhớ 250 có thể lưu trữ thông tin chuyển động của khối mà thông tin chuyển động trong hình ảnh hiện tại được dẫn ra (hoặc được giải mã) từ đó và/hoặc thông tin chuyển động của các khối trong hình ảnh mà chúng đã được xây dựng lại sẵn. Thông tin chuyển động được lưu trữ có thể được truyền đến bộ dự đoán liên 260 sao cho nó được sử dụng làm thông tin chuyển động của khối lân cận theo không gian hoặc thông tin chuyển động của khối lân cận theo thời gian. Bộ nhớ 250 có thể lưu trữ các mẫu được xây dựng lại của các khối được xây dựng lại trong hình ảnh hiện tại và chuyển các mẫu được xây dựng lại này đến bộ dự đoán nội 265.

Trong sáng chế này, các phương án được mô tả trong bộ lọc 160, bộ dự đoán liên 180, và bộ dự đoán nội 185 của thiết bị nguồn ảnh 100 có thể được áp dụng tương đương hoặc tương ứng với bộ lọc 240, bộ dự đoán liên 260, và bộ dự đoán nội 265 của thiết bị nhận ảnh 200.

Tổng quan về việc phân vùng ảnh

Phương pháp lập mã video/ảnh theo sáng chế này có thể được thực hiện dựa trên cấu trúc phân vùng ảnh như sau. Cụ thể, các thủ tục về việc dự đoán, xử lý phần dư (biến đổi (ngược), lượng tử hóa (giải lượng tử hóa), v.v.), lập mã phần tử cú pháp, và lọc, mà chúng sẽ được mô tả sau, có thể được thực hiện dựa trên CTU, CU (và/hoặc TU, PU) được dẫn ra dựa trên cấu trúc phân vùng ảnh. Ảnh có thể được phân vùng theo các đơn vị khối và thủ tục phân vùng khối có thể được thực hiện trong bộ phân vùng ảnh 110 của thiết bị mã hóa. Thông tin liên quan đến việc phân vùng có thể được mã hóa bởi bộ mã hóa entropy 190 và được truyền đến thiết bị giải mã dưới dạng luồng bit. Bộ giải mã entropy 210 của thiết bị giải mã có thể dẫn ra cấu trúc

phân vùng khối của hình ảnh hiện tại dựa trên thông tin liên quan đến việc phân vùng thu được từ luồng bit, và dựa trên điều này, có thể thực hiện chuỗi các thủ tục (ví dụ, dự đoán, xử lý phần dư, xây dựng lại khối/hình ảnh, lọc trong vòng, v.v) để giải mã ảnh.

Các hình ảnh có thể được phân vùng thành trình tự gồm các đơn vị cây lập mã (CTU). Fig.4 thể hiện ví dụ trong đó hình ảnh được phân vùng thành các CTU. CTU này có thể tương ứng với khối cây lập mã (coding tree block, CTB). Theo cách khác, CTU có thể bao gồm khối cây lập mã của các mẫu độ chói và hai khối cây lập mã của các mẫu sắc độ tương ứng. Ví dụ, đối với hình ảnh mà nó chứa ba mảng mẫu, CTU có thể bao gồm khối $N \times N$ của các mẫu độ chói và hai khối tương ứng của các mẫu sắc độ. Kích thước được cho phép tối đa của CTU cho việc lập mã và dự đoán có thể khác với kích thước được cho phép tối đa của CTU cho việc biến đổi. Ví dụ, kích thước được cho phép tối đa của khối độ chói trong CTU có thể là 128×128 kể cả khi kích thước tối đa của các khối biến đổi độ chói là 64×64 .

Tổng quan về việc phân vùng CTU

Như được mô tả trên đây, đơn vị lập mã có thể được thu nhận bằng cách phân vùng theo cách đệ quy đơn vị cây lập mã (CTU) hoặc đơn vị lập mã lớn nhất (LCU) theo cấu trúc cây tứ phân/cây nhị phân/cây tam phân (QT/BT/TT). Ví dụ, CTU trước hết có thể được phân vùng thành các cấu trúc cây tứ phân. Sau đó, các nút lá của cấu trúc cây tứ phân này có thể còn được phân vùng bằng cấu trúc cây đa loại.

Việc phân vùng theo cây tứ phân nghĩa là CU (hoặc CTU) hiện tại được phân vùng thành bốn phần bằng nhau. Qua việc phân vùng theo cây tứ phân, CU hiện tại có thể được phân vùng thành bốn CU có chiều rộng giống nhau và chiều cao giống nhau. Khi CU hiện tại không còn được phân vùng thành các cấu trúc cây tứ phân, CU hiện tại tương ứng với nút lá của cấu trúc cây tứ phân. CU tương ứng với nút lá của cấu trúc cây tứ phân có thể không còn được phân vùng và có thể được sử dụng làm đơn vị lập mã cuối cùng được mô tả trên đây. Theo cách khác, CU tương ứng với nút lá của cấu trúc cây tứ phân có thể còn được phân vùng bằng cấu trúc cây đa loại.

Fig.5 là hình thể hiện phương án về loại phân vùng của khối theo cấu trúc cây đa loại. Việc phân vùng theo cấu trúc cây đa loại có thể bao gồm hai loại chia tách theo cấu trúc cây nhị phân và hai loại chia tách theo cấu trúc cây tam phân.

Hai loại chia tách theo cấu trúc cây nhị phân có thể bao gồm việc chia tách nhị phân dọc (SPLIT_BT_VER) và việc chia tách nhị phân ngang (SPLIT_BT_HOR). Việc chia tách nhị phân dọc (SPLIT_BT_VER) nghĩa là CU hiện tại được chia tách thành hai phần bằng nhau theo hướng dọc. Như được thể hiện trên Fig.4, qua việc chia tách nhị phân dọc, hai CU có chiều cao giống với CU hiện tại và có chiều rộng mà nó là nửa chiều rộng của CU hiện tại có thể được tạo ra. Việc chia tách nhị phân ngang (SPLIT_BT_HOR) nghĩa là CU hiện tại được chia tách thành hai phần bằng nhau theo hướng ngang. Như được thể hiện trên Fig.5, qua việc chia tách nhị phân ngang, hai CU có chiều cao là nửa chiều cao của CU hiện tại và có chiều rộng giống với CU hiện tại có thể được tạo ra.

Hai loại chia tách theo cấu trúc cây tam phân có thể bao gồm việc chia tách tam phân dọc (SPLIT_TT_VER) và việc chia tách tam phân ngang (SPLIT_TT_HOR). Trong việc chia tách tam phân dọc (SPLIT_TT_VER), CU hiện tại được chia tách theo hướng dọc ở tỷ lệ là 1:2:1. Như được thể hiện trên Fig.4, qua việc chia tách tam phân dọc, hai CU có chiều cao giống với CU hiện tại và có chiều rộng mà nó là 1/5 chiều rộng của CU hiện tại và CU có chiều cao giống với CU hiện tại và có chiều rộng mà nó là nửa chiều rộng của CU hiện tại có thể được tạo ra. Trong việc chia tách tam phân ngang (SPLIT_TT_HOR), CU hiện tại được chia tách theo hướng ngang ở tỷ lệ là 1:2:1. Như được thể hiện trên Fig.5, qua việc chia tách tam phân ngang, hai CU có chiều cao mà nó là 1/4 chiều cao của CU hiện tại và có chiều rộng giống với CU hiện tại và CU có chiều cao mà nó là nửa chiều cao của CU hiện tại và có chiều rộng giống với CU hiện tại có thể được tạo ra.

Fig.6 là hình thể hiện cơ chế báo hiệu thông tin chia tách khối trong cây tứ phân với cấu trúc cây đa loại lồng nhau theo sáng chế.

Ở đây, CTU được xem như nút gốc của cây tứ phân, và được phân vùng cho lần đầu tiên

thành cấu trúc cây tứ phân. Thông tin (ví dụ, `qt_split_flag`) chỉ ra liệu việc chia tách cây tứ phân có được thực hiện đối với CU (CTU hay nút (`QT_node`) của cây tứ phân) hiện tại hay không có thể được báo hiệu. Ví dụ, khi `qt_split_flag` có giá trị thứ nhất (ví dụ, “1”), CU hiện tại có thể được phân vùng cây tứ phân. Thêm vào đó, khi `qt_split_flag` có giá trị thứ hai (ví dụ, “0”), CU hiện tại không được phân vùng cây tứ phân, mà trở thành nút lá (`QT_leaf_node`) của cây tứ phân. Mỗi nút lá cây tứ phân khi đó có thể được phân vùng thêm thành các cấu trúc cây đa loại. Tức là, nút lá của cây tứ phân có thể trở thành nút (`MTT_node`) của cây đa loại. Trong cấu trúc cây đa loại, cờ thứ nhất (ví dụ, `Mtt_split_cu_flag`) có thể được báo hiệu để chỉ ra liệu nút hiện tại có được phân vùng thêm hay không. Nếu nút tương ứng được phân vùng thêm (ví dụ, nếu cờ thứ nhất là 1) thì cờ thứ hai (ví dụ, `Mtt_split_cu_vertical_flag`) có thể được báo hiệu để chỉ ra hướng chia tách. Ví dụ, hướng chia tách có thể là hướng dọc nếu cờ thứ hai là 1 và có thể là hướng ngang nếu cờ thứ hai là 0. Sau đó, cờ thứ ba (ví dụ, `Mtt_split_cu_binary_flag`) có thể được báo hiệu để chỉ ra liệu loại chia tách là loại chia tách nhị phân hay loại chia tách tam phân. Ví dụ, loại chia tách có thể là loại chia tách nhị phân khi cờ thứ ba là 1 và có thể là loại chia tách tam phân khi cờ thứ ba là 0. Nút của cây đa loại thu nhận được bằng việc chia tách nhị phân hoặc việc chia tách tam phân có thể còn được phân vùng thành các cấu trúc cây đa loại. Tuy nhiên, nút của cây đa loại này có thể không được phân vùng thành các cấu trúc cây tứ phân. Nếu cờ thứ nhất là 0 thì nút tương ứng của cây đa loại không còn được chia tách nữa mà trở thành nút lá (`MTT_leaf_node`) của cây đa loại. CU tương ứng với nút lá của cây đa loại có thể được sử dụng làm đơn vị lập mã cuối cùng được mô tả trên đây.

Dựa trên `mtt_split_cu_vertical_flag` và `mtt_split_cu_binary_flag`, chế độ chia tách cây đa loại (`MttSplitMode`) của CU có thể được dẫn ra như được thể hiện trong bảng 1 dưới đây. Trong phần mô tả sau đây, chế độ chia tách cây đa loại có thể được gọi là loại chia tách đa cây hoặc loại chia tách.

Bảng 1

<code>MttSplitMode</code>	<code>mtt_split_cu_vertical_flag</code>	<code>mtt_split_cu_binary_flag</code>
---------------------------	---	---------------------------------------

SPLIT TT HOR	0	0
SPLIT BT HOR	0	1
SPLIT TT VER	1	0
SPLIT BT VER	1	1

Fig.7 là hình thể hiện ví dụ trong đó CTU được phân vùng thành nhiều CU bằng cách áp dụng cây đa loại sau khi áp dụng cây tứ phân. Trên Fig.7, các cạnh khối nét đậm 710 biểu diễn việc phân vùng cây tứ phân và các cạnh còn lại 720 biểu diễn việc phân vùng cây đa loại. CU này có thể tương ứng với khối lập mã (coding block, CB). Trong một phương án, CU có thể bao gồm khối lập mã của các mẫu độ chói và hai khối lập mã của các mẫu sắc độ tương ứng với các mẫu độ chói này. Kích thước CB hoặc TB (mẫu) thành phần sắc độ có thể được dẫn ra dựa trên kích thước CB hoặc TB (mẫu) thành phần độ chói theo tỷ lệ thành phần tùy theo định dạng màu (định dạng sắc độ, ví dụ, 4:4:4, 4:2:2, 4:2:0 hoặc dạng tương tự) của hình ảnh/ảnh. Trong trường hợp định dạng màu 4:4:4, kích thước CB/TB thành phần sắc độ có thể được thiết lập bằng kích thước CB/TB thành phần độ chói. Trong trường hợp định dạng màu 4:2:2, chiều rộng của CB/TB thành phần sắc độ có thể được thiết lập thành nửa chiều rộng của CB/TB thành phần độ chói và chiều cao của CB/TB thành phần sắc độ có thể được thiết lập thành chiều cao của CB/TB thành phần độ chói. Trong trường hợp định dạng màu 4:2:0, chiều rộng của CB/TB thành phần sắc độ có thể được thiết lập thành nửa chiều rộng của CB/TB thành phần độ chói và chiều cao của CB/TB thành phần sắc độ có thể được thiết lập thành nửa chiều cao của CB/TB thành phần độ chói.

Trong một phương án, khi kích thước của CTU là 128 dựa trên đơn vị mẫu độ chói, kích thước của CU có thể có kích thước từ 128x128 đến 4x4 mà nó là kích thước giống như CTU. Trong một phương án, trong trường hợp định dạng màu (hoặc định dạng sắc độ) 4:2:0, kích thước CB sắc độ có thể có kích thước từ 64x64 đến 2x2.

Trong khi đó, trong một phương án, kích thước CU và kích thước TU có thể là giống nhau. Theo cách khác, có thể có nhiều TU trong một vùng CU. Kích thước TU nhìn chung có thể thể hiện kích thước khối biến đổi (TB) (mẫu) thành phần độ chói.

Kích thước TU có thể được dẫn ra dựa vào kích thước TB được cho phép lớn nhất maxTbSize mà nó là giá trị được xác định trước. Ví dụ, khi kích thước CU lớn hơn maxTbSize , nhiều TU (TB) có maxTbSize có thể được dẫn ra từ CU và việc biến đổi/biến đổi ngược có thể được thực hiện theo các đơn vị là TU (TB). Ví dụ, kích thước TB độ chói được cho phép lớn nhất có thể là 64×64 và kích thước TB sắc độ được cho phép lớn nhất có thể là 32×32 . Nếu chiều rộng hoặc chiều cao của CB được phân vùng theo cấu trúc cây là lớn hơn chiều rộng hoặc chiều cao biến đổi lớn nhất thì CB có thể được phân vùng tự động (hoặc ngầm định) cho đến khi giới hạn kích thước TB theo các hướng ngang và dọc được thỏa mãn.

Thêm vào đó, ví dụ, khi việc dự đoán nội được áp dụng, chế độ/loại dự đoán nội có thể được dẫn ra theo các đơn vị là CU (hoặc CB) và thủ tục dẫn ra mẫu tham chiếu lân cận và tạo ra mẫu dự đoán có thể được thực hiện theo các đơn vị là TU (hoặc TB). Trong trường hợp này, có thể có một hoặc nhiều TU (hoặc TB) trong một vùng CU (hoặc CB) và, trong trường hợp này, nhiều TU (hoặc TB) này có thể chia sẻ chế độ/loại dự đoán nội giống nhau.

Trong khi đó, đối với sơ đồ cây lập mã cây tứ phân với cây đa loại lồng nhau, các tham số sau đây có thể được báo hiệu dưới dạng các phần tử cú pháp SPS từ thiết bị mã hóa đến thiết bị giải mã. Ví dụ, ít nhất một tham số trong số kích thước CTU mà nó là tham số thể hiện kích thước nút gốc của cây tứ phân, MinQTSize mà nó là tham số thể hiện kích thước nút lá cây tứ phân tối thiểu được cho phép, MaxBtSize mà nó là tham số thể hiện kích thước nút gốc cây nhị phân tối đa được cho phép, MaxTtSize mà nó là tham số thể hiện kích thước nút gốc cây tam phân tối đa được cho phép, MaxMttDepth mà nó là tham số thể hiện độ sâu hệ phân cấp tối đa được cho phép của việc chia tách cây đa loại từ nút lá cây tứ phân, MinBtSize mà nó là tham số thể hiện kích thước nút lá cây nhị phân tối thiểu được cho phép, hoặc MinTtSize mà nó là tham số thể hiện kích thước nút lá cây tam phân tối thiểu được cho phép có thể được báo hiệu.

Để làm một phương án về việc sử dụng định dạng sắc độ 4:2:0, kích thước CTU có thể được thiết lập thành các khối độ chói 128×128 và hai khối sắc độ 64×64 tương ứng với các khối độ chói này. Trong trường hợp này, MinOTSize có thể được thiết lập là 16×16 , MaxBtSize có

thể được thiết lập thành 128×128 , $MaxTtSize$ có thể được thiết lập thành 64×64 , $MinBtSize$ và $MinTtSize$ có thể được thiết lập thành 4×4 , và $MaxMttDepth$ có thể được thiết lập thành 4. Việc phân vùng cây tứ phân có thể được áp dụng cho CTU để tạo ra các nút lá cây tứ phân. Nút lá cây tứ phân có thể được gọi là nút lá QT. Nút lá cây tứ phân có thể có kích thước từ kích thước 16×16 (ví dụ, $MinOTSize$) đến kích thước 128×128 (ví dụ, kích thước CTU). Nếu nút lá QT là 128×128 thì nút lá này có thể không được phân vùng thêm thành cây nhị phân/cây tam phân. Điều này là bởi vì, trong trường hợp này, kể cả nếu được phân vùng thì nó cũng vượt quá $MaxBtSize$ và $MaxTtSize$ (ví dụ, 64×64). Trong các trường hợp khác, các nút lá QT có thể được phân vùng thêm thành cây đa loại. Do đó, nút lá QT là nút gốc cho cây đa loại, và nút lá QT có thể có giá trị chiều sâu cây đa loại ($mttDepth$) 0. Nếu chiều sâu cây đa loại đạt tới $MaxMttdepth$ (ví dụ, 4) thì việc phân vùng thêm nữa có thể không được xem xét nữa. Nếu chiều rộng của nút cây đa loại bằng $MinBtSize$ và nhỏ hơn hoặc bằng $2 \times MinTtSize$ thì việc không phân vùng ngang thêm nữa có thể được xem xét. Nếu chiều cao của nút cây đa loại bằng $MinBtSize$ và nhỏ hơn hoặc bằng $2 \times MinTtSize$ thì việc không phân vùng dọc thêm nữa có thể được xem xét. Khi việc phân vùng không được xem xét, thiết bị mã hóa có thể bỏ qua việc báo hiệu thông tin phân vùng. Trong trường hợp này, thiết bị giải mã có thể dẫn ra thông tin phân vùng bằng giá trị được xác định trước.

Trong khi đó, một CTU có thể bao gồm khối lập mã của các mẫu độ chói (ở dưới đây được gọi là “khối độ chói”) và hai khối lập mã của các mẫu sắc độ tương ứng với chúng (ở dưới đây được gọi là “các khối sắc độ”). Sơ đồ cây lập mã được mô tả trên đây có thể được áp dụng tương đương hoặc riêng biệt cho khối độ chói và khối sắc độ của CU hiện tại. Cụ thể, các khối độ chói và sắc độ trong một CTU có thể được phân vùng thành cấu trúc cây khối giống nhau và, trong trường hợp này, cấu trúc cây này có thể được thể hiện dưới dạng `SINGLE_TREE`. Theo cách khác, các khối độ chói và sắc độ trong một CTU có thể được phân vùng thành các cấu trúc cây khối riêng rẽ, và, trong trường hợp này, cấu trúc cây này có thể được thể hiện dưới dạng `DUAL_TREE`. Tức là, khi CTU được phân vùng thành các cây kép,

cấu trúc cây khối cho khối độ chói và cấu trúc cây khối cho khối sắc độ có thể hiện hữu riêng rẽ. Trong trường hợp này, cấu trúc cây khối cho khối độ chói có thể được gọi là DUAL_TREE_LUMA, và cấu trúc cây khối cho thành phần sắc độ có thể được gọi là DUAL_TREE_CHROMA. Đối với lát/các nhóm ô xếp P và B, các khối độ chói và sắc độ trong một CTU có thể được giới hạn để có cấu trúc cây lập mã giống nhau. Tuy nhiên, đối với lát/các nhóm ô xếp I, các khối độ chói và sắc độ có thể có cấu trúc cây khối riêng biệt với nhau. Nếu cấu trúc cây khối riêng biệt được áp dụng thì CTB độ chói có thể được phân vùng thành các CU dựa trên cấu trúc cây lập mã cụ thể, và CTB sắc độ có thể được phân vùng thành các CU sắc độ dựa trên một cấu trúc cây lập mã khác. Tức là, điều này nghĩa là CU trong lát/nhóm ô xếp I, mà cấu trúc cây khối riêng biệt được áp dụng với nó, có thể bao gồm khối lập mã của các thành phần độ chói hoặc các khối lập mã của hai thành phần sắc độ và CU của lát/nhóm ô xếp P hoặc B có thể bao gồm các khối của ba thành phần màu (thành phần độ chói và hai thành phần sắc độ).

Mặc dù cấu trúc cây lập mã cây tứ phân với cây đa loại lồng nhau đã được mô tả, cấu trúc trong đó CU được phân vùng không bị giới hạn vào đó. Ví dụ, cấu trúc BT và cấu trúc TT có thể được diễn giải như khái niệm được bao gồm trong cấu trúc phân vùng đa cây (Multiple partitioning tree, MPT), và CU có thể được diễn giải như là được phân vùng thông qua cấu trúc QT và cấu trúc MPT. Trong ví dụ trong đó CU được phân vùng qua cấu trúc QT và cấu trúc MPT, phần tử cú pháp (ví dụ, MPT_split_type) bao gồm thông tin về việc nút lá của cấu trúc QT này được phân vùng thành bao nhiêu khối và phần tử cú pháp (ví dụ, MPT_split_mode) bao gồm thông tin về việc nút lá của cấu trúc QT được phân vùng theo hướng nào trong các hướng dọc và ngang có thể được báo hiệu để xác định cấu trúc phân vùng.

Trong một ví dụ khác, CU có thể được phân vùng theo cách khác so với cấu trúc QT, cấu trúc BT hoặc cấu trúc TT. Tức là, không giống việc CU có chiều sâu thấp hơn được phân vùng thành 1/4 của CU có chiều sâu cao hơn theo cấu trúc QT, CU có chiều sâu thấp hơn được phân vùng thành 1/2 của CU có chiều sâu cao hơn theo cấu trúc BT, hoặc CU có chiều sâu thấp hơn

được phân vùng thành $1/4$ hoặc $1/2$ của CU có chiều sâu cao hơn theo cấu trúc TT, CU có chiều sâu thấp hơn có thể được phân vùng thành $1/5$, $1/3$, $3/8$, $3/5$, $2/3$, hoặc $5/8$ của CU có chiều sâu cao hơn trong một số trường hợp, và phương pháp phân vùng CU không bị giới hạn vào đó.

Cấu trúc khối lập mã cây tứ phân với cây đa loại có thể cung cấp cấu trúc phân vùng khối rất linh hoạt. Vì các loại phân vùng được hỗ trợ trong cây đa loại, các mẫu dạng phân vùng khác nhau có thể có tiềm năng tạo ra cấu trúc khối lập mã giống nhau trong một số trường hợp. Trong thiết bị mã hóa và thiết bị giải mã, qua việc hạn chế sự xuất hiện của các mẫu dạng phân vùng dư thừa này, lượng dữ liệu của thông tin phân vùng có thể được giảm xuống.

Thêm vào đó, khi mã hóa và giải mã video/ảnh theo sáng chế, đơn vị xử lý ảnh có thể có cấu trúc phân cấp. Một hình ảnh có thể được chia thành một hoặc nhiều ô xếp, viên gạch, lát và/hoặc nhóm ô xếp. Một lát có thể bao gồm một hoặc nhiều viên gạch. Một viên gạch có thể bao gồm một hoặc nhiều hàng CTU trong một ô xếp. Lát có thể bao gồm một số lượng nguyên các viên gạch của hình ảnh. Một nhóm ô xếp có thể bao gồm một hoặc nhiều ô xếp. Một ô xếp có thể bao gồm một hoặc nhiều CTU. CTU có thể được chia thành một hoặc nhiều CU. Ô xếp có thể là vùng hình chữ nhật bao gồm hàng ô xếp cụ thể và cột ô xếp cụ thể gồm nhiều CTU trong hình ảnh. Nhóm ô xếp có thể bao gồm một số lượng nguyên các ô xếp theo việc quét mảnh ô xếp trong hình ảnh. Tiêu đề lát có thể mang thông tin/các tham số áp dụng được cho lát (các khối trong lát) tương ứng. Khi thiết bị mã hóa hoặc thiết bị giải mã có bộ xử lý đa lõi, thủ tục mã hóa/giải mã cho ô xếp, lát, viên gạch và/hoặc nhóm ô xếp có thể được thực hiện song song.

Trong sáng chế, các tên hoặc các khái niệm của lát hoặc nhóm ô xếp có thể được sử dụng theo cách thay thế được cho nhau. Tức là, tiêu đề nhóm ô xếp có thể được gọi là tiêu đề lát. Ở đây, lát có thể có một trong các loại lát bao gồm lát nội (intra, I), lát dự đoán (predictive, P), và lát song dự đoán (bi-predictive, B). Đối với các khối trong lát I, việc dự đoán liên không được sử dụng để dự đoán và chỉ việc dự đoán nội là có thể được sử dụng. Tất nhiên, kể cả trong trường hợp này, giá trị mẫu gốc có thể được lập mã và được báo hiệu mà không có việc dự

đoán. Đối với các khối trong lát P, việc dự đoán nội hoặc dự đoán liên có thể được sử dụng. Khi việc dự đoán liên được sử dụng, chỉ việc đơn dự đoán có thể được sử dụng. Trong khi đó, đối với các khối trong lát B, việc dự đoán nội hoặc dự đoán liên có thể được sử dụng. Khi việc dự đoán liên được sử dụng, lên đến việc song dự đoán có thể được sử dụng.

Thiết bị mã hóa có thể xác định ô xếp/nhóm ô xếp, viên gạch, lát và kích thước đơn vị lập mã tối đa và tối thiểu theo các đặc điểm (ví dụ, độ phân giải) của ảnh video hoặc có xem xét đến hiệu suất lập mã và việc xử lý song song. Thêm vào đó, thông tin về điều này hoặc thông tin có khả năng dẫn ra điều này có thể được đưa vào trong luồng bit.

Thiết bị giải mã có thể thu nhận thông tin chỉ ra liệu ô xếp/nhóm ô xếp, viên gạch hoặc lát của hình ảnh hiện tại hoặc CTU trong ô xếp có được phân vùng thành nhiều đơn vị lập mã hay không. Thiết bị mã hóa và thiết bị giải mã có thể làm tăng hiệu suất mã hóa bằng cách báo hiệu thông tin này theo các điều kiện cụ thể.

Tiêu đề lát (cú pháp tiêu đề lát) có thể bao gồm thông tin/các tham số mà chúng có thể được áp dụng chung cho lát này. APS (cú pháp APS) hoặc PPS (cú pháp PPS) có thể bao gồm thông tin/các tham số mà chúng có thể được áp dụng chung cho một hoặc nhiều hình ảnh. SPS (cú pháp SPS) có thể bao gồm thông tin/các tham số mà chúng có thể được áp dụng chung cho một hoặc nhiều trình tự. VPS (cú pháp VPS) có thể bao gồm thông tin/các tham số mà chúng có thể được áp dụng chung cho nhiều lớp. DPS (cú pháp DPS) có thể bao gồm thông tin/các tham số mà chúng có thể được áp dụng chung cho video tổng thể. DPS có thể bao gồm thông tin/các tham số liên quan đến việc kết hợp của trình tự video được lập mã (coded video sequence, CVS).

Thêm vào đó, ví dụ, thông tin về việc phân vùng và cấu hình của ô xếp/nhóm ô xếp/viên gạch/lát có thể được xây dựng ở pha mã hóa qua cú pháp mức cao này và được truyền đến thiết bị giải mã dưới dạng luồng bit.

Tổng quan về việc dự đoán nội

Ở dưới đây, việc dự đoán nội được thực hiện bởi thiết bị mã hóa và giải mã được mô tả

trên đây sẽ được mô tả chi tiết hơn. Việc dự đoán nội có thể thể hiện việc dự đoán để tạo ra các mẫu dự đoán cho khối hiện tại dựa trên các mẫu tham chiếu trong hình ảnh mà khối hiện tại thuộc về hình ảnh này (ở dưới đây được gọi là hình ảnh hiện tại).

Phần mô tả sẽ được đưa ra với tham chiếu đến Fig.8. Khi việc dự đoán nội được áp dụng cho khối hiện tại 801, các mẫu tham chiếu lân cận cần được sử dụng cho việc dự đoán nội của khối hiện tại 801 có thể được dẫn ra. Các mẫu tham chiếu lân cận của khối hiện tại có thể bao gồm tổng cộng $2 \times n_H$ mẫu bao gồm các mẫu 811 liên kề với biên trái của khối hiện tại có kích thước $n_W \times n_H$ và các mẫu 812 liên kề với phía dưới cùng bên trái, tổng cộng $2 \times n_W$ mẫu bao gồm các mẫu 821 liên kề với biên trên cùng của khối hiện tại và các mẫu 822 liên kề với phía trên cùng bên phải, và một mẫu 831 liên kề với phía trên cùng bên trái của khối hiện tại. Theo cách khác, các mẫu tham chiếu lân cận của khối hiện tại có thể bao gồm nhiều cột mẫu lân cận trên cùng và nhiều hàng mẫu lân cận bên trái.

Thêm vào đó, các mẫu tham chiếu lân cận của khối hiện tại có thể bao gồm tổng cộng n_H mẫu 841 liên kề với biên phải của khối hiện tại có kích thước $n_W \times n_H$, tổng cộng n_W mẫu 851 liên kề với biên dưới cùng của khối hiện tại và một mẫu 842 liên kề với phía dưới cùng bên phải của khối hiện tại.

Tuy nhiên, một số mẫu trong số các mẫu tham chiếu lân cận của khối hiện tại vẫn chưa được giải mã hoặc có thể không khả dụng. Trong trường hợp này, thiết bị giải mã có thể xây dựng các mẫu tham chiếu lân cận cần được sử dụng cho việc dự đoán bằng cách thay thế các mẫu không khả dụng bằng các mẫu khả dụng. Theo cách khác, các mẫu tham chiếu lân cận cần được sử dụng cho việc dự đoán có thể được xây dựng qua việc nội suy các mẫu khả dụng.

Khi các mẫu tham chiếu lân cận được dẫn ra, (i) mẫu dự đoán có thể được dẫn ra dựa trên trung bình hoặc việc nội suy của các mẫu tham chiếu lân cận của khối hiện tại và (ii) mẫu dự đoán có thể được dẫn ra dựa trên mẫu tham chiếu có mặt trong hướng (dự đoán) cụ thể so với mẫu dự đoán này trong số các mẫu tham chiếu lân cận của khối hiện tại. Trường hợp (i) có thể được gọi là chế độ không hướng hoặc chế độ không có góc, và trường hợp (ii) có thể được gọi

là chế độ có hướng hoặc chế độ có góc. Thêm vào đó, mẫu dự đoán có thể được tạo ra thông qua việc nội suy giữa mẫu lân cận thứ nhất và mẫu lân cận thứ hai được đặt theo hướng ngược với hướng dự đoán của chế độ dự đoán nội của khối hiện tại dựa trên mẫu dự đoán của khối hiện tại trong số các mẫu tham chiếu lân cận. Trường hợp được mô tả trên đây có thể được gọi là việc dự đoán nội nội suy tuyến tính (linear interpolation intra prediction, LIP). Thêm vào đó, các mẫu dự đoán sắc độ có thể được tạo ra dựa trên các mẫu độ chói nhờ sử dụng mô hình tuyến tính. Trường hợp này có thể được gọi là chế độ LM. Ngoài ra, mẫu dự đoán tạm thời của khối hiện tại có thể được dẫn ra dựa trên các mẫu tham chiếu lân cận được lọc, và mẫu dự đoán của khối hiện tại có thể được dẫn ra bằng cách lấy tổng được định trọng số mẫu dự đoán tạm thời này và ít nhất một mẫu tham chiếu được dẫn ra theo chế độ dự đoán nội trong số các mẫu tham chiếu lân cận đã có, nghĩa là, các mẫu tham chiếu lân cận không được lọc. Trường hợp được mô tả trên đây có thể được gọi là việc dự đoán nội phụ thuộc vào vị trí (position dependent intra prediction, PDPC). Thêm vào đó, đường mẫu tham chiếu với độ chính xác dự đoán cao nhất có thể được chọn từ trong số nhiều đường mẫu tham chiếu lân cận của khối hiện tại và mẫu dự đoán có thể được dẫn ra nhờ sử dụng mẫu tham chiếu được đặt theo hướng dự đoán trên đường tương ứng. Tại thời điểm này, việc mã hóa dự đoán nội có thể được thực hiện bằng cách chỉ dẫn (báo hiệu) đường mẫu tham chiếu được sử dụng cho thiết bị giải mã. Trường hợp được mô tả trên đây có thể được gọi là việc dự đoán nội nhiều đường tham chiếu (multi-reference line, MRL) hoặc việc dự đoán nội dựa trên MRL. Thêm vào đó, khối hiện tại có thể được phân vùng thành các phân vùng con dọc hoặc ngang, việc dự đoán nội có thể được thực hiện dựa trên cùng một chế độ dự đoán nội, và các mẫu tham chiếu lân cận có thể được dẫn ra và được sử dụng theo các đơn vị là các phân vùng con. Tức là, trong trường hợp này, chế độ dự đoán nội cho khối hiện tại được áp dụng tương đương cho các phân vùng con, và các mẫu tham chiếu lân cận có thể được dẫn ra và được sử dụng theo các đơn vị là các phân vùng con, qua đó làm tăng hiệu năng dự đoán nội trong một số trường hợp. Phương pháp dự đoán này có thể được gọi là việc dự đoán phân vùng con nội (intra sub-partition, ISP) hoặc việc dự đoán nội

dựa trên ISP. Các phương pháp dự đoán nội này có thể được gọi là loại dự đoán nội, để được phân biệt với chế độ dự đoán nội (ví dụ, chế độ DC, chế độ phẳng (Planar) hoặc chế độ có hướng). Loại dự đoán nội có thể được gọi bằng nhiều thuật ngữ khác nhau như kỹ thuật dự đoán nội hoặc chế độ dự đoán nội bổ sung. Ví dụ, loại dự đoán nội (hoặc chế độ dự đoán nội bổ sung, v.v.) này có thể bao gồm ít nhất một thành phần trong số LIP, PDPC, MRL, và ISP được mô tả trên đây. Phương pháp dự đoán nội bình thường ngoại trừ các loại dự đoán nội cụ thể như LIP, PDPC, MRL và ISP có thể được gọi là loại dự đoán nội bình thường. Loại dự đoán nội bình thường có thể chỉ trường hợp trong đó loại dự đoán nội cụ thể trên đây không được áp dụng, và việc dự đoán có thể được thực hiện dựa trên chế độ dự đoán nội được mô tả trên đây. Trong khi đó, việc lọc sau có thể được thực hiện đối với mẫu dự đoán được dẫn ra khi cần.

Cụ thể, thủ tục dự đoán nội có thể bao gồm bước xác định chế độ/loại dự đoán nội, bước dẫn ra mẫu tham chiếu lân cận và bước dẫn ra mẫu dự đoán dựa trên chế độ/loại dự đoán nội. Thêm vào đó, việc lọc sau có thể được thực hiện đối với mẫu dự đoán được dẫn ra khi cần.

Trong khi đó, ngoài các loại dự đoán nội được mô tả trên đây, việc dự đoán nội được định trọng số tuyến tính afin (affine linear weighted intra prediction, ALWIP) có thể được sử dụng. ALWIP có thể được gọi là việc dự đoán nội được định trọng số tuyến tính (linear weighted intra prediction, LWIP), việc dự đoán nội được định trọng số ma trận (matrix weighted intra prediction, MIP) hoặc việc dự đoán nội dựa trên ma trận. Khi MIP được áp dụng cho khối hiện tại, i) nhờ sử dụng các mẫu tham chiếu lân cận được trải qua thủ tục lấy trung bình, ii) thủ tục nhân vector ma trận có thể được thực hiện và iii) thủ tục nội suy ngang/dọc có thể được thực hiện thêm khi cần, qua đó dẫn ra các mẫu dự đoán cho khối hiện tại. Các chế độ dự đoán nội được sử dụng cho MIP có thể được xây dựng khác với các chế độ dự đoán nội được sử dụng trong việc dự đoán nội LIP, PDPC, MRL, ISP hoặc việc dự đoán nội bình thường được mô tả trên đây. Chế độ dự đoán nội cho MIP có thể được gọi là chế độ dự đoán nội MIP, chế độ dự đoán MIP hoặc chế độ MIP. Ví dụ, ma trận và phần bù (offset) được sử dụng trong phép nhân

vector ma trận có thể được thiết lập khác nhau tùy theo chế độ dự đoán nội cho MIP. Ở đây, ma trận này có thể được gọi là ma trận trọng số (MIP), và phần bù này có thể được gọi là vector phần bù (MIP) hoặc vector định thiên (MIP). Phương pháp MIP chi tiết sẽ được mô tả dưới đây.

Thủ tục xây dựng lại khối dựa trên việc dự đoán nội và đơn vị dự đoán nội trong thiết bị mã hóa có thể sơ lược bao gồm, ví dụ, nội dung sau đây. S910 có thể được thực hiện bởi bộ dự đoán nội 185 của thiết bị mã hóa, và S920 có thể được thực hiện bởi bộ xử lý phần dư bao gồm ít nhất một thành phần trong số bộ trừ 115, bộ biến đổi 120, bộ lượng tử hóa 130, bộ giải lượng tử hóa 140 và bộ biến đổi ngược 150 của thiết bị mã hóa. Cụ thể, S920 có thể được thực hiện bởi bộ trừ 115 của thiết bị mã hóa. Trong S930, thông tin dự đoán có thể được dẫn ra bởi bộ dự đoán nội 185, và có thể được mã hóa bởi bộ mã hóa entropy 190. Trong S930, thông tin phần dư có thể được dẫn ra bởi bộ xử lý phần dư, và có thể được mã hóa bởi bộ mã hóa entropy 190. Thông tin phần dư là thông tin về các mẫu phần dư. Thông tin phần dư có thể bao gồm thông tin về các hệ số biến đổi được lượng tử hóa cho các mẫu phần dư. Như được mô tả trên đây, các mẫu phần dư có thể được dẫn ra dưới dạng các hệ số biến đổi thông qua bộ biến đổi 120 của thiết bị mã hóa, và các hệ số biến đổi có thể được dẫn ra dưới dạng các hệ số biến đổi được lượng tử hóa thông qua bộ lượng tử hóa 130. Thông tin về các hệ số biến đổi được lượng tử hóa có thể được mã hóa bởi bộ mã hóa entropy 190 thông qua thủ tục lập mã phần dư.

Thiết bị mã hóa có thể thực hiện việc dự đoán nội đối với khối hiện tại (S910). Thiết bị mã hóa có thể dẫn ra chế độ/loại dự đoán nội cho khối hiện tại, dẫn ra các mẫu tham chiếu lân cận của khối hiện tại, và tạo ra các mẫu dự đoán trong khối hiện tại dựa trên chế độ/loại dự đoán nội và các mẫu tham chiếu lân cận này. Ở đây, thủ tục để xác định chế độ/loại dự đoán nội, thủ tục để dẫn ra các mẫu tham chiếu lân cận và thủ tục để tạo ra các mẫu dự đoán có thể được thực hiện đồng thời, hoặc một thủ tục bất kỳ có thể được thực hiện trước thủ tục khác. Ví dụ, mặc dù không được thể hiện, bộ dự đoán nội 185 của thiết bị mã hóa có thể bao gồm đơn vị xác định chế độ/loại dự đoán nội, đơn vị dẫn ra mẫu tham chiếu, đơn vị dẫn ra mẫu dự đoán. Đơn vị xác định chế độ/loại dự đoán nội có thể xác định chế độ/loại dự đoán nội cho khối hiện tại,

đơn vị dẫn ra mẫu tham chiếu có thể dẫn ra các mẫu tham chiếu lân cận của khối hiện tại, và đơn vị dẫn ra mẫu dự đoán có thể dẫn ra các mẫu dự đoán của khối hiện tại. Trong khi đó, khi thủ tục lọc mẫu dự đoán được mô tả dưới đây được thực hiện, bộ dự đoán nội 185 có thể còn bao gồm bộ lọc mẫu dự đoán. Thiết bị mã hóa có thể xác định chế độ/loại được áp dụng cho khối hiện tại từ trong số nhiều chế độ/loại dự đoán nội. Thiết bị mã hóa có thể so sánh các chi phí RD của các chế độ/loại dự đoán nội và xác định chế độ/loại dự đoán nội tối ưu cho khối hiện tại.

Trong khi đó, thiết bị mã hóa có thể thực hiện thủ tục lọc mẫu dự đoán. Việc lọc mẫu dự đoán có thể được gọi là việc lọc sau. Một số hoặc tất cả các mẫu dự đoán có thể được lọc bởi thủ tục lọc mẫu dự đoán. Trong một số trường hợp, thủ tục lọc mẫu dự đoán có thể được lược bỏ.

Thiết bị mã hóa có thể tạo ra các mẫu phần dư cho khối hiện tại dựa trên các mẫu dự đoán (được lọc) (S920). Thiết bị mã hóa có thể so sánh các mẫu dự đoán từ các mẫu gốc của khối hiện tại dựa trên pha và dẫn ra các mẫu phần dư.

Thiết bị mã hóa có thể mã hóa thông tin ảnh bao gồm thông tin về việc dự đoán nội (thông tin dự đoán) và thông tin phần dư của các mẫu phần dư (S930). Thông tin dự đoán có thể bao gồm thông tin chế độ dự đoán nội và thông tin loại dự đoán nội. Thiết bị mã hóa có thể xuất ra thông tin ảnh được mã hóa dưới dạng luồng bit. Luồng bit được xuất ra có thể được truyền đến thiết bị giải mã qua phương tiện lưu trữ hoặc mạng.

Thông tin phần dư có thể bao gồm cú pháp lập mã phần dư được mô tả dưới đây. Thiết bị mã hóa có thể biến đổi/lượng tử hóa các mẫu phần dư để dẫn ra các hệ số biến đổi được lượng tử hóa. Thông tin phần dư có thể bao gồm thông tin về các hệ số biến đổi được lượng tử hoá.

Trong khi đó, như được mô tả trên đây, thiết bị mã hóa có thể tạo ra hình ảnh được xây dựng lại (bao gồm các mẫu được xây dựng lại và khối được xây dựng lại). Về vấn đề này, thiết bị mã hóa có thể thực hiện việc giải lượng tử hóa/biến đổi ngược đối với các hệ số biến đổi

được lượng tử hóa lần nữa để dẫn ra các mẫu phần dư (được cải biến). Các mẫu phần dư được biến đổi/được lượng tử hóa và sau đó được giải lượng tử hóa/được biến đổi ngược, để dẫn ra các mẫu phần dư giống như các mẫu phần dư được dẫn ra trong thiết bị giải mã như được mô tả trên đây. Thiết bị mã hóa có thể tạo ra khối được xây dựng lại bao gồm các mẫu được xây dựng lại cho khối hiện tại dựa trên các mẫu dự đoán và các mẫu phần dư (được cải biến). Hình ảnh được xây dựng lại cho hình ảnh hiện tại có thể được tạo ra dựa trên khối được xây dựng lại. Như được mô tả trên đây, thủ tục lọc trong vòng có thể còn áp dụng được cho hình ảnh được xây dựng lại.

Thủ tục giải mã video/ảnh dựa trên việc dự đoán nội và bộ dự đoán nội trong thiết bị giải mã, ví dụ, có thể sơ lược bao gồm nội dung sau đây. Thiết bị giải mã có thể thực hiện hoạt động tương ứng với hoạt động được thực hiện trong thiết bị mã hóa.

S1010 đến S1030 có thể được thực hiện bởi bộ dự đoán nội 265 của thiết bị giải mã, và thông tin dự đoán của S1010 và thông tin phần dư của S1040 có thể được thu nhận từ luồng bit bởi bộ giải mã entropy 210 của thiết bị giải mã. Bộ xử lý phần dư bao gồm ít nhất một thành phần trong số bộ giải lượng tử hóa 220 và bộ biến đổi ngược 230 của thiết bị giải mã có thể dẫn ra các mẫu phần dư cho khối hiện tại dựa trên thông tin phần dư. Cụ thể, bộ giải lượng tử hóa 220 của bộ xử lý phần dư có thể thực hiện việc giải lượng tử hóa dựa trên các hệ số biến đổi được lượng tử hóa được dẫn ra dựa trên thông tin phần dư để dẫn ra các hệ số biến đổi và bộ giải lượng tử hóa 220 của bộ xử lý phần dư có thể thực hiện việc biến đổi ngược đối với các hệ số biến đổi để dẫn ra các mẫu phần dư cho khối hiện tại. S1050 có thể được thực hiện bởi bộ cộng 235 hoặc bộ xây dựng lại của thiết bị giải mã.

Cụ thể, thiết bị giải mã có thể dẫn ra chế độ/loại dự đoán nội cho khối hiện tại dựa trên thông tin dự đoán nhận được (thông tin chế độ/loại dự đoán nội) (S1010). Thiết bị giải mã có thể dẫn ra các mẫu tham chiếu lân cận của khối hiện tại (S1020). Thiết bị giải mã có thể tạo ra các mẫu dự đoán trong các khối hiện tại dựa trên chế độ/loại dự đoán nội và các mẫu tham chiếu lân cận (S1030). Trong trường hợp này, thiết bị giải mã có thể thực hiện thủ tục lọc mẫu

dự đoán. Việc lọc mẫu dự đoán có thể được gọi là việc lọc sau. Một số hoặc tất cả các mẫu dự đoán có thể được lọc bởi thủ tục lọc mẫu dự đoán. Trong một số trường hợp, thủ tục lọc mẫu dự đoán có thể được lược bỏ.

Thiết bị giải mã có thể tạo ra các mẫu phần dư cho khối hiện tại dựa trên thông tin phần dư nhận được. Thiết bị giải mã có thể tạo ra các mẫu được xây dựng lại cho khối hiện tại dựa trên các mẫu dự đoán và các mẫu phần dư, và dẫn ra khối xây dựng lại bao gồm các mẫu được xây dựng lại này (S1040). Hình ảnh được xây dựng lại cho hình ảnh hiện tại có thể được tạo ra dựa trên khối được xây dựng lại. Như được mô tả trên đây, thủ tục lọc trong vòng có thể còn áp dụng được cho hình ảnh được xây dựng lại.

Ở đây, mặc dù không được thể hiện, bộ dự đoán nội 265 của thiết bị giải mã có thể bao gồm đơn vị xác định chế độ/loại dự đoán nội, đơn vị dẫn ra mẫu tham chiếu và đơn vị dẫn ra mẫu dự đoán, đơn vị xác định chế độ/loại dự đoán nội có thể xác định chế độ/loại nội cho khối hiện tại dựa trên thông tin chế độ/loại dự đoán nội được thu nhận bởi bộ giải mã entropy 210, đơn vị dẫn ra mẫu tham chiếu có thể dẫn ra các mẫu tham chiếu lân cận của khối hiện tại, và đơn vị dẫn ra mẫu dự đoán có thể dẫn ra các mẫu dự đoán của khối hiện tại. Trong khi đó, khi thủ tục lọc mẫu dự đoán được mô tả trên đây được thực hiện, bộ dự đoán nội 265 có thể còn bao gồm bộ lọc mẫu dự đoán.

Thông tin chế độ dự đoán nội có thể bao gồm thông tin cờ (ví dụ, `intra_luma_mpm_flag`) chỉ ra liệu chế độ khả dĩ nhất (most probable mode, MPM) hay chế độ còn lại được áp dụng cho khối hiện tại, và khi MPM được áp dụng cho khối hiện tại, thông tin chế độ dự đoán có thể còn bao gồm thông tin chỉ số (ví dụ, `intra_luma_mpm_idx`) chỉ ra một ứng viên trong số các ứng viên chế độ dự đoán nội (các ứng viên MPM). Các ứng viên chế độ dự đoán nội (các ứng viên MPM) có thể được tạo cấu hình dưới dạng danh sách ứng viên MPM hoặc danh sách MPM. Thêm vào đó, khi MPM này không được áp dụng cho khối hiện tại, thông tin chế độ dự đoán nội có thể còn bao gồm thông tin chế độ còn lại (ví dụ, `intra_luma_mpm_remainder`) chỉ ra một chế độ trong số các chế độ dự đoán nội còn lại, ngoại trừ các ứng viên chế độ dự đoán

nội (các ứng viên MPM) đã nêu. Thiết bị giải mã có thể xác định chế độ dự đoán nội của khối hiện tại dựa trên thông tin chế độ dự đoán nội này. Danh sách MPM riêng rẽ có thể được tạo cấu hình cho MIP được mô tả trên đây.

Thêm vào đó, thông tin loại dự đoán nội có thể được thi hành dưới các dạng khác nhau. Ví dụ, thông tin loại dự đoán nội có thể bao gồm thông tin chỉ số loại dự đoán nội chỉ ra một loại trong số các loại dự đoán nội. Như là một ví dụ khác, thông tin loại dự đoán nội có thể bao gồm ít nhất một thông tin trong số thông tin đường mẫu tham chiếu (ví dụ, `intra_luma_ref_idx`) chỉ ra liệu MRL có được áp dụng cho khối hiện tại hay không và đường mẫu tham chiếu thứ bao nhiêu được sử dụng nếu được áp dụng, thông tin cờ ISP (ví dụ, `intra_subpartitions_mode_flag`) chỉ ra liệu ISP có được áp dụng cho khối hiện tại hay không, thông tin loại ISP (ví dụ, `intra_subpartitions_split_flag`) chỉ ra loại chia tách của các phân vùng con khi ISP được áp dụng, thông tin cờ chỉ ra liệu PDCP có được áp dụng hay không hoặc thông tin cờ chỉ ra liệu LIP có được áp dụng hay không. Thêm vào đó, thông tin loại dự đoán nội có thể bao gồm cờ MIP chỉ ra liệu MIP có được áp dụng cho khối hiện tại hay không.

Thông tin chế độ dự đoán nội và/hoặc thông tin loại dự đoán nội có thể được mã hóa/giải mã qua phương pháp lập mã được mô tả trong sáng chế này. Ví dụ, thông tin chế độ dự đoán nội và/hoặc thông tin loại dự đoán nội có thể được mã hóa/giải mã thông qua việc lập mã entropy (ví dụ, CABAC hoặc CAVLC) dựa trên mã nhị phân (rice) được cắt cụt.

Tổng quan về việc dự đoán liên

Ở dưới đây, công nghệ chi tiết về việc dự đoán liên trong phần mô tả về việc mã hóa và giải mã với tham chiếu đến Fig.2 và Fig.3 sẽ được mô tả. Trong trường hợp thiết bị giải mã, phương pháp giải mã video/ảnh dựa trên việc dự đoán liên và bộ dự đoán liên trong thiết bị giải mã có thể vận hành theo phần mô tả sau đây. Trong trường hợp thiết bị mã hóa, phương pháp mã hóa video/ảnh dựa trên việc dự đoán liên và bộ dự đoán liên trong thiết bị mã hóa có thể vận hành theo phần mô tả sau đây. Thêm vào đó, trong phần mô tả sau đây, dữ liệu được mã hóa bằng phần mô tả sau đây có thể được lưu trữ dưới dạng luồng bit.

Bộ dự đoán của thiết bị mã hóa/giải mã có thể thực hiện việc dự đoán liên theo các đơn vị là các khối để dẫn ra mẫu dự đoán. Việc dự đoán liên có thể thể hiện việc dự đoán được dẫn ra theo phương pháp phụ thuộc vào các phần tử dữ liệu (ví dụ, các giá trị mẫu, thông tin chuyển động, v.v.) của hình ảnh (các hình ảnh) khác ngoài hình ảnh hiện tại. Khi việc dự đoán liên được áp dụng cho khối hiện tại, khối được dự đoán (mảng mẫu dự đoán) cho khối hiện tại có thể được dẫn ra dựa trên khối tham chiếu (mảng mẫu tham chiếu) được chỉ rõ bởi vector chuyển động trên hình ảnh tham chiếu được chỉ ra bởi chỉ số hình ảnh tham chiếu. Trong trường hợp này, để làm giảm lượng thông tin chuyển động được truyền trong chế độ dự đoán liên, thông tin chuyển động của khối hiện tại có thể được dự đoán theo các đơn vị là các khối, các khối con hoặc các mẫu dựa trên sự tương quan của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể bao gồm vector chuyển động và chỉ số hình ảnh tham chiếu. Thông tin chuyển động có thể còn bao gồm thông tin loại dự đoán liên (dự đoán L0, dự đoán L1, song dự đoán, v.v.). Khi việc dự đoán liên được áp dụng, các khối lân cận có thể bao gồm khối lân cận theo không gian được đặt trong hình ảnh hiện tại và khối lân cận theo thời gian được đặt trong hình ảnh tham chiếu. Hình ảnh tham chiếu chứa khối tham chiếu và hình ảnh tham chiếu chứa khối lân cận theo thời gian có thể là giống nhau hoặc khác nhau. Khối lân cận theo thời gian có thể được gọi là khối tham chiếu được đặt cùng vị trí hoặc CU được đặt cùng vị trí (collocated CU hay colCU), và hình ảnh tham chiếu chứa khối lân cận theo thời gian có thể được gọi là hình ảnh được đặt cùng vị trí (collocated picture, colPic). Ví dụ, danh sách ứng viên thông tin chuyển động có thể được tạo cấu hình dựa trên các khối lân cận của khối hiện tại, và cờ hoặc thông tin chỉ số chỉ ra ứng viên nào được chọn (được sử dụng) để dẫn ra vector chuyển động của khối hiện tại và/hoặc chỉ số hình ảnh tham chiếu có thể được báo hiệu. Việc dự đoán liên có thể được thực hiện dựa trên các chế độ dự đoán khác nhau và ví dụ, trong trường hợp chế độ bỏ qua và chế độ hợp nhất, thông tin chuyển động của khối hiện tại có thể là giống như thông tin chuyển động của khối lân cận được chọn. Trong trường hợp chế độ bỏ qua, không giống chế độ hợp nhất, tín hiệu phần dư có thể không được truyền. Trong trường

hợp chế độ dự đoán thông tin chuyển động (motion vector prediction, MVP), vector chuyển động của khối lân cận được chọn có thể được sử dụng làm bộ dự đoán vector chuyển động, và chênh lệch vector chuyển động có thể được báo hiệu. Trong trường hợp này, vector chuyển động của khối hiện tại có thể được dẫn ra nhờ sử dụng tổng của bộ dự đoán vector chuyển động và chênh lệch vector chuyển động.

Thông tin chuyển động có thể bao gồm thông tin chuyển động L0 và/hoặc thông tin chuyển động L1 tùy theo loại dự đoán liên (dự đoán L0, dự đoán L1, song dự đoán, v.v.). Vector chuyển động theo hướng L0 có thể được gọi là vector chuyển động L0 hoặc MVL0, và vector chuyển động theo hướng L1 có thể được gọi là vector chuyển động L1 hoặc MVL1. Việc dự đoán dựa trên vector chuyển động L0 có thể được gọi là việc dự đoán L0, việc dự đoán dựa trên vector chuyển động L1 có thể được gọi là việc dự đoán L1, và việc dự đoán dựa trên cả vector chuyển động L0 và vector chuyển động L1 có thể được gọi là việc song dự đoán. Ở đây, vector chuyển động L0 có thể thể hiện vector chuyển động được liên kết với danh sách hình ảnh tham chiếu L0 (L0), và vector chuyển động L1 có thể thể hiện vector chuyển động được liên kết với danh sách hình ảnh tham chiếu L1 (L1). Danh sách hình ảnh tham chiếu L0 có thể bao gồm các hình ảnh trước hình ảnh hiện tại theo thứ tự xuất ra để làm các hình ảnh tham chiếu, và danh sách hình ảnh tham chiếu L1 có thể bao gồm các hình ảnh sau hình ảnh hiện tại theo thứ tự xuất ra. Các hình ảnh trước khối hiện tại có thể được gọi là các hình ảnh (tham chiếu) thuận chiều và các hình ảnh sau khối hiện tại có thể được gọi là các hình ảnh (tham chiếu) ngược chiều. Danh sách hình ảnh tham chiếu L0 có thể còn bao gồm các hình ảnh sau hình ảnh hiện tại theo thứ tự xuất ra để làm các hình ảnh tham chiếu. Trong trường hợp này, trong danh sách hình ảnh tham chiếu L0, các hình ảnh trước khối hiện tại được đánh chỉ số trước và sau đó các hình ảnh sau khối hiện tại được đánh chỉ số. Danh sách hình ảnh tham chiếu L1 có thể còn bao gồm các hình ảnh trước hình ảnh hiện tại theo thứ tự xuất ra để làm các hình ảnh tham chiếu. Trong trường hợp này, trong danh sách hình ảnh tham chiếu 1, các hình ảnh sau khối hiện tại có thể được đánh chỉ số trước và sau đó các hình ảnh trước khối hiện tại có thể được đánh chỉ số. Ở

đây, thứ tự xuất ra có thể tương ứng với thứ tự số đếm thứ tự hình ảnh (picture order count, POC).

Thủ tục mã hóa video/ảnh dựa trên việc dự đoán liên và bộ dự đoán liên trong thiết bị mã hóa, ví dụ, có thể sơ lược là như sau. Phần mô tả sẽ được đưa ra với tham chiếu đến Fig.11. Thiết bị mã hóa thực hiện việc dự đoán liên đối với khối hiện tại (S1110). Thiết bị mã hóa có thể dẫn ra chế độ dự đoán liên và thông tin chuyển động của khối hiện tại, và tạo ra các mẫu dự đoán của khối hiện tại. Ở đây, các thủ tục xác định chế độ dự đoán liên, dẫn ra thông tin chuyển động và dẫn ra mẫu dự đoán có thể được thực hiện đồng thời và thủ tục bất kỳ có thể được thực hiện trước thủ tục khác. Ví dụ, bộ dự đoán liên của thiết bị mã hóa có thể bao gồm đơn vị xác định chế độ dự đoán, đơn vị dẫn ra thông tin chuyển động và đơn vị dẫn ra mẫu dự đoán, đơn vị xác định chế độ dự đoán có thể xác định chế độ dự đoán cho khối hiện tại, đơn vị dẫn ra thông tin chuyển động có thể dẫn ra thông tin chuyển động của khối hiện tại, và đơn vị dẫn ra mẫu dự đoán có thể dẫn ra các mẫu dự đoán của khối hiện tại. Ví dụ, bộ dự đoán liên của thiết bị mã hóa có thể tìm kiếm khối tương tự với khối hiện tại bên trong vùng nhất định (vùng tìm kiếm) của các hình ảnh tham chiếu thông qua việc ước lượng chuyển động, và dẫn ra khối tham chiếu mà chênh lệch của nó so với khối hiện tại là tối thiểu hoặc nhỏ hơn hoặc bằng tiêu chí nhất định. Dựa trên điều này, chỉ số hình ảnh tham chiếu chỉ ra hình ảnh tham chiếu mà khối tham chiếu được đặt trong đó có thể được dẫn ra, và vector chuyển động có thể được dẫn ra dựa trên chênh lệch về vị trí giữa khối tham chiếu và khối hiện tại. Thiết bị mã hóa có thể xác định chế độ được áp dụng cho khối hiện tại trong số các chế độ dự đoán khác nhau. Thiết bị mã hóa có thể so sánh các chi phí RD cho các chế độ dự đoán khác nhau này và xác định chế độ dự đoán tối ưu cho khối hiện tại.

Ví dụ, thiết bị mã hóa có thể xây dựng danh sách ứng viên hợp nhất, mà nó sẽ được mô tả dưới đây, khi chế độ bỏ qua hoặc chế độ hợp nhất được áp dụng cho khối hiện tại, và dẫn ra khối tham chiếu mà chênh lệch của nó so với khối hiện tại là tối thiểu hoặc nhỏ hơn hoặc bằng tiêu chí nhất định trong số các khối tham chiếu được chỉ ra bởi các ứng viên hợp nhất được

chứa trong danh sách ứng viên hợp nhất. Trong trường hợp này, ứng viên hợp nhất liên quan đến khối tham chiếu được dẫn ra có thể được chọn, và thông tin chỉ số hợp nhất chỉ ra ứng viên hợp nhất được chọn này có thể được tạo ra và được báo hiệu đến thiết bị giải mã. Thông tin chuyển động của khối hiện tại có thể được dẫn ra nhờ sử dụng thông tin chuyển động của ứng viên hợp nhất được chọn.

Như là một ví dụ khác, thiết bị mã hóa có thể xây dựng danh sách ứng viên (A)MVP, mà nó sẽ được mô tả dưới đây, khi chế độ (A)MVP được áp dụng cho khối hiện tại, và sử dụng vector chuyển động của ứng viên bộ dự đoán vector chuyển động (motion vector predictor, mvp) được chọn từ trong số các ứng viên mvp được chứa trong danh sách ứng viên (A)MVP làm mvp của khối hiện tại. Trong trường hợp này, vector chuyển động chỉ ra khối tham chiếu được dẫn ra bằng việc ước lượng chuyển động có thể được sử dụng làm vector chuyển động của khối hiện tại, và ứng viên mvp có vector chuyển động mà chênh lệch của nó so với vector chuyển động của khối hiện tại là nhỏ nhất trong số các ứng viên mvp có thể là ứng viên mvp được chọn. Chênh lệch vector chuyển động (MVD) được thu bằng cách lấy vector chuyển động của khối hiện tại trừ đi mvp có thể được dẫn ra. Trong trường hợp này, thông tin về MVD có thể được báo hiệu đến thiết bị giải mã. Thêm vào đó, khi chế độ (A)MVP được áp dụng, giá trị của chỉ số hình ảnh tham chiếu có thể được xây dựng dưới dạng thông tin chỉ số hình ảnh tham chiếu và được báo hiệu đến thiết bị giải mã.

Thiết bị mã hóa có thể dẫn ra các mẫu phần dư dựa trên các mẫu dự đoán (S1120). Thiết bị mã hóa có thể dẫn ra các mẫu phần dư thông qua việc so sánh giữa các mẫu gốc của khối hiện tại và các mẫu dự đoán.

Thiết bị mã hóa mã hóa thông tin ảnh bao gồm thông tin dự đoán và thông tin phần dư (S1130). Thiết bị mã hóa có thể xuất ra thông tin ảnh được mã hóa dưới dạng luồng bit. Thông tin dự đoán có thể bao gồm thông tin chế độ dự đoán (ví dụ, cờ bỏ qua, cờ hợp nhất, chỉ số chế độ, v.v.) và thông tin chuyển động để làm thông tin liên quan đến thủ tục dự đoán. Thông tin chuyển động có thể bao gồm thông tin chọn ứng viên (ví dụ, chỉ số hợp nhất, cờ mvp hoặc chỉ

số mvp) mà nó là thông tin để dẫn ra vector chuyển động. Thêm vào đó, thông tin chuyển động có thể bao gồm thông tin về MVD và/hoặc thông tin chỉ số hình ảnh tham chiếu được mô tả trên đây. Thêm vào đó, thông tin chuyển động có thể bao gồm thông tin chỉ ra liệu việc dự đoán L0, việc dự đoán L1, hay việc song dự đoán được áp dụng. Thông tin phần dư là thông tin về các mẫu phần dư. Thông tin phần dư có thể bao gồm thông tin về các hệ số biến đổi được lượng tử hóa cho các mẫu phần dư.

Luồng bit được xuất ra có thể được lưu trữ trong phương tiện lưu trữ (kỹ thuật số) và được truyền đến thiết bị giải mã hoặc có thể được truyền đến thiết bị giải mã qua mạng.

Trong khi đó, như được mô tả trên đây, thiết bị mã hóa có thể tạo ra hình ảnh được xây dựng lại (bao gồm các mẫu được xây dựng lại và khối được xây dựng lại) dựa trên các mẫu tham chiếu và các mẫu phần dư. Điều này là để cho phép thiết bị giải mã dẫn ra kết quả dự đoán giống như kết quả được thực hiện trong thiết bị mã hóa, qua đó làm tăng hiệu suất lập mã. Theo đó, thiết bị mã hóa có thể lưu trữ hình ảnh được xây dựng lại (hoặc các mẫu được xây dựng lại hoặc khối được xây dựng lại) trong bộ nhớ và sử dụng hình ảnh được xây dựng lại này làm hình ảnh tham chiếu cho việc dự đoán liên. Như được mô tả trên đây, thủ tục lọc trong vòng có thể còn áp dụng được cho hình ảnh được xây dựng lại.

Thủ tục giải mã video/ảnh dựa trên việc dự đoán liên và bộ dự đoán liên trong thiết bị giải mã, ví dụ, có thể sơ lược bao gồm nội dung sau đây.

Thiết bị giải mã có thể thực hiện hoạt động tương ứng với hoạt động được thực hiện bởi thiết bị mã hóa. Thiết bị giải mã có thể thực hiện việc dự đoán đối với khối hiện tại dựa trên thông tin dự đoán nhận được và dẫn ra các mẫu dự đoán.

Cụ thể, thiết bị giải mã có thể xác định chế độ dự đoán cho khối hiện tại dựa trên thông tin dự đoán nhận được (S1210). Thiết bị giải mã có thể xác định chế độ dự đoán liên nào được áp dụng cho khối hiện tại dựa trên thông tin chế độ dự đoán trong thông tin dự đoán.

Ví dụ, việc liệu chế độ hợp nhất hay chế độ (A)MVP được áp dụng cho khối hiện tại có thể được xác định dựa trên cờ hợp nhất. Theo cách khác, một ứng viên trong số các ứng viên

chế độ dự đoán liên khác nhau có thể được chọn dựa trên chỉ số chế độ. Các ứng viên chế độ dự đoán liên có thể bao gồm chế độ bỏ qua, chế độ hợp nhất và/hoặc chế độ (A)MVP hoặc có thể bao gồm các chế độ dự đoán liên khác nhau mà chúng sẽ được mô tả dưới đây.

Thiết bị giải mã dẫn ra thông tin chuyển động của khối hiện tại dựa trên chế độ dự đoán liên được xác định (S1220). Ví dụ, thiết bị giải mã có thể xây dựng danh sách ứng viên hợp nhất được mô tả dưới đây và chọn một ứng viên trong số các ứng viên hợp nhất được chứa trong danh sách ứng viên hợp nhất, khi chế độ bỏ qua hoặc chế độ hợp nhất được áp dụng cho khối hiện tại. Việc chọn này có thể được thực hiện dựa trên thông tin chọn ứng viên (chỉ số hợp nhất) được mô tả trên đây. Thông tin chuyển động của khối hiện tại có thể được dẫn ra nhờ sử dụng thông tin chuyển động của ứng viên hợp nhất được chọn này. Thông tin chuyển động của ứng viên hợp nhất được chọn có thể được sử dụng làm thông tin chuyển động của khối hiện tại.

Như là một ví dụ khác, thiết bị giải mã có thể xây dựng danh sách ứng viên (A)MVP, mà nó sẽ được mô tả dưới đây, khi chế độ (A)MVP được áp dụng cho khối hiện tại, và sử dụng vector chuyển động của ứng viên bộ dự đoán vector chuyển động (motion vector predictor, mvp) được chọn từ trong số các ứng viên mvp được chứa trong danh sách ứng viên (A)MVP làm mvp của khối hiện tại. Việc chọn có thể được thực hiện dựa trên thông tin chọn được mô tả trên đây (cờ mvp hoặc chỉ số mvp). Trong trường hợp này, MVD của khối hiện tại có thể được dẫn ra dựa trên thông tin về MVD, và vector chuyển động của khối hiện tại có thể được dẫn ra dựa trên MVD và mvp của khối hiện tại. Thêm vào đó, chỉ số hình ảnh tham chiếu của khối hiện tại có thể được dẫn ra dựa trên thông tin chỉ số hình ảnh tham chiếu. Hình ảnh được chỉ ra bởi chỉ số hình ảnh tham chiếu trong danh sách hình ảnh tham chiếu cho khối hiện tại có thể được dẫn ra làm hình ảnh tham chiếu được tham chiếu cho việc dự đoán liên khối hiện tại.

Trong khi đó, như được mô tả trên đây, thông tin chuyển động của khối hiện tại có thể được dẫn ra mà không xây dựng danh sách ứng viên. Trong trường hợp này, thông tin chuyển động của khối hiện tại có thể được dẫn ra theo thủ tục trong chế độ dự đoán mà nó sẽ được mô

tả dưới đây. Trong trường hợp này, việc xây dựng danh sách ứng viên như được mô tả trên đây có thể được lược bỏ.

Thiết bị giải mã có thể tạo ra các mẫu dự đoán cho khối hiện tại dựa trên thông tin chuyển động của khối hiện tại (S1230). Trong trường hợp này, hình ảnh tham chiếu có thể được dẫn ra dựa trên chỉ số hình ảnh tham chiếu của khối hiện tại, và các mẫu dự đoán của khối hiện tại có thể được dẫn ra nhờ sử dụng các mẫu của khối tham chiếu được chỉ ra bởi vector chuyển động của khối hiện tại trên hình ảnh tham chiếu. Trong trường hợp này, như được mô tả trên đây, trong một số trường hợp, thủ tục lọc mẫu dự đoán cho tất cả hoặc một số mẫu dự đoán của khối hiện tại có thể được thực hiện thêm.

Ví dụ, bộ dự đoán liên của thiết bị giải mã có thể bao gồm đơn vị xác định chế độ dự đoán, đơn vị dẫn ra thông tin chuyển động, và đơn vị dẫn ra mẫu dự đoán, đơn vị xác định chế độ dự đoán có thể xác định chế độ dự đoán cho khối hiện tại dựa trên thông tin chế độ dự đoán nhận được, đơn vị dẫn ra thông tin chuyển động có thể dẫn ra thông tin chuyển động (vector chuyển động và/hoặc chỉ số hình ảnh tham chiếu) của khối hiện tại dựa trên thông tin chuyển động nhận được, và đơn vị dẫn ra mẫu dự đoán có thể dẫn ra các mẫu dự đoán của khối hiện tại.

Thiết bị giải mã tạo ra các mẫu phần dư cho khối hiện tại dựa trên thông tin phần dư nhận được (S1240). Thiết bị giải mã có thể tạo ra các mẫu được xây dựng lại cho khối hiện tại dựa trên các mẫu dự đoán và các mẫu phần dư, và tạo ra hình ảnh được xây dựng lại dựa trên điều này (S1250). Như được mô tả trên đây, thủ tục lọc trong vòng có thể còn áp dụng được cho hình ảnh được xây dựng lại.

Như được mô tả trên đây, thủ tục dự đoán liên có thể bao gồm bước xác định chế độ dự đoán liên, bước dẫn ra thông tin chuyển động theo chế độ dự đoán được xác định, và bước thực hiện việc dự đoán dựa trên thông tin chuyển động được dẫn ra (việc tạo ra mẫu dự đoán). Thủ tục dự đoán liên có thể được thực hiện trong thiết bị mã hóa và thiết bị giải mã như được mô tả trên đây.

Việc lượng tử hóa/giải lượng tử hóa

Như được mô tả trên đây, bộ lượng tử hóa của thiết bị mã hóa có thể dẫn ra các hệ số biến đổi được lượng tử hóa bằng cách áp dụng việc lượng tử hóa cho các hệ số biến đổi, và bộ giải lượng tử hóa của thiết bị mã hóa hoặc bộ giải lượng tử hóa của thiết bị giải mã có thể dẫn ra các hệ số biến đổi bằng cách áp dụng việc giải lượng tử hóa cho các hệ số biến đổi được lượng tử hóa.

Khi mã hóa và giải mã ảnh động/ảnh tĩnh, tỷ lệ lượng tử hóa có thể được thay đổi và tỷ lệ nén có thể được điều chỉnh nhờ sử dụng tỷ lệ lượng tử hóa được thay đổi này. Từ góc nhìn về cách thi hành, khi xem xét đến độ phức tạp, thay vì sử dụng trực tiếp tỷ lệ lượng tử hóa, tham số lượng tử hóa (quantization parameter, QP) có thể được sử dụng. Ví dụ, các tham số lượng tử hóa có các giá trị nguyên từ 0 đến 63 có thể được sử dụng và mỗi giá trị tham số lượng tử hóa có thể tương ứng với một tỷ lệ lượng tử hóa thực tế. Thêm vào đó, tham số lượng tử hóa QP_Y cho thành phần độ chói (mẫu độ chói) và tham số lượng tử hóa QP_C cho thành phần sắc độ (mẫu sắc độ) có thể được thiết lập khác nhau.

Trong quy trình lượng tử hóa, hệ số biến đổi C có thể được nhận và chia cho tỷ lệ lượng tử hóa Q_{step} , qua đó thu phép biến đổi được lượng tử hóa. Trong trường hợp này, tỷ lệ lượng tử hóa có thể được nhân với thang chia tỷ lệ (scale) khi xem xét đến độ phức tạp tính toán để tạo thành số nguyên, và phép toán dịch chuyển có thể được thực hiện bằng giá trị tương ứng với giá trị thang chia tỷ lệ này. Thang chia tỷ lệ lượng tử hóa có thể được dẫn ra dựa trên tích số của tỷ lệ lượng tử hóa và giá trị thang chia tỷ lệ. Tức là, thang chia tỷ lệ lượng tử hóa có thể được dẫn ra theo QP. Bằng cách áp dụng thang chia tỷ lệ lượng tử hóa cho hệ số biến đổi C , hệ số biến đổi được lượng tử hóa C' có thể được dẫn ra.

Quy trình giải lượng tử hóa là quy trình ngược của quy trình lượng tử hóa. Bằng cách nhân hệ số biến đổi được lượng tử hóa C' với tỷ lệ lượng tử hóa Q_{step} , qua đó thu hệ số biến đổi được xây dựng lại C'' . Thêm vào đó, thang chia tỷ lệ mức có thể được dẫn ra theo tham số lượng tử hóa, và thang chia tỷ lệ mức này có thể được áp dụng cho hệ số biến đổi được lượng

tử hóa C' , qua đó dẫn ra hệ số biến đổi được xây dựng lại C'' . Hệ số biến đổi được xây dựng lại C'' có thể hơi khác so với hệ số biến đổi ban đầu C do sự mất mát của quy trình biến đổi và/hoặc lượng tử hóa. Theo đó, kể cả trong thiết bị mã hóa, việc giải lượng tử hóa có thể được thực hiện theo cách giống như trong thiết bị giải mã.

Trong khi đó, công nghệ lượng tử hóa định trọng số tần số thích ứng để điều chỉnh cường độ lượng tử hóa theo tần số có thể được áp dụng. Công nghệ lượng tử hóa định trọng số tần số thích ứng đề cập đến phương pháp để áp dụng theo cách khác nhau cường độ lượng tử hóa tùy theo tần số. Trong việc lượng tử hóa định trọng số tần số thích ứng, cường độ lượng tử hóa có thể được áp dụng theo cách khác nhau tùy theo tần số nhờ sử dụng ma trận định tỷ lệ lượng tử hóa định trước. Tức là, quy trình lượng tử hóa/giải lượng tử hóa được mô tả trên đây có thể được thực hiện dựa trên ma trận định tỷ lệ lượng tử hóa. Ví dụ, các ma trận định tỷ lệ lượng tử hóa khác nhau có thể được sử dụng tùy theo kích thước của khối hiện tại và/hoặc việc liệu chế độ dự đoán được áp dụng cho khối hiện tại là việc dự đoán liên hay việc dự đoán nội để tạo ra tín hiệu phần dư của khối hiện tại. Ma trận định tỷ lệ lượng tử hóa có thể được gọi là ma trận lượng tử hóa hoặc ma trận định tỷ lệ. Ma trận định tỷ lệ lượng tử hóa có thể được định trước. Thêm vào đó, đối với việc định tỷ lệ thích ứng tần số, thông tin thang chia tỷ lệ lượng tử hóa tần số cho ma trận định tỷ lệ lượng tử hóa có thể được xây dựng/mã hóa bởi thiết bị mã hóa và được báo hiệu đến thiết bị giải mã. Thông tin thang chia tỷ lệ lượng tử hóa tần số này có thể được gọi là thông tin định tỷ lệ lượng tử hóa. Thông tin thang chia tỷ lệ lượng tử hóa tần số này có thể bao gồm dữ liệu danh sách định tỷ lệ `scaling_list_data`. Ma trận định tỷ lệ lượng tử hóa (được cải biến) có thể được dẫn ra dựa trên dữ liệu danh sách định tỷ lệ này. Thêm vào đó, thông tin thang chia tỷ lệ lượng tử hóa tần số có thể bao gồm còn có mặt chỉ ra việc liệu dữ liệu danh sách định tỷ lệ có có mặt hay không. Theo cách khác, khi dữ liệu danh sách định tỷ lệ được báo hiệu tại mức cao hơn (ví dụ, SPS), thông tin chỉ ra liệu dữ liệu danh sách định tỷ lệ có được cải biến tại mức thấp hơn (ví dụ, PPS hoặc tiêu đề nhóm ô xếp, v.v.) hay không có thể được đưa thêm vào.

Việc biến đổi/biến đổi ngược

Như được mô tả trên đây, thiết bị mã hóa có thể dẫn ra khối phần dư (các mẫu phần dư) dựa trên khối (các khối dự đoán) được dự đoán thông qua việc dự đoán nội/liên/IBC, và dẫn ra các hệ số biến đổi được lượng tử hóa bằng cách áp dụng việc biến đổi và lượng tử hóa cho các mẫu phần dư được dẫn ra. Thông tin về các hệ số biến đổi được lượng tử hóa (thông tin phần dư) có thể được đưa vào và được mã hóa trong cú pháp lập mã phần dư và được xuất ra dưới dạng luồng bit. Thiết bị giải mã có thể thu nhận và giải mã thông tin về các hệ số biến đổi được lượng tử hóa (thông tin phần dư) từ luồng bit để dẫn ra các hệ số biến đổi được lượng tử hóa. Thiết bị giải mã có thể dẫn ra các mẫu phần dư qua việc giải lượng tử hóa/biến đổi ngược dựa trên các hệ số biến đổi được lượng tử hóa này. Như được mô tả trên đây, ít nhất một thành phần trong số việc lượng tử hóa/giải lượng tử hóa và/hoặc việc biến đổi/biến đổi ngược có thể được bỏ qua. Khi việc biến đổi/biến đổi ngược được bỏ qua, hệ số biến đổi có thể được gọi là hệ số hoặc hệ số phần dư hoặc có thể vẫn được gọi là hệ số biến đổi cho nhất quán về cách nói. Việc liệu việc biến đổi/biến đổi ngược có được bỏ qua hay không có thể được báo hiệu dựa trên cờ bỏ qua biến đổi (ví dụ, `transform_skip_flag`).

Việc biến đổi/biến đổi ngược có thể được thực hiện dựa trên nhân (các nhân) biến đổi. Ví dụ, sơ đồ lựa chọn nhiều biến đổi (multiple transform selection, MTS) để thực hiện việc biến đổi/biến đổi ngược là có thể áp dụng được. Trong trường hợp này, một số tập hợp nhân biến đổi trong số nhiều tập hợp nhân biến đổi có thể được chọn và được áp dụng cho khối hiện tại. Nhân biến đổi có thể được gọi theo nhiều thuật ngữ, chẳng hạn như ma trận biến đổi hoặc loại biến đổi. Ví dụ, tập hợp nhân biến đổi có thể chỉ ra sự kết hợp của nhân biến đổi hướng dọc (nhân biến đổi dọc, vertical transform kernel) và nhân biến đổi hướng ngang (nhân biến đổi ngang, horizontal transform kernel).

Việc biến đổi/biến đổi ngược có thể được thực hiện theo các đơn vị là CU hoặc TU. Tức là, việc biến đổi/biến đổi ngược là có thể áp dụng được cho các mẫu phần dư trong CU hoặc các mẫu phần dư trong TU. Kích thước CU có thể là bằng với kích thước TU hoặc nhiều TU

có thể có mặt trong một vùng CU. Trong khi đó, kích thước CU nhìn chung có thể biểu thị kích thước CB (mẫu) thành phần độ chói. Kích thước TU nhìn chung có thể biểu thị kích thước TB (mẫu) thành phần độ chói. Kích thước CB hoặc TB thành phần (mẫu) sắc độ có thể được dẫn ra dựa trên kích thước CB hoặc TB thành phần (mẫu) độ chói theo tỷ lệ thành phần theo định dạng màu (định dạng sắc độ) (ví dụ, 4:4:4, 4:2:2, 4:2:0, v.v.). Kích thước TU có thể được dẫn ra dựa trên maxTbSize. Ví dụ, khi kích thước CU lớn hơn maxTbSize, nhiều TU (TB) có maxTbSize có thể được dẫn ra từ CU và việc biến đổi/biến đổi ngược có thể được thực hiện theo các đơn vị là TU (TB). maxTbSize có thể được xem xét để xác định xem liệu có áp dụng các loại dự đoán nội khác nhau chẳng hạn như ISP hay không. Thông tin về maxTbSize có thể được xác định trước hoặc có thể được tạo ra và được mã hóa trong thiết bị mã hóa và được báo hiệu đến thiết bị giải mã.

Việc lập mã entropy

Toàn bộ hoặc một phần thông tin video/ảnh có thể được mã hóa entropy bởi bộ mã hóa entropy 190 như được mô tả trên đây với tham chiếu đến Fig.2, và toàn bộ hoặc một phần thông tin video/ảnh được mô tả với tham chiếu đến Fig.3 có thể được giải mã entropy bởi bộ giải mã entropy 310. Trong trường hợp này, thông tin video/ảnh có thể được mã hóa/giải mã theo các đơn vị là phân tử cú pháp. Trong sáng chế này, thông tin mã hóa/giải mã có thể bao gồm việc mã hóa/giải mã bởi phương pháp được mô tả trong đoạn này.

Fig.13 là sơ đồ khối của CABAC để mã hóa một phân tử cú pháp. Trong quy trình mã hóa của CABAC, trước hết, khi tín hiệu đầu vào là phân tử cú pháp khác với giá trị nhị phân, tín hiệu đầu vào này có thể được biến đổi thành giá trị nhị phân qua việc nhị phân hóa. Khi tín hiệu đầu vào đã có giá trị nhị phân, việc nhị phân hóa có thể được đi vòng qua. Ở đây, số nhị phân 0 hoặc 1 tạo cấu hình giá trị nhị phân có thể được gọi là ngăn (bin). Ví dụ, khi chuỗi nhị phân (chuỗi ngăn) sau khi nhị phân hóa là 110, mỗi thành phần trong số 1, 1 và 0 có thể được gọi là một ngăn. Ngăn (các ngăn) cho một phân tử cú pháp có thể thể hiện giá trị của phân tử cú pháp tương ứng.

Các ngăn được nhị phân hóa có thể là đầu vào cho cỗ máy lập mã chính quy hoặc cỗ máy lập mã đi vòng (bypass). Cỗ máy lập mã chính quy có thể cấp phát mô hình ngữ cảnh phản ánh giá trị xác suất cho ngăn tương ứng và mã hóa bit tương ứng dựa trên mô hình ngữ cảnh được cấp phát. Cỗ máy lập mã chính quy có thể lập mã mỗi ngăn và sau đó cập nhật mô hình xác suất cho ngăn tương ứng. Các ngăn được lập mã theo cách này có thể được gọi là các ngăn được lập mã ngữ cảnh. Cỗ máy lập mã đi vòng có thể đi vòng qua thủ tục để ước lượng xác suất đối với ngăn đầu vào và thủ tục để cập nhật chế độ xác suất được áp dụng cho ngăn tương ứng sau khi lập mã. Cỗ máy lập mã đi vòng có thể lập mã ngăn đầu vào bằng cách áp dụng phân phối xác suất đều (ví dụ, 50:50) thay vì cấp phát ngữ cảnh, qua đó cải thiện tốc độ lập mã. Các ngăn được lập mã theo cách này có thể được gọi là ngăn đi vòng. Mô hình ngữ cảnh có thể được cấp phát và cập nhật cho mỗi ngăn được lập mã ngữ cảnh (được lập mã chính quy), và mô hình ngữ cảnh có thể được chỉ ra dựa trên `ctxidx` hoặc `ctxInc`. `ctxidx` có thể được dẫn ra dựa trên `ctxInc`. Cụ thể, ví dụ, chỉ số ngữ cảnh `ctxidx` chỉ ra mô hình ngữ cảnh cho mỗi ngăn trong số các ngăn được lập mã chính quy có thể được dẫn ra là tổng của lượng gia chỉ số ngữ cảnh (context index increment, `ctxInc`) và phần bù chỉ số ngữ cảnh (context index offset, `ctxIdxOffset`). Ở đây, `ctxInc` có thể được dẫn ra khác nhau cho mỗi ngăn. `ctxIdxOffset` có thể được thể hiện bởi giá trị thấp nhất của `ctxIdx`. Giá trị thấp nhất của `ctxIdx` có thể được gọi là giá trị ban đầu `initValue` của `ctxIdx`. `ctxIdxOffset` nhìn chung là giá trị được sử dụng để được phân biệt với các mô hình ngữ cảnh cho các phần tử cú pháp khác, và mô hình ngữ cảnh cho một phần tử cú pháp có thể được phân biệt/được dẫn ra dựa trên `ctxinc`.

Trong thủ tục mã hóa entropy, việc liệu việc mã hóa được thực hiện thông qua cỗ máy lập mã chính quy hay cỗ máy lập mã đi vòng có thể được xác định và đường lập mã có thể được chuyển đổi. Việc giải mã entropy có thể được thực hiện theo thứ tự ngược của cùng quy trình như việc mã hóa entropy.

Việc lập mã entropy được mô tả trên đây có thể được thực hiện, ví dụ, như được thể hiện trên Fig.14 và Fig.15. Tham chiếu đến Fig.14 và Fig.15, thiết bị mã hóa (bộ mã hóa entropy)

có thể thực hiện thủ tục lập mã entropy đối với thông tin ảnh/video. Thông tin ảnh/video có thể bao gồm thông tin liên quan đến việc phân vùng, thông tin liên quan đến việc dự đoán (ví dụ, thông tin phân biệt việc dự đoán liên/nội, thông tin chế độ dự đoán nội, thông tin chế độ dự đoán liên, v.v.), thông tin phần dư, thông tin liên quan đến việc lọc trong vòng, v.v., hoặc có thể bao gồm các phần tử cú pháp khác nhau liên quan đến chúng. Việc lập mã entropy có thể được thực hiện theo các đơn vị là phần tử cú pháp. Các bước S1410 đến S1420 trên Fig.14 có thể được thực hiện bởi bộ mã hóa entropy 190 của thiết bị mã hóa trên Fig.2.

Thiết bị mã hóa có thể thực hiện việc nhị phân hóa đối với phần tử cú pháp mục tiêu (S1410). Ở đây, việc nhị phân hóa có thể được dựa trên các phương pháp nhị phân hóa khác nhau, chẳng hạn như quy trình nhị phân hóa Rice được cắt cụt (Truncated Rice), quy trình nhị phân hóa chiều dài cố định (Fixed-length), v.v, và phương pháp nhị phân hóa cho phần tử cú pháp mục tiêu có thể được định trước. Thủ tục nhị phân hóa có thể được thực hiện bởi đơn vị nhị phân hóa 191 trong bộ mã hóa entropy 190.

Thiết bị mã hóa có thể mã hóa entropy đối với phần tử cú pháp mục tiêu (S1420). Thiết bị mã hóa có thể thực hiện việc mã hóa dựa trên việc lập mã chính quy (dựa trên ngữ cảnh) hoặc dựa trên việc lập mã đi vòng đối với chuỗi ngắn của phần tử cú pháp mục tiêu dựa trên kỹ thuật lập mã entropy như CABAC (context-adaptive arithmetic coding, việc lập mã số học tương thích ngữ cảnh) hoặc CAVLC (context-adaptive variable length coding, việc lập mã chiều dài biến đổi thích ứng ngữ cảnh), và đầu ra của chúng có thể được đưa vào trong luồng bit. Thủ tục mã hóa entropy có thể được thực hiện bởi bộ xử lý mã hóa entropy 192 trong bộ mã hóa entropy 190. Như được mô tả trên đây, luồng bit có thể được truyền đến thiết bị giải mã qua phương tiện lưu trữ (kỹ thuật số) hoặc mạng.

Tham chiếu đến Fig.16 và Fig.17, thiết bị giải mã (bộ giải mã entropy) có thể giải mã thông tin ảnh/video đã được mã hóa. Thông tin ảnh/video có thể bao gồm thông tin liên quan đến việc phân vùng, thông tin liên quan đến việc dự đoán (ví dụ, thông tin phân biệt việc dự đoán liên/nội, thông tin chế độ dự đoán nội, thông tin chế độ dự đoán liên, v.v.), thông tin phần

dur, thông tin liên quan đến việc lọc trong vòng, v.v., hoặc có thể bao gồm các phần tử cú pháp khác nhau liên quan đến chúng. Việc lập mã entropy có thể được thực hiện theo các đơn vị là phần tử cú pháp. Các bước S1610 đến S1620 trên Fig.16 có thể được thực hiện bởi bộ giải mã entropy 210 của thiết bị giải mã trên Fig.3.

Thiết bị giải mã có thể thực hiện việc nhị phân hóa đối với phần tử cú pháp mục tiêu (S1610). Ở đây, việc nhị phân hóa có thể được dựa trên các phương pháp nhị phân hóa khác nhau, chẳng hạn như quy trình nhị phân hóa Rice được cắt cụt (Truncated Rice), quy trình nhị phân hóa chiều dài cố định (Fixed-length), v.v, và phương pháp nhị phân hóa cho phần tử cú pháp mục tiêu có thể được định trước. Thiết bị giải mã có thể dẫn ra các chuỗi ngăn khả dụng (các ứng viên chuỗi ngăn) để được các giá trị khả dụng của phần tử cú pháp mục tiêu qua thủ tục nhị phân hóa. Thủ tục nhị phân hóa có thể được thực hiện bởi đơn vị nhị phân hóa 211 trong bộ giải mã entropy 210.

Thiết bị giải mã có thể thực hiện việc giải mã entropy đối với phần tử cú pháp mục tiêu (S1620). Thiết bị giải mã có thể so sánh chuỗi ngăn được dẫn ra với các chuỗi ngăn khả dụng dành cho phần tử cú pháp tương ứng, trong khi tuần tự giải mã và phân tích cú pháp các ngăn dành cho phần tử cú pháp mục tiêu từ bit (các bit) đầu vào trong luồng bit. Nếu chuỗi ngăn được dẫn ra là bằng một trong các chuỗi ngăn khả dụng thì giá trị tương ứng với chuỗi ngăn tương ứng có thể được dẫn ra làm giá trị của phần tử cú pháp tương ứng. Nếu không thì thủ tục được mô tả trên đây có thể được thực hiện lần nữa sau khi bit tiếp theo trong luồng bit được phân tích cú pháp thêm. Qua quy trình này, thông tin tương ứng có thể được báo hiệu nhờ sử dụng bit chiều dài biến đổi mà không sử dụng bit khởi đầu hoặc kết thúc dành cho thông tin cụ thể (phần tử cú pháp cụ thể) trong luồng bit. Thông qua điều này, tương đối ít bit hơn có thể được cấp phát cho giá trị thấp và hiệu suất lập mã tổng thể có thể được cải thiện.

Thiết bị giải mã có thể thực hiện việc giải mã dựa trên ngữ cảnh hoặc dựa trên việc đi vòng đối với các ngăn trong chuỗi ngăn từ luồng bit dựa trên kỹ thuật lập mã entropy như CABAC hoặc CAVLC. Thủ tục giải mã entropy có thể được thực hiện bởi bộ xử lý giải mã

entropy 212 trong bộ giải mã entropy 210. Luồng bit có thể bao gồm nhiều thông tin đa dạng cho việc giải mã ảnh/video như được mô tả trên đây. Như được mô tả trên đây, luồng bit có thể được truyền đến thiết bị giải mã qua phương tiện lưu trữ (kỹ thuật số) hoặc mạng.

Trong sáng chế này, bảng (bảng cú pháp) bao gồm các phần tử cú pháp có thể được sử dụng để chỉ ra việc báo hiệu thông tin từ thiết bị mã hóa đến thiết bị giải mã. Thứ tự của các phần tử cú pháp của bảng bao gồm các phần tử cú pháp được sử dụng trong sáng chế có thể chỉ ra thứ tự phân tích cú pháp các phần tử cú pháp từ luồng bit. Thiết bị mã hóa có thể xây dựng và mã hóa phần tử cú pháp sao cho thiết bị giải mã phân tích cú pháp phần tử cú pháp theo thứ tự phân tích cú pháp, và thiết bị giải mã có thể phân tích cú pháp và giải mã các phần tử cú pháp của bảng cú pháp tương ứng từ luồng bit theo thứ tự phân tích cú pháp và thu các giá trị của các phần tử cú pháp.

Thủ tục lập mã ảnh/video nói chung

Trong việc lập mã ảnh/video, hình ảnh tạo cấu trúc ảnh/video có thể được mã hóa/giải mã theo thứ tự giải mã. Thứ tự hình ảnh tương ứng với thứ tự xuất ra của hình ảnh được giải mã có thể được thiết lập khác với thứ tự giải mã, và, dựa trên điều này, không chỉ việc dự đoán thuận chiều mà cả việc dự đoán ngược chiều có thể được thực hiện trong quá trình dự đoán liên.

Fig.18 thể hiện ví dụ về thủ tục giải mã hình ảnh dưới dạng sơ đồ, mà phương án (các phương án) của sáng chế có thể áp dụng được với nó. Trên Fig.18, S1810 có thể được thực hiện trong bộ giải mã entropy 210 của thiết bị giải mã, S1820 có thể được thực hiện trong bộ dự đoán bao gồm bộ dự đoán nội 265 và bộ dự đoán liên 260, S1830 có thể được thực hiện trong bộ xử lý phần dư bao gồm bộ giải lượng tử hóa 220 và bộ biến đổi ngược 230, S1840 có thể được thực hiện trong bộ cộng 235, và S1850 có thể được thực hiện trong bộ lọc 240. S1810 có thể bao gồm thủ tục giải mã thông tin được mô tả trong sáng chế, S1820 có thể bao gồm thủ tục dự đoán liên/nội được mô tả trong sáng chế, S1830 có thể bao gồm thủ tục xử lý phần dư được mô tả trong sáng chế, S1840 có thể bao gồm thủ tục xây dựng lại khối/hình ảnh được mô

tả trong sáng chế, và S1850 có thể bao gồm thủ tục lọc trong vòng được mô tả trong sáng chế.

Tham chiếu đến Fig.18, thủ tục giải mã hình ảnh có thể sơ lược bao gồm thủ tục (S1810) để thu thông tin ảnh/video (qua việc giải mã) từ luồng bit, thủ tục xây dựng lại hình ảnh (S1820 đến S1840) và thủ tục lọc trong vòng (S1850) đối với hình ảnh được xây dựng lại. Thủ tục xây dựng lại hình ảnh có thể được thực hiện dựa trên các mẫu dự đoán và các mẫu phần dư được thu qua việc dự đoán liên/nội (S1820) và việc xử lý phần dư (S1830) (việc giải lượng tử hóa và biến đổi ngược hệ số biến đổi được lượng tử hóa) được mô tả trong sáng chế này. Hình ảnh được xây dựng lại được cải biến có thể được tạo ra qua thủ tục lọc trong vòng đối với hình ảnh được xây dựng lại được tạo ra qua thủ tục xây dựng lại hình ảnh, hình ảnh được xây dựng lại được cải biến này có thể được xuất ra dưới dạng hình ảnh được giải mã, được lưu trữ trong bộ đệm hình ảnh được giải mã hoặc bộ nhớ 250 của thiết bị giải mã và được sử dụng làm hình ảnh tham chiếu trong thủ tục dự đoán liên khi sau này giải mã hình ảnh này. Trong một số trường hợp, thủ tục lọc trong vòng có thể được lược bỏ. Trong trường hợp này, hình ảnh được xây dựng lại có thể được xuất ra dưới dạng hình ảnh được giải mã, được lưu trữ trong bộ đệm hình ảnh được giải mã hoặc bộ nhớ 250 của thiết bị giải mã, và được sử dụng làm hình ảnh tham chiếu trong thủ tục dự đoán liên khi sau này giải mã hình ảnh này. Thủ tục lọc trong vòng (S1850) có thể bao gồm thủ tục lọc khử khối, thủ tục phân bù thích ứng mẫu (sample adaptive offset, SAO), thủ tục bộ lọc vòng thích ứng (adaptive loop filter, ALF) và/hoặc thủ tục bộ lọc song phương (bi-lateral), như được mô tả trên đây, mà một vài hoặc tất cả trong số chúng có thể được lược bỏ. Thêm vào đó, một hoặc một số thành phần trong số thủ tục lọc khử khối, thủ tục phân bù thích ứng mẫu (SAO), thủ tục bộ lọc vòng thích ứng (ALF) và/hoặc thủ tục bộ lọc song phương có thể được áp dụng tuần tự hoặc tất cả trong số chúng có thể được áp dụng tuần tự. Ví dụ, sau khi thủ tục lọc khử khối được áp dụng cho hình ảnh được xây dựng lại, thủ tục SAO có thể được thực hiện. Theo cách khác, ví dụ, sau khi thủ tục lọc khử khối được áp dụng cho hình ảnh được xây dựng lại, thủ tục ALF có thể được thực hiện. Việc này có thể được thực hiện tương tự kể cả trong thiết bị mã hóa.

Fig.19 thể hiện ví dụ về thủ tục mã hóa hình ảnh dưới dạng sơ đồ, mà phương án (các phương án) của sáng chế có thể áp dụng được với nó. Trên Fig.19, S1910 có thể được thực hiện trong bộ dự đoán bao gồm bộ dự đoán nội 185 hoặc bộ dự đoán liên 180 của thiết bị mã hóa được mô tả trên đây với tham chiếu đến Fig.2, S1920 có thể được thực hiện trong bộ xử lý phần dư bao gồm bộ biến đổi 120 và/hoặc bộ lượng tử hóa 130, và S1930 có thể được thực hiện trong bộ mã hóa entropy 190. S1910 có thể bao gồm thủ tục dự đoán liên/nội được mô tả trong sáng chế, S1920 có thể bao gồm thủ tục xử lý phần dư được mô tả trong sáng chế, và S1930 có thể bao gồm thủ tục mã hóa thông tin được mô tả trong sáng chế.

Tham chiếu đến Fig.19, thủ tục mã hóa hình ảnh có thể sơ lược bao gồm không chỉ thủ tục để mã hóa và xuất ra thông tin cho việc xây dựng lại hình ảnh (ví dụ, thông tin dự đoán, thông tin phần dư, thông tin phân vùng, v.v.) dưới dạng luồng bit mà cả thủ tục để tạo ra hình ảnh được xây dựng lại cho hình ảnh hiện tại và thủ tục (tùy chọn) để áp dụng việc lọc trong vòng cho hình ảnh được xây dựng lại, như được mô tả đối với Fig.2. Thiết bị mã hóa có thể dẫn ra các mẫu phần dư (được cải biến) từ hệ số biến đổi được lượng tử hóa qua bộ giải lượng tử hóa 140 và bộ biến đổi ngược 150, và tạo ra hình ảnh được xây dựng lại dựa trên các mẫu dự đoán mà chúng là đầu ra của S1910 và các mẫu phần dư (được cải biến) này. Hình ảnh được xây dựng lại được tạo ra theo cách này có thể là bằng với hình ảnh được xây dựng lại được tạo ra trong thiết bị giải mã. Hình ảnh được xây dựng lại được cải biến có thể được tạo ra qua thủ tục lọc trong vòng đối với hình ảnh được xây dựng lại, có thể được lưu trữ trong bộ đệm hình ảnh được giải mã hoặc bộ nhớ 170, và có thể được sử dụng làm hình ảnh tham chiếu trong thủ tục dự đoán liên khi sau này mã hóa hình ảnh này, tương tự với thiết bị giải mã. Như được mô tả trên đây, trong một số trường hợp, một phần hoặc toàn bộ thủ tục lọc trong vòng có thể được lược bỏ. Khi thủ tục lọc trong vòng được thực hiện, thông tin (tham số) liên quan đến việc lọc (trong vòng) có thể được mã hóa trong bộ mã hóa entropy 190 và được xuất ra dưới dạng luồng bit, và thiết bị giải mã có thể thực hiện thủ tục lọc trong vòng nhờ sử dụng cùng phương pháp với thiết bị mã hóa dựa trên thông tin liên quan đến việc lọc này.

Qua thủ tục lọc trong vòng như vậy, nhiễu xảy ra trong việc lập mã ảnh/video, chẳng hạn như thành phần lạ tạo khối và thành phần lạ tiếng vọng, có thể được giảm xuống và chất lượng thị giác chủ quan/khách quan có thể được cải thiện. Thêm vào đó, qua việc thực hiện thủ tục lọc trong vòng trong cả thiết bị mã hóa và thiết bị giải mã, thiết bị mã hóa và thiết bị giải mã có thể dẫn ra kết quả dự đoán giống nhau, độ tin cậy của việc lập mã hình ảnh có thể được tăng lên và lượng dữ liệu cần được truyền cho việc lập mã hình ảnh có thể được giảm xuống.

Như được mô tả trên đây, thủ tục xây dựng lại hình ảnh có thể được thực hiện không chỉ trong thiết bị giải mã mà cả trong thiết bị mã hóa. Khối được xây dựng lại có thể được tạo ra dựa trên việc dự đoán nội/dự đoán liên theo các đơn vị là các khối, và hình ảnh được xây dựng lại bao gồm các khối được xây dựng lại có thể được tạo ra. Khi hình ảnh/lát/nhóm ô xếp hiện tại là hình ảnh/lát/nhóm ô xếp I, các khối được chứa trong hình ảnh/lát/nhóm ô xếp hiện tại này có thể được xây dựng lại chỉ dựa trên việc dự đoán nội. Mặt khác, khi hình ảnh/lát/nhóm ô xếp hiện tại là hình ảnh/lát/nhóm ô xếp P hoặc B, các khối được chứa trong hình ảnh/lát/nhóm ô xếp hiện tại này có thể được xây dựng lại dựa trên việc dự đoán nội hoặc việc dự đoán liên. Trong trường hợp này, việc dự đoán liên có thể được áp dụng cho một số khối trong hình ảnh/lát/nhóm ô xếp hiện tại này và việc dự đoán nội có thể được áp dụng cho các khối còn lại. Thành phần màu của hình ảnh có thể bao gồm thành phần độ chói và thành phần sắc độ và các phương pháp và các phương án của sáng chế là có thể áp dụng được với thành phần độ chói và thành phần sắc độ trừ khi bị giới hạn một cách tường minh trong sáng chế này.

Ví dụ về lớp và cấu trúc lập mã

Video/ảnh được lập mã theo sáng chế có thể được xử lý, ví dụ, theo lớp và cấu trúc lập mã mà chúng sẽ được mô tả dưới đây.

Fig.20 là hình thể hiện cấu trúc lớp cho ảnh được lập mã. Ảnh được lập mã có thể được phân loại thành lớp lập mã video (video coding layer, VCL) cho quy trình giải mã ảnh và để xử lý bản thân nó, hệ thống bên dưới để truyền và lưu trữ thông tin được mã hóa, và lớp trừu tượng mạng (NAL) có mặt giữa VCL và hệ thống bên dưới này và chịu trách nhiệm về chức

năng thích ứng mạng.

Trong VCL, dữ liệu VCL bao gồm dữ liệu ảnh (dữ liệu lát) được nén có thể được tạo ra hoặc tin nhắn thông tin tăng cường bổ sung (supplemental enhancement information, SEI) cần thiết thêm cho quy trình giải mã của ảnh hoặc tập tham số bao gồm thông tin chẳng hạn như tập tham số hình ảnh (PPS), tập tham số trình tự (SPS) hoặc tập tham số video (VPS) có thể được tạo ra.

Trong NAL, thông tin tiêu đề (tiêu đề đơn vị NAL) có thể được thêm vào phần tải hữu ích trình tự byte thô (raw byte sequence payload, RBSP) được tạo ra trong VCL để tạo ra đơn vị NAL. Trong trường hợp này, RBSP chỉ dữ liệu lát, tập tham số, tin nhắn SEI được tạo ra trong VCL. Tiêu đề đơn vị NAL có thể bao gồm thông tin loại đơn vị NAL được chỉ rõ tùy theo dữ liệu RBSP được chứa trong đơn vị NAL tương ứng.

Như được thể hiện trên hình, đơn vị NAL có thể được phân loại thành đơn vị NAL VCL (VCL NAL unit) và đơn vị NAL phi VCL (non-VCL NAL unit) tùy theo RBSP được tạo ra trong VCL này. Đơn vị NAL VCL có thể nghĩa là đơn vị NAL chứa thông tin về ảnh (dữ liệu lát), và đơn vị NAL phi VCL có thể nghĩa là đơn vị NAL chứa thông tin (tập tham số hoặc tin nhắn SEI) cần thiết để giải mã ảnh.

Đơn vị NAL VCL và đơn vị NAL phi VCL có thể được gắn với thông tin tiêu đề và được truyền qua mạng theo tiêu chuẩn dữ liệu của hệ thống bên dưới. Ví dụ, đơn vị NAL có thể được cải biến thành định dạng dữ liệu của tiêu chuẩn được xác định trước, chẳng hạn như định dạng tệp tin H.266/VVC, RTP (Real-time Transport Protocol, giao thức truyền thời gian thực), hoặc TS (Transport Stream, luồng truyền), và được truyền qua các mạng khác nhau.

Như được mô tả trên đây, trong đơn vị NAL, loại đơn vị NAL có thể được chỉ rõ theo cấu trúc dữ liệu RBSP được chứa trong đơn vị NAL tương ứng, và thông tin về loại đơn vị NAL có thể được lưu trữ trong tiêu đề đơn vị NAL và được báo hiệu.

Ví dụ, việc này phần lớn có thể được phân loại thành loại đơn vị NAL VCL và loại đơn vị NAL phi VCL phụ thuộc vào việc liệu đơn vị NAL này có chứa thông tin về ảnh (dữ liệu

lát) hay không. Loại đơn vị NAL VCL có thể được phân loại theo đặc tính và loại của hình ảnh được chứa trong đơn vị NAL VCL này, và loại đơn vị NAL phi VCL có thể được phân loại theo loại của tập tham số.

Ví dụ về loại đơn vị NAL được chỉ rõ theo loại của tập tham số/thông tin được chứa trong loại đơn vị NAL phi VCL sẽ được liệt kê dưới đây.

- Đơn vị NAL DCI (Decoding capability information, thông tin năng lực giải mã): loại dành cho đơn vị NAL chứa DCI
- Đơn vị NAL VPS (tập tham số video): loại dành cho đơn vị NAL chứa VPS
- Đơn vị NAL SPS (tập tham số trình tự): loại dành cho đơn vị NAL chứa SPS
- Đơn vị NAL PPS (tập tham số hình ảnh): loại dành cho đơn vị NAL chứa PPS
- Đơn vị NAL APS (tập tham số thích ứng): loại dành cho đơn vị NAL chứa APS
- Đơn vị NAL PH (Picture header, tiêu đề hình ảnh): loại dành cho đơn vị NAL chứa PH

Các loại đơn vị NAL được mô tả trên đây có thể có thông tin cú pháp dành cho loại đơn vị NAL, và thông tin cú pháp này có thể được lưu trữ trong tiêu đề đơn vị NAL và được báo hiệu. Ví dụ, thông tin cú pháp này có thể là `nal_unit_type`, và các loại đơn vị NAL có thể được chỉ rõ dưới dạng các giá trị `nal_unit_type`.

Trong khi đó, như được mô tả trên đây, một hình ảnh có thể bao gồm nhiều lát, và một lát có thể bao gồm tiêu đề lát và dữ liệu lát. Trong trường hợp này, một tiêu đề hình ảnh có thể còn được thêm vào nhiều lát (tiêu đề lát và tập dữ liệu lát) trong một hình ảnh. Tiêu đề hình ảnh (cú pháp tiêu đề hình ảnh) có thể bao gồm thông tin/các tham số áp dụng được chung cho hình ảnh này.

Tiêu đề lát (cú pháp tiêu đề lát) có thể bao gồm thông tin/các tham số áp dụng được chung cho lát này. APS (cú pháp APS) hoặc PPS (cú pháp PPS) có thể bao gồm thông tin/các tham số áp dụng được chung cho một hoặc nhiều lát hoặc hình ảnh. SPS (cú pháp SPS) có thể bao gồm thông tin/các tham số áp dụng được chung cho một hoặc nhiều trình tự. VPS (cú pháp VPS) có thể bao gồm thông tin/các tham số áp dụng được chung cho nhiều lớp. DCI (cú pháp

DCI) có thể bao gồm thông tin/các tham số có thể áp dụng được chung cho video tổng thể. DCI có thể bao gồm thông tin/các tham số liên quan đến năng lực giải mã. Trong sáng chế này, cú pháp mức cao (high level syntax, HLS) có thể bao gồm ít nhất một cú pháp trong số cú pháp APS, cú pháp PPS, cú pháp SPS, cú pháp VPS, cú pháp DCI, cú pháp tiêu đề hình ảnh hoặc cú pháp tiêu đề lát. Trong khi đó, trong sáng chế này, cú pháp mức thấp (low level syntax, LLS), ví dụ, có thể bao gồm cú pháp dữ liệu lát, cú pháp CTU, cú pháp đơn vị lập mã, cú pháp đơn vị biến đổi, v.v.

Trong sáng chế này, thông tin ảnh/video được mã hóa trong thiết bị mã hóa và được báo hiệu đến thiết bị giải mã dưới dạng luồng bit có thể bao gồm không chỉ thông tin liên quan đến việc phân vùng trong hình ảnh, thông tin dự đoán nội/liên, thông tin phần dư, thông tin việc lọc trong vòng, mà cả thông tin về tiêu đề lát, thông tin về tiêu đề hình ảnh, thông tin về APS, thông tin về PPS, thông tin về SPS, thông tin về VPS và/hoặc thông tin về DCI. Thêm vào đó, thông tin ảnh/video có thể còn bao gồm thông tin ràng buộc chung và/hoặc thông tin về tiêu đề đơn vị NAL.

Việc phân vùng hình ảnh sử dụng hình ảnh con, lát và ô xếp

Một hình ảnh có thể được phân vùng thành ít nhất một hàng ô xếp và ít nhất một cột ô xếp. Một ô xếp có thể gồm trình tự của các CTU và có thể bao trùm vùng hình chữ nhật của một hình ảnh.

Lát có thể gồm một số lượng nguyên các ô xếp hoàn chỉnh hoặc một số lượng nguyên các hàng CTU hoàn chỉnh liên tiếp trong một hình ảnh.

Đối với lát, hai chế độ có thể được hỗ trợ: một chế độ có thể được gọi là chế độ lát quét mảnh và chế độ còn lại có thể được gọi là chế độ lát hình chữ nhật. Trong chế độ lát quét mảnh, một lát có thể bao gồm trình tự ô xếp hoàn chỉnh có mặt trong một hình ảnh theo thứ tự quét mảnh ô xếp. Trong chế độ lát hình chữ nhật, một lát có thể bao gồm nhiều ô xếp hoàn chỉnh được tập hợp để tạo ra vùng hình chữ nhật của hình ảnh hoặc nhiều hàng CTU hoàn chỉnh liên tiếp của một ô xếp được tập hợp để tạo ra vùng hình chữ nhật của hình ảnh. Các ô xếp trong lát

hình chữ nhật có thể được quét theo thứ tự quét mảnh ô xếp trong vùng hình chữ nhật tương ứng với lát. Hình ảnh con có thể bao gồm ít nhất một lát được tập hợp để bao trùm vùng hình chữ nhật của hình ảnh.

Để mô tả mối quan hệ phân vùng của hình ảnh chi tiết hơn, phần mô tả sẽ được đưa ra với tham chiếu đến Fig.21 đến Fig.24. Fig.21 đến Fig.24 thể hiện các phương án về việc phân vùng hình ảnh nhờ sử dụng ô xếp, lát và hình ảnh con. Fig.21 thể hiện ví dụ về hình ảnh được phân vùng thành 12 ô xếp và ba lát quét mảnh. Fig.22 thể hiện ví dụ về hình ảnh được phân vùng thành 24 ô xếp (sáu cột ô xếp và bốn hàng ô xếp) và 9 lát hình chữ nhật. Fig.23 thể hiện ví dụ về hình ảnh được phân vùng thành bốn ô xếp (hai cột ô xếp và hai hàng ô xếp) và bốn lát hình chữ nhật.

Fig.24 thể hiện ví dụ về việc phân vùng hình ảnh thành các hình ảnh con. Trên Fig.24, hình ảnh có thể được phân vùng thành 12 ô xếp bên trái bao trùm một lát gồm các CTU 4x4 và sáu ô xếp bên phải bao trùm hai lát được tập hợp theo chiều dọc gồm các CTU 2x2, sao cho một hình ảnh được phân vùng thành 24 lát và 24 hình ảnh con có các diện tích khác nhau. Trong ví dụ trên Fig.24, lát riêng rẽ tương ứng với hình ảnh con riêng rẽ.

Việc báo hiệu HLS (high level syntax, cú pháp mức cao) và ngữ nghĩa

Như được mô tả trên đây, HLS có thể được mã hóa và/hoặc được báo hiệu cho việc mã hóa video và/hoặc ảnh. Như được mô tả trên đây, trong sáng chế này, thông tin video/ảnh có thể được đưa vào trong HLS. Thêm vào đó, phương pháp mã hóa ảnh/video có thể được thực hiện dựa trên thông tin ảnh/video này.

Tiêu đề hình ảnh và tiêu đề lát

Hình ảnh được mã hóa có thể gồm ít nhất một lát. Các tham số mô tả hình ảnh được mã hóa có thể được báo hiệu trong tiêu đề hình ảnh (picture header, PH) hoặc các tham số mô tả lát có thể được báo hiệu trong tiêu đề lát (slice header, SH). PH có thể được truyền dưới dạng loại đơn vị NAL cho nó. SH có thể được cung cấp ở điểm bắt đầu của đơn vị NAL tạo cấu hình phân tải hữu ích của lát (ví dụ, dữ liệu lát).

Việc báo hiệu việc phân vùng hình ảnh

Trong một phương án, một hình ảnh có thể được phân vùng thành nhiều hình ảnh con, ô xếp và/hoặc lát. Việc báo hiệu hình ảnh con có thể được cung cấp trong tập tham số trình tự. Việc báo hiệu ô xếp và lát hình chữ nhật có thể được cung cấp trong tập tham số hình ảnh. Thêm vào đó, việc báo hiệu lát quét mảnh có thể được cung cấp trong tiêu đề lát.

Fig.25 thể hiện phương án về cú pháp dành cho tập tham số trình tự. Trong cú pháp trên Fig.25, các phần tử cú pháp là như sau.

Phần tử cú pháp `subpic_info_present_flag` có thể chỉ ra liệu thông tin hình ảnh con có có mặt hay không. Ví dụ, giá trị thứ nhất (ví dụ, 0) của `subpic_info_present_flag` có thể chỉ ra rằng thông tin hình ảnh con của trình tự video lớp được lập mã (coded layer video sequence, CLVS) không có mặt trong luồng bit, và chỉ một hình con có mặt trong hình ảnh riêng rẽ của CLVS. Giá trị thứ hai (ví dụ, 1) của `subpic_info_present_flag` có thể chỉ ra rằng thông tin hình ảnh con của trình tự video lớp được lập mã (CLVS) có mặt trong luồng bit, và ít nhất một hình con có thể có mặt trong hình ảnh riêng rẽ của CLVS.

Ở đây, CLVS có thể nghĩa là lớp của trình tự video được mã hóa. CLVS có thể là trình tự của các PU có `nuh_layer_id` giống như đơn vị dự đoán (prediction unit, PU) của hình ảnh làm tươi giải mã dần dần (gradual decoding refresh, GDR) hoặc hình ảnh điểm truy cập ngẫu nhiên nội (intra random access point, IRAP) mà nó không được xuất ra cho đến khi tín hiệu được xây dựng lại được tạo ra.

Phần tử cú pháp `sps_num_subpics_minus1` có thể chỉ ra số lượng hình ảnh con. Ví dụ, giá trị thu được bằng cách cộng 1 vào nó có thể thể hiện số lượng hình ảnh con thuộc về hình ảnh riêng rẽ này của CLVS. Giá trị của `sps_num_subpics_minus1` có thể có giá trị từ 0 đến $\text{Ceil}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}) * \text{Ceil}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}) - 1$. Khi giá trị của `sps_num_subpics_minus1` không có mặt, giá trị của `sps_num_subpics_minus1` có thể được dẫn ra là 0.

Giá trị 1 của phần tử cú pháp `sps_independent_subpics_flag` có thể chỉ ra rằng việc dự đoán nội không được thực hiện vượt ra ngoài biên của hình ảnh con trong CLVS, việc dự đoán liên không được thực hiện, và hoạt động lọc trong vòng không được thực hiện.

Giá trị 0 của phần tử cú pháp `sps_independent_subpics_flag` có thể chỉ ra rằng việc dự đoán liên hoặc hoạt động lọc trong vòng có thể được thực hiện vượt ra ngoài biên của hình ảnh con trong CLVS. Khi giá trị của `sps_independent_subpics_flag` không có mặt, giá trị của `sps_independent_subpics_flag` có thể được dẫn ra là 0.

Phần tử cú pháp `subpic_ctu_top_left_x[i]` có thể chỉ ra vị trí ngang của CTU trên cùng bên trái của hình ảnh con thứ i theo các đơn vị là `CtbSizeY`. Chiều dài của phần tử cú pháp `subpic_ctu_top_left_x[i]` có thể là $\text{Ceil}(\text{Log}_2((\text{pic_width_max_in_luma_samples} + \text{CtbSizeY} - 1) \gg \text{CtbLog2SizeY}))$ bit. Khi `subpic_ctu_top_left_x[i]` không có mặt, giá trị của nó có thể được dẫn ra là 0. Ở đây, `pic_width_max_in_luma_samples` có thể là biến số chỉ ra chiều rộng tối đa của hình ảnh được biểu diễn theo các đơn vị là mẫu độ chói. `CtbSizeY` có thể là biến số chỉ ra kích thước của kích thước đơn vị mẫu độ chói của CTB. `CtbLog2SizeY` có thể là biến số chỉ ra giá trị thu được bằng cách lấy logarit cơ số 2 của kích thước đơn vị mẫu độ chói của CTB.

Phần tử cú pháp `subpic_ctu_top_left_y[i]` có thể chỉ ra vị trí dọc của CTU trên cùng bên trái của hình ảnh con thứ i theo các đơn vị là `CtbSizeY`. Chiều dài của phần tử cú pháp `subpic_ctu_top_left_y[i]` có thể là $\text{Ceil}(\text{Log}_2((\text{pic_height_max_in_luma_samples} + \text{CtbSizeY} - 1) \gg \text{CtbLog2SizeY}))$ bit. Ở đây, `pic_height_max_in_luma_samples` có thể là biến số chỉ ra chiều cao tối đa của hình ảnh được biểu diễn theo các đơn vị là mẫu độ chói. Khi `subpic_ctu_top_left_y[i]` không có mặt, giá trị của nó có thể được dẫn ra là 0.

Giá trị thu được bằng cách cộng 1 vào phần tử cú pháp `subpic_width_minus1[i]` có thể chỉ ra chiều rộng của hình ảnh con thứ nhất và đơn vị của nó có thể là `CtbSizeY`. Chiều dài của `subpic_width_minus1[i]` có thể là

$\text{Ceil}(\text{Log2}((\text{pic_width_max_in_luma_samples} + \text{CtbSizeY} - 1) \gg \text{CtbLog2SizeY}))$ bit.

Khi giá trị của $\text{subpic_width_minus1}[i]$ không có mặt, giá trị của $\text{subpic_width_minus1}[i]$ có thể được tính là $((\text{pic_width_max_in_luma_samples} + \text{CtbSizeY} - 1) \gg \text{CtbLog2SizeY}) - \text{subpic_ctu_top_left_x}[i] - 1$.

Giá trị thu được bằng cách cộng 1 vào phần tử cú pháp $\text{subpic_height_minus1}[i]$ có thể chỉ ra chiều cao của hình ảnh con thứ nhất và đơn vị của nó có thể là CtbSizeY . Chiều dài của $\text{subpic_height_minus1}[i]$ có thể là $\text{Ceil}(\text{Log2}((\text{pic_height_max_in_luma_samples} + \text{CtbSizeY} - 1) \gg \text{CtbLog2SizeY}))$ bit. Khi $\text{subpic_height_minus1}[i]$ không có mặt, giá trị của $\text{subpic_height_minus1}[i]$ có thể được tính là $((\text{pic_height_max_in_luma_samples} + \text{CtbSizeY} - 1) \gg \text{CtbLog2SizeY}) - \text{subpic_ctu_top_left_y}[i] - 1$.

Giá trị 1 của phần tử cú pháp $\text{subpic_treated_as_pic_flag}[i]$ chỉ ra rằng hình ảnh con thứ i của mỗi hình ảnh được mã hóa trong CLVS được xử lý như một hình ảnh ngoại trừ hoạt động lọc trong vòng. Giá trị 0 của $\text{subpic_treated_as_pic_flag}[i]$ chỉ ra rằng hình ảnh con thứ i của mỗi hình ảnh được mã hóa trong CLVS không được xử lý như một hình ảnh ngoại trừ hoạt động lọc trong vòng. Khi $\text{subpic_treated_as_pic_flag}[i]$ không có mặt, giá trị của $\text{subpic_treated_as_pic_flag}[i]$ có thể được thiết lập là giá trị của $\text{sps_independent_subpics_flag}$.

Giá trị 1 của phần tử cú pháp $\text{loop_filter_across_subpic_enabled_flag}[i]$ có thể chỉ ra rằng việc lọc trong vòng có thể được thực hiện vượt ra ngoài biên của hình ảnh con thứ i trong mỗi hình ảnh được mã hóa trong CLVS. Giá trị 0 của $\text{loop_filter_across_subpic_enabled_flag}[i]$ có thể chỉ ra rằng việc lọc trong vòng không được thực hiện vượt ra ngoài biên của hình ảnh con thứ i trong mỗi hình ảnh được mã hóa trong CLVS. Khi giá trị của $\text{loop_filter_across_subpic_enabled_flag}[i]$ không có mặt, giá trị của

`loop_filter_across_subpic_enabled_flag[ i ]` có thể được xác định là `1 - sps_independent_subpics_flag`.

Fig.26 là hình thể hiện một phương án về cú pháp của tập tham số hình ảnh. Trong cú pháp trên Fig.26, các phần tử cú pháp là như sau.

Giá trị thứ nhất (ví dụ, 0) của phần tử cú pháp `no_pic_partition_flag` có thể chỉ ra rằng mỗi hình ảnh tham chiếu đến PPS có thể được phân vùng thành hai hay hơn hai ô xếp hoặc lát. Giá trị thứ hai (ví dụ, 1) của `no_pic_partition_flag` có thể chỉ ra rằng việc phân vùng hình ảnh không được áp dụng cho mỗi hình ảnh tham chiếu đến PPS.

Giá trị thu được bằng cách cộng 5 vào phần tử cú pháp `pps_log2_ctu_size_minus5` có thể chỉ ra kích thước khối lập mã độ chói của mỗi CTU. Giá trị của `pps_log2_ctu_size_minus5` có thể được giới hạn là bằng `sps_log2_ctu_size_minus5` chỉ ra cùng một giá trị trong tập tham số trình tự.

Giá trị thu được bằng cách cộng 1 vào phần tử cú pháp `num_exp_tile_columns_minus1` chỉ ra số lượng chiều rộng cột ô xếp được cung cấp tường minh. Giá trị của `num_exp_tile_columns_minus1` có thể có giá trị từ 0 đến `PicWidthInCtbsY - 1`. Khi giá trị của `no_pic_partition_flag` là 1, giá trị của `num_exp_tile_columns_minus1` có thể được dẫn ra là 0.

Giá trị thu được bằng cách cộng 1 vào phần tử cú pháp `num_exp_tile_rows_minus1` có thể chỉ ra số lượng chiều cao hàng ô xếp được cung cấp tường minh. Giá trị của `num_exp_tile_rows_minus1` có thể có giá trị từ 0 đến `PicHeightInCtbsY - 1`. Khi giá trị của `no_pic_partition_flag` là 1, giá trị của `num_exp_tile_rows_minus1` có thể được dẫn ra là 0.

Giá trị thu được bằng cách cộng 1 vào phần tử cú pháp `tile_column_width_minus1[ i ]` có thể chỉ ra chiều rộng của cột ô xếp thứ i theo các đơn vị là CTB. Ở đây, i có thể có giá trị từ 0 đến `num_exp_tile_columns_minus1 - 1`. `tile_column_width_minus1[ num_exp_tile_columns_minus1 ]` có thể được sử dụng để dẫn ra chiều rộng của ô xếp với chỉ số của cột ô xếp bằng hoặc lớn hơn `num_exp_tile_columns_minus1`. Giá trị của `tile_column_width_minus1[ i ]` có thể có giá trị từ

0 đến $\text{PicWidthInCtbsY} - 1$. Khi $\text{tile_column_width_minus1}[i]$ không được cung cấp từ luồng bit, giá trị của $\text{tile_column_width_minus1}[0]$ có thể được thiết lập là giá trị của $\text{PicWidthInCtbsY} - 1$.

Giá trị thu được bằng cách cộng 1 vào phần tử cú pháp $\text{tile_row_height_minus1}[i]$ có thể chỉ ra chiều cao của hàng ô xếp thứ i theo các đơn vị là CTB. Ở đây, i có thể có giá trị từ 0 đến $\text{num_exp_tile_rows_minus1} - 1$. $\text{tile_row_height_minus1}[\text{num_exp_tile_rows_minus1}]$ có thể được sử dụng để dẫn ra chiều cao của ô xếp với chỉ số của hàng ô xếp bằng hoặc lớn hơn $\text{num_exp_tile_rows_minus1}$. Giá trị của $\text{tile_row_height_minus1}[i]$ có thể có giá trị từ 0 đến $\text{PicHeightInCtbsY} - 1$. Khi $\text{tile_row_height_minus1}[i]$ không được cung cấp từ luồng bit, giá trị của $\text{tile_row_height_minus1}[0]$ có thể được thiết lập là giá trị của $\text{PicHeightInCtbsY} - 1$.

Giá trị 0 của phần tử cú pháp rect_slice_flag có thể chỉ ra rằng các ô xếp trong mỗi lát được quét theo thứ tự quét mảnh và thông tin lát không được báo hiệu thông qua tập tham số hình ảnh. Giá trị 1 của rect_slice_flag có thể chỉ ra rằng các lát trong mỗi lát bao trùm vùng hình chữ nhật của hình ảnh và thông tin lát được báo hiệu thông qua tập tham số hình ảnh. Ở đây, biến số NumTilesInPic có thể thể hiện số lượng ô xếp có mặt trong hình ảnh. Khi rect_slice_flag không có mặt trong luồng bit, giá trị của rect_slice_flag có thể được dẫn ra là 1. Trong khi đó, khi giá trị của $\text{subpic_info_present_flag}$ là 1, giá trị của rect_slice_flag có thể bị cưỡng ép là 1.

Giá trị 1 của phần tử cú pháp $\text{single_slice_per_subpic_flag}$ có thể chỉ ra rằng mỗi hình ảnh con gồm chỉ một lát hình chữ nhật. Giá trị 0 của $\text{single_slice_per_subpic_flag}$ có thể chỉ ra rằng mỗi hình ảnh con gồm ít nhất một lát hình chữ nhật. Khi $\text{single_slice_per_subpic_flag}$ không có mặt trong luồng bit, giá trị của $\text{single_slice_per_subpic_flag}$ có thể được dẫn ra là 0.

Giá trị thu được bằng cách cộng 1 vào phần tử cú pháp $\text{num_slices_in_pic_minus1}$ có thể chỉ ra số lượng lát trong hình ảnh. Giá trị 0 của phần tử cú pháp $\text{tile_idx_delta_present_flag}$ có thể chỉ ra rằng phần tử cú pháp $\text{tile_idx_delta}[i]$ không có mặt trong tập tham số hình ảnh và tất cả hình ảnh tham chiếu đến tập tham số hình ảnh này được phân vùng thành các hàng lát

hình chữ nhật và các cột lát hình chữ nhật theo thứ tự quét mảnh lát. Giá trị 1 của `tile_idx_delta_present_flag` có thể chỉ ra rằng phần tử cú pháp `tile_idx_delta[i]` có thể có mặt trong tập tham số hình ảnh và tất cả các lát hình chữ nhật thuộc về các hình ảnh tham chiếu đến tập tham số hình ảnh đều được chỉ rõ theo thứ tự được chỉ ra bởi giá trị của `tile_idx_delta[i]` theo giá trị `i` tăng lên. Khi `tile_idx_delta_present_flag` không có mặt, giá trị của `tile_idx_delta_present_flag` có thể được dẫn ra là 0.

Giá trị thu được bằng cách cộng 1 vào phần tử cú pháp `slice_width_in_tiles_minus1[i]` có thể chỉ ra chiều rộng của lát hình chữ nhật thứ `i` theo các đơn vị là cột ô xếp. Giá trị của `slice_width_in_tiles_minus1[i]` có thể có giá trị từ 0 đến `NumTileColumns - 1`. Ở đây, khi `i` nhỏ hơn `num_slices_in_pic_minus1` và giá trị của `NumTileColumns` là 1, giá trị của `slice_width_in_tiles_minus1[i]` có thể được dẫn ra là 0. Ở đây, biến số `NumTileColumns` có thể chỉ ra số lượng cột ô xếp có mặt trong hình ảnh hiện tại. Ở đây, biến số `NumTileRows` có thể chỉ ra số lượng hàng ô xếp có mặt trong hình ảnh hiện tại.

Giá trị thu được bằng cách cộng 1 vào phần tử cú pháp `slice_height_in_tiles_minus1[i]` có thể chỉ ra chiều cao của lát hình chữ nhật thứ `i` theo đơn vị là hàng ô xếp, khi giá trị của `num_exp_slices_in_tile[i]` là 0. Giá trị của `slice_height_in_tiles_minus1[i]` có thể có giá trị từ 0 đến `NumTileRows - 1`. Khi giá trị của `i` nhỏ hơn `num_slices_in_pic_minus1` và giá trị của `slice_height_in_tiles_minus1[i]` không được thu nhận từ luồng bit, giá trị của `slice_height_in_tiles_minus1[i]` có thể được dẫn ra bằng phương trình sau đây.

Phương trình 1

$$\text{slice_height_in_tiles_minus1}[i] = \text{NumTileRows} - 1 - \text{slice_height_in_tiles_minus1}[i-1]$$

`SliceTopLeftTileIdx` có thể là biến số chỉ ra chỉ số của ô xếp trên cùng bên trái của lát.

Phần tử cú pháp `num_exp_slices_in_tile[i]` có thể chỉ ra số lượng chiều cao của lát được cung cấp tường minh đối với các lát trong ô xếp (ví dụ, ô xếp có chỉ số ô xếp giống như `SliceTopLeftTileIdx[i]`) bao gồm lát thứ `i`. Giá trị của `num_exp_slices_in_tile[i]` có thể có giá

trị là từ 0 đến $\text{RowHeight}[\text{SliceTopLeftTileIdx}[i] / \text{NumTileColumns}] - 1$. Khi $\text{num_exp_slices_in_tile}[i]$ không được cung cấp từ luồng bit, giá trị của $\text{num_exp_slices_in_tile}[i]$ có thể được dẫn ra là 0. Ở đây, $\text{RowHeight}[i]$ có thể là biến số chỉ ra chiều cao của ô xếp thứ i theo các đơn vị là CTB. Ở đây, khi giá trị của $\text{num_exp_slices_in_tile}[i]$ là 0, ô xếp bao gồm lát thứ i có thể không được phân vùng thành nhiều ô xếp.

Giá trị thu được bằng cách cộng 1 vào phần tử cú pháp $\text{exp_slice_height_in_ctus_minus1}[i][j]$ có thể chỉ ra chiều cao của lát hình chữ nhật thứ j trong ô xếp bao gồm lát thứ i theo các đơn vị là CTU. Giá trị của $\text{exp_slice_height_in_ctus_minus1}[i][j]$ có thể có giá trị từ 0 đến $\text{RowHeight}[\text{SliceTopLeftTileIdx}[i] / \text{NumTileColumns}] - 1$.

Biến số $\text{NumSlicesInTile}[i]$ có thể chỉ ra số lượng lát có mặt trong ô xếp bao gồm lát thứ i .

Phần tử cú pháp $\text{tile_idx_delta}[i]$ có thể chỉ ra chênh lệch giữa chỉ số ô xếp của ô xếp bao gồm CTU thứ nhất trong lát hình chữ nhật thứ $(i+1)$ và chỉ số ô xếp của ô xếp bao gồm CTU thứ nhất trong lát hình chữ nhật thứ i . Giá trị của $\text{tile_idx_delta}[i]$ có thể có giá trị từ $-\text{NumTilesInPic} + 1$ đến $\text{NumTilesInPic} - 1$. Khi giá trị của $\text{tile_idx_delta}[i]$ không có mặt trong luồng bit, giá trị của $\text{tile_idx_delta}[i]$ có thể được dẫn ra là 0. Khi giá trị của $\text{tile_idx_delta}[i]$ có mặt, giá trị của $\text{tile_idx_delta}[i]$ có thể bị cưỡng ép là giá trị khác không.

Giá trị 1 của phần tử cú pháp $\text{loop_filter_across_tiles_enabled_flag}$ có thể chỉ ra rằng hoạt động lọc trong vòng có thể được thực hiện vượt ra ngoài biên ô xếp trong hình ảnh tham chiếu đến tập tham số hình ảnh. Giá trị 0 của $\text{loop_filter_across_tiles_enabled_flag}$ có thể chỉ ra rằng hoạt động lọc trong vòng không được thực hiện vượt ra ngoài biên ô xếp trong hình ảnh tham chiếu đến tập tham số hình ảnh.

Hoạt động lọc trong vòng có thể bao gồm một bộ lọc bất kỳ trong số bộ lọc khử tạo khối, bộ lọc phân bù thích ứng mẫu (sample adaptive offset, SAO) hoặc bộ lọc vòng thích ứng

(adaptive loop filter, ALF). Khi `loop_filter_across_tiles_enabled_flag` không có mặt trong luồng bit, giá trị của `loop_filter_across_tiles_enabled_flag` có thể được dẫn ra là 1.

Giá trị 1 của phần tử cú pháp `loop_filter_across_slices_enabled_flag` có thể chỉ ra rằng hoạt động lọc trong vòng có thể được thực hiện vượt ra ngoài biên lát trong hình ảnh tham chiếu đến tập tham số hình ảnh. Giá trị 0 của `loop_filter_across_slice_enabled_flag` có thể chỉ ra rằng hoạt động lọc trong vòng không được thực hiện vượt ra ngoài biên lát trong hình ảnh tham chiếu đến tập tham số hình ảnh. Hoạt động lọc trong vòng có thể bao gồm một bộ lọc bất kỳ trong số bộ lọc khử tạo khối, bộ lọc phân bù thích ứng mẫu (sample adaptive offset, SAO) hoặc bộ lọc vòng thích ứng (adaptive loop filter, ALF). Khi `loop_filter_across_slice_enabled_flag` không có mặt trong luồng bit, giá trị của `loop_filter_across_slice_enabled_flag` có thể được dẫn ra là 1.

Fig.27 là hình thể hiện phương án về cú pháp của tiêu đề lát. Trong cú pháp trên Fig.27, các phần tử cú pháp là như sau.

Phần tử cú pháp `slice_subpic_id` có thể chỉ ra ID của hình ảnh con của hình ảnh con bao gồm lát. Khi giá trị của `slice_subpic_id` có mặt trong luồng bit, giá trị của biến số `CurrSubpicIdx` có thể được dẫn ra là giá trị của `CurrSubpicIdx` với giá trị của `SubpicIdVal[ CurrSubpicIdx ]` của `slice_subpic_id`. Nếu không (`slice_subpic_id` không có mặt trong luồng bit) thì giá trị của `CurrSubpicIdx` có thể được dẫn ra là 0. Chiều dài của `slice_subpic_id` có thể là `sps_subpic_id_len_minus1 + 1` bit. Ở đây, `NumSlicesInSubpic[i]` có thể là biến số chỉ ra số lượng lát trong hình ảnh con thứ i. Biến số `CurrSubpicIdx` có thể chỉ ra chỉ số của hình ảnh con hiện tại.

Phần tử cú pháp `slice_address` chỉ ra địa chỉ lát của lát. Khi `slice_address` không được cung cấp, giá trị của `slice_address` có thể được dẫn ra là 0.

Trong khi đó, khi giá trị của `rect_slice_flag` là 0, địa chỉ lát có thể là bằng chỉ số ô xếp quét mảnh của ô xếp thứ nhất trong lát và chiều dài của phần tử cú pháp `slice_address` có thể là $\text{Ceil}(\text{Log2}(\text{NumTilesInPic}))$ bit, và `slice_address` có thể có giá trị từ 0 đến

$\text{NumTilesInPic} - 1$. Nếu không (nếu giá trị của rect_slice_flag là giá trị khác không, ví dụ, 1) thì địa chỉ của lát có thể là chỉ số lát mức hình ảnh con của lát, và chiều dài của phần tử cú pháp slice_address có thể là $\text{Ceil}(\text{Log2}(\text{NumSlicesInSubpic}[\text{CurrSubpicIdx}]))$ bit, và phần tử cú pháp slice_address có thể có giá trị từ 0 đến $\text{NumSlicesInSubpic}[\text{CurrSubpicIdx}] - 1$.

Phần tử cú pháp $\text{sh_extra_bit}[i]$ có thể có giá trị là 0 hoặc 1. Thiết bị giải mã có thể thực hiện việc giải mã bất kể giá trị của $\text{sh_extra_bit}[i]$. Vì vậy, thiết bị mã hóa cần tạo ra luồng bit sao cho việc giải mã được thực hiện bất kể giá trị của $\text{sh_extra_bit}[i]$. Ở đây, NumExtraShBits có thể là biến số chỉ ra số lượng bit cần thiết thêm để báo hiệu thông tin trong tiêu đề lát.

Giá trị thu được bằng cách cộng 1 vào phần tử cú pháp $\text{num_tiles_in_slice_minus1}$ có thể chỉ ra số lượng ô xếp trong lát, nếu có mặt. Giá trị của $\text{num_tiles_in_slice_minus1}$ có thể có giá trị từ 0 đến $\text{NumTilesInPic} - 1$.

Biến số $\text{NumCtusInCurrSlice}$ chỉ ra số lượng CTU trong lát hiện tại và danh sách $\text{CtbAddrInCurrSlice}[i]$ chỉ ra địa chỉ quét mảnh hình ảnh của CTB thứ i trong lát (trong đó, i có giá trị từ 0 đến $\text{NumCtusInCurrSlice} - 1$) có thể được dẫn ra như sau.

Bảng 2

```

if( rect_slice_flag ) {
    picLevelSliceIdx = slice_address
    for( j = 0; j < CurrSubpicIdx; j++ )
        picLevelSliceIdx += NumSlicesInSubpic[ j ]
    NumCtusInCurrSlice = NumCtusInSlice[ picLevelSliceIdx ]
    for( i = 0; i < NumCtusInCurrSlice; i++ )
        CtbAddrInCurrSlice[ i ] = CtbAddrInSlice[ picLevelSliceIdx ][ i ]
} else {
    NumCtusInCurrSlice = 0
    for( tileIdx = slice_address; tileIdx <=
slice_address + num_tiles_in_slice_minus1; tileIdx++ ) {
        tileX = tileIdx % NumTileColumns
    }
}

```

```

        tileY = tileIdx / NumTileColumns
        for( ctbY = tileRowBd[ tileY ]; ctbY < tileRowBd[ tileY + 1 ];
        ctbY++ ) {
            for( ctbX = tileColBd[ tileX ]; ctbX < tileColBd[ tileX + 1 ];
            ctbX++ ) {
                CtbAddrInCurrSlice[ NumCtusInCurrSlice ] =
                ctbY * PicWidthInCtb + ctbX
                NumCtusInCurrSlice++
            }
        }
    }
}

```

Các biến số SubpicLeftBoundaryPos, SubpicTopBoundaryPos, SubpicRightBoundaryPos, và SubpicBotBoundaryPos có thể được dẫn ra theo thuật toán sau đây.

Bảng 3

```

    if( subpic_treated_as_pic_flag[ CurrSubpicIdx ] ) {
        SubpicLeftBoundaryPos =
        subpic_ctu_top_left_x[ CurrSubpicIdx ] * CtbSizeY
        SubpicRightBoundaryPos = Min( pic_width_max_in_luma_samples - 1,
        ( subpic_ctu_top_left_x[ CurrSubpicIdx ] +
        subpic_width_minus1[ CurrSubpicIdx ] + 1 ) * CtbSizeY - 1 )
        SubpicTopBoundaryPos =
        subpic_ctu_top_left_y[ CurrSubpicIdx ] * CtbSizeY
        SubpicBotBoundaryPos = Min( pic_height_max_in_luma_samples - 1,
        ( subpic_ctu_top_left_y[ CurrSubpicIdx ] +
        subpic_height_minus1[ CurrSubpicIdx ] + 1 ) * CtbSizeY - 1 )
    }
}

```

Sự cải thiện về việc báo hiệu việc phân vùng hình ảnh

Việc báo hiệu liên quan đến việc phân vùng hình ảnh được mô tả trên đây có vấn đề là thông tin không cần thiết được báo hiệu khi lát là lát hình chữ nhật. Ví dụ, khi lát là lát tứ giác (ví dụ, hình chữ nhật), chiều rộng của lát riêng rẽ có thể được báo hiệu theo các đơn vị là các ô xếp. Tuy nhiên, khi ô xếp trên cùng bên trái của lát là ô xếp của cột ô xếp cuối cùng, chiều rộng của lát có thể không là giá trị khác với một đơn vị ô xếp. Ví dụ, trong trường hợp này, chiều rộng của lát có thể có chỉ giá trị chiều rộng được dẫn ra trong một đơn vị ô xếp. Theo đó, chiều rộng của lát như vậy có thể không được báo hiệu hoặc có thể được giới hạn là một đơn vị ô xếp.

Tương tự, khi lát là lát tứ giác (ví dụ, hình chữ nhật), chiều rộng của lát riêng rẽ có thể được báo hiệu theo các đơn vị ô xếp. Tuy nhiên, khi ô xếp ở vị trí trên cùng bên trái của lát là ô xếp của hàng ô xếp cuối cùng, chiều cao của lát này có thể không là giá trị khác với một đơn vị ô xếp. Theo đó, chiều cao của lát như vậy có thể không được báo hiệu hoặc có thể được giới hạn là một đơn vị ô xếp.

Phương pháp sau đây là có thể áp dụng được để giải quyết các vấn đề được mô tả trên đây. Phương án sau đây là có thể áp dụng được khi lát là lát tứ giác (ví dụ, hình chữ nhật) và chiều rộng và/hoặc chiều cao của lát riêng rẽ được báo hiệu theo các đơn vị là các ô xếp. Các phương pháp sau đây có thể được áp dụng riêng rẽ hoặc có thể được kết hợp với ít nhất một phương án khác.

Phương pháp 1. Khi ô xếp thứ nhất (ví dụ, ô xếp của góc trên cùng bên trái) của lát hình chữ nhật là ô xếp được đặt tại cột ô xếp cuối cùng của hình ảnh, việc báo hiệu chiều rộng của lát có thể không được cung cấp. Trong trường hợp này, chiều rộng của lát có thể được dẫn ra là một đơn vị ô xếp.

Ví dụ, phần tử cú pháp `slice_width_in_tiles_minus1[i]` có thể không có mặt trong luồng bit. Giá trị của phần tử cú pháp `slice_width_in_tiles_minus1[i]` có thể được dẫn ra là 0.

Phương pháp 2. Kể cả khi ô xếp thứ nhất (ví dụ, ô xếp của góc trên cùng bên trái) của lát hình chữ nhật là ô xếp được đặt tại cột ô xếp cuối cùng của hình ảnh, việc báo hiệu chiều rộng

của lát này có thể được cung cấp. Tuy nhiên, trong trường hợp này, chiều rộng của lát có thể được giới hạn là một đơn vị ô xếp.

Ví dụ, phần tử cú pháp `slice_width_in_tiles_minus1[i]` có thể có mặt trong luồng bit và do đó được phân tích cú pháp. Tuy nhiên, giá trị của phần tử cú pháp `slice_width_in_tiles_minus1[i]` có thể được giới hạn là 0.

Phương pháp 3. Khi ô xếp thứ nhất (ví dụ, ô xếp của góc trên cùng bên trái) của lát hình chữ nhật là ô xếp được đặt tại hàng ô xếp cuối cùng của hình ảnh, việc báo hiệu chiều cao của lát có thể không được cung cấp. Trong trường hợp này, chiều cao của lát có thể được dẫn ra là một đơn vị ô xếp.

Ví dụ, phần tử cú pháp `slice_height_in_tiles_minus1[i]` có thể không có mặt trong luồng bit. Giá trị của phần tử cú pháp `slice_height_in_tiles_minus1[i]` có thể được dẫn ra là 0.

Phương pháp 4. Khi ô xếp thứ nhất (ví dụ, ô xếp của góc trên cùng bên trái) của lát hình chữ nhật là ô xếp được đặt tại hàng ô xếp cuối cùng của hình ảnh, việc báo hiệu chiều cao của lát có thể được cung cấp. Tuy nhiên, trong trường hợp này, chiều cao của lát có thể được giới hạn là một đơn vị ô xếp.

Ví dụ, phần tử cú pháp `slice_height_in_tiles_minus1[i]` có thể có mặt trong luồng bit và do đó được phân tích cú pháp. Tuy nhiên, giá trị của phần tử cú pháp `slice_height_in_tiles_minus1[i]` có thể được giới hạn là bằng 0.

Trong một phương án, các phương án được mô tả trên đây là có thể áp dụng được cho phương pháp mã hóa và giải mã như được thể hiện trên Fig.28 và Fig.29. Thiết bị mã hóa theo một phương án có thể dẫn ra lát và/hoặc ô xếp trong hình ảnh hiện tại (S2810). Thêm vào đó, thiết bị mã hóa có thể mã hóa hình ảnh hiện tại dựa trên lát và/hoặc ô xếp được dẫn ra (S2820).

Tương tự, thiết bị giải mã theo một phương án có thể thu nhận thông tin video/ảnh từ luồng bit (S2910). Thêm vào đó, thiết bị giải mã có thể dẫn ra lát và/hoặc ô xếp có mặt trong hình ảnh hiện tại dựa trên thông tin video/ảnh (bao gồm thông tin về lát và/hoặc ô xếp) (S2920). Thêm vào đó, thiết bị giải mã có thể xây dựng lại và/hoặc giải mã hình ảnh hiện tại dựa trên lát

và/hoặc ô xếp (S2930).

Đối với việc xử lý trên đây của thiết bị mã hóa và thiết bị giải mã, thông tin về lát và/hoặc ô xếp có thể bao gồm thông tin và cú pháp được mô tả trên đây. Thông tin video hoặc ảnh có thể bao gồm HLS. HLS có thể bao gồm thông tin về lát và/hoặc thông tin về ô xếp. HLS có thể còn bao gồm thông tin về hình ảnh con. Thông tin về lát có thể bao gồm thông tin chỉ rõ ít nhất một lát thuộc về hình ảnh hiện tại. Thêm vào đó, thông tin về ô xếp có thể bao gồm thông tin chỉ rõ ít nhất một ô xếp thuộc về hình ảnh hiện tại. Thông tin về hình ảnh con có thể bao gồm thông tin chỉ rõ ít nhất một hình ảnh con thuộc về hình ảnh hiện tại. Ô xếp bao gồm ít nhất một lát có thể có mặt trong một hình ảnh.

Ví dụ, trong S2930 trên Fig.29, hình ảnh hiện tại có thể được xây dựng lại và/hoặc được giải mã dựa trên lát và/hoặc ô xếp được dẫn ra. Bằng cách phân vùng một hình ảnh, có thể thu được hiệu suất mã hóa và giải mã theo các khía cạnh khác nhau.

Ví dụ, hình ảnh có thể được phân vùng cho việc xử lý song song và khả năng phục hồi lỗi. Trong trường hợp xử lý song song, một số phương án được thực hiện trong CPU đa lõi có thể đòi hỏi việc phân vùng hình ảnh nguồn thành các ô xếp và/hoặc các lát. Các lát và/hoặc các ô xếp riêng rẽ có thể được xử lý song song trong các lõi khác nhau. Điều này là rất hiệu quả để thực hiện việc lập mã video thời gian thực độ phân giải cao mà nó không thể được thực hiện bằng các phương pháp khác. Thêm vào đó, việc phân vùng này có ưu điểm là làm giảm các ràng buộc về bộ nhớ, bằng cách giảm thông tin được chia sẻ giữa các ô xếp. Bởi vì các ô xếp được phân bổ cho các mạch trình khác nhau trong khi việc xử lý song song được thực hiện, kiến trúc song song là hữu ích qua cơ chế phân vùng của nó. Ví dụ, trong quy trình dẫn ra thông tin chuyển động trong việc dự đoán liên, các khối lân cận có mặt trong các lát và/hoặc ô xếp khác nhau có thể được giới hạn để không được sử dụng. Thông tin ngữ cảnh được sử dụng để mã hóa thông tin và/hoặc phân tử cú pháp có thể được khởi tạo cho mỗi lát và/hoặc ô xếp.

Khả năng phục hồi lỗi có thể được gây ra bằng cách áp dụng việc bảo vệ lỗi không bằng nhau (unequal error protection, UEP) cho lát và/hoặc ô xếp được mã hóa.

Phương án 1

Ở dưới đây, phương án dựa trên các phương pháp 1 và 3 được mô tả trên đây sẽ được mô tả. Phương án sau đây có thể được áp dụng để cải thiện kỹ thuật mã hóa/giải mã chẳng hạn như đặc tả VVC.

Trong một phương án, bảng cú pháp để báo hiệu tập tham số hình ảnh có thể được thiết lập như được thể hiện trên Fig.30. Trong một phương án khác, bảng cú pháp để báo hiệu tập tham số hình ảnh có thể được thiết lập như thể hiện trên Fig.31.

Trong phương án trên Fig.30, đối với i có giá trị từ 0 đến $\text{num_slices_in_pic_minus1}-1$, khi giá trị của NumTileColumns lớn hơn 1 và giá trị của $\text{SliceTopLeftTileIdx}[i] \% \text{NumTileColumns}$ không là $\text{NumTileColumns} - 1$, phần tử cú pháp $\text{slice_width_in_tiles_minus1}[i]$ có thể được thu nhận tuần tự đối với i .

Thêm vào đó, đối với i có giá trị từ 0 đến $\text{num_slices_in_pic_minus1}-1$, khi giá trị của NumTileRows lớn hơn 1, giá trị của $\text{tile_idx_delta_present_flag}$ là 1 hoặc giá trị của $\text{SliceTopLeftTileIdx}[i] \% \text{NumTileColumns}$ là 0 và giá trị của $\text{SliceTopLeftTileIdx}[i] / \text{NumTileColumns}$ không là $\text{NumTileRows} - 1$, phần tử cú pháp $\text{slice_height_in_tiles_minus1}[i]$ có thể được thu nhận tuần tự đối với i .

Trong phương án trên Fig.30 và Fig.31, phần tử cú pháp $\text{slice_width_in_tiles_minus1}[i]$ có thể là phần tử cú pháp chỉ ra chiều rộng của lát hình chữ nhật thứ i . Ví dụ, giá trị thu được bằng cách cộng 1 vào $\text{slice_width_in_tiles_minus1}[i]$ có thể chỉ ra chiều rộng của lát hình chữ nhật thứ i theo các đơn vị là các cột ô xếp. Giá trị của $\text{slice_width_in_tiles_minus1}[i]$ có thể có giá trị từ 0 đến $\text{NumTileColumns}-1$. Khi $\text{slice_width_in_tiles_minus1}[i]$ không được thu nhận từ luồng bit, giá trị của $\text{slice_width_in_tiles_minus1}[i]$ có thể được dẫn ra là 0.

Khi định nghĩa của $\text{slice_width_in_tiles_minus1}[i]$ được thay đổi như trên, các ràng buộc rằng, “khi i nhỏ hơn $\text{num_slices_in_pic_minus1}$ và giá trị của NumTileColumns là bằng 1, giá trị của $\text{slice_width_in_tiles_minus1}[i]$ được dẫn ra là 0” có thể được lược bỏ. Do đó, như trong phương án trên Fig.31, điều kiện “ $\text{NumTileColumns} > 1$ ” có thể được xóa khỏi cú

pháp tập tham số hình ảnh.

$\text{slice_height_in_tiles_minus1}[i]$ có thể là phần tử cú pháp chỉ ra chiều cao của lát hình chữ nhật thứ i . Ví dụ, khi giá trị của $\text{num_exp_slices_in_tile}[i]$ là 0, giá trị thu được bằng cách cộng 1 vào $\text{slice_height_in_tiles_minus1}[i]$ có thể chỉ ra chiều cao của lát hình chữ nhật thứ i theo các đơn vị là các hàng ô xếp. Giá trị của $\text{slice_height_in_tiles_minus1}[i]$ có thể có giá trị từ 0 đến $\text{NumTileRows}-1$.

Khi $\text{slice_height_in_tiles_minus1}[i]$ không được thu nhận từ luồng bit, giá trị của $\text{slice_height_in_tiles_minus1}[i]$ có thể được dẫn ra như sau.

Đầu tiên, khi giá trị của NumTileRow là 1 hoặc giá trị của $\text{SliceTopLeftTileIdx}[i] \% \text{NumTileColumns}$ là $\text{NumTileColumns}-1$, giá trị của $\text{slice_height_in_tiles_minus1}[i]$ có thể được dẫn ra là 0.

Nếu không (ví dụ, khi giá trị của NumTileRow không là 1 và giá trị của $\text{SliceTopLeftTileIdx}[i] \% \text{NumTileColumns}$ không là $\text{NumTileColumns}-1$) thì giá trị của $\text{slice_height_in_tiles_minus1}[i]$ có thể được dẫn ra là $\text{slice_height_in_tiles_minus1}[i-1]$. Ví dụ, giá trị của $\text{slice_height_in_tiles_minus1}[i]$ có thể được thiết lập là $\text{slice_height_in_tiles_minus1}[i-1]$ mà nó là giá trị chiều cao của lát trước. Ví dụ, các giá trị của $\text{slice_height_in_tiles_minus1}[i]$ của tất cả các lát trong một ô xếp có thể được thiết lập bằng nhau.

Khi định nghĩa của $\text{slice_height_in_tiles_minus1}[i]$ được thay đổi như trên, ràng buộc hiện có rằng, khi “ i nhỏ hơn $\text{num_slices_in_pic_minus1}$ và giá trị của NumTileRows là bằng 1, giá trị của $\text{slice_width_in_tiles_minus1}[i]$ được dẫn ra là 0” có thể được lược bỏ. Do đó, như trong phương án trên Fig.31, điều kiện “ $\text{NumTileRows} > 1$ ” có thể được xóa khỏi cú pháp tập tham số hình ảnh.

Phương án 2

Ở dưới đây, phương án dựa trên các phương pháp 2 và 4 được mô tả trên đây sẽ được mô tả. Phương án sau đây có thể được áp dụng để cải thiện kỹ thuật mã hóa/giải mã chẳng hạn

nhu đặc tả VVC.

Trong một phương án, $\text{slice_width_in_tiles_minus1}[i]$ có thể là phần tử cú pháp chỉ ra chiều rộng của lát hình chữ nhật thứ i . Ví dụ, giá trị thu được bằng cách cộng 1 vào $\text{slice_width_in_tiles_minus1}[i]$ có thể chỉ ra chiều rộng của lát hình chữ nhật thứ i theo các đơn vị là các cột ô xếp. Giá trị của $\text{slice_width_in_tiles_minus1}[i]$ có thể có giá trị từ 0 đến $\text{NumTileColumns}-1$. Khi $\text{slice_width_in_tiles_minus1}[i]$ không được thu nhận từ luồng bit, giá trị của $\text{slice_width_in_tiles_minus1}[i]$ có thể được dẫn ra là 0.

Tại thời điểm này, khi i nhỏ hơn $\text{num_slices_in_pic_minus1}$ và giá trị của NumTileColumns là bằng 1, giá trị của $\text{slice_width_in_tiles_minus1}[i]$ có thể được dẫn ra là 0. Thêm vào đó, đối với sự hợp cách của luồng bit, khi ô xếp thứ nhất của lát hình chữ nhật thứ i là ô xếp cuối cùng của cột ô xếp, giá trị của $\text{slice_width_in_tiles_minus1}[i]$ có thể được cưỡng ép là 0.

$\text{slice_height_in_tiles_minus1}[i]$ có thể là phần tử cú pháp chỉ ra chiều cao của lát hình chữ nhật thứ i . Ví dụ, khi giá trị của $\text{num_exp_slices_in_tile}[i]$ là 0, giá trị thu được bằng cách cộng 1 vào $\text{slice_height_in_tiles_minus1}[i]$ có thể chỉ ra chiều cao của lát hình chữ nhật thứ i theo các đơn vị là các hàng ô xếp. Giá trị của $\text{slice_height_in_tiles_minus1}[i]$ có thể có giá trị từ 0 đến $\text{NumTileRows}-1$.

Tại thời điểm này, khi i nhỏ hơn $\text{num_slices_in_pic_minus1}$ và giá trị của $\text{slice_height_in_tiles_minus1}[i]$ không được thu nhận từ luồng bit, giá trị của $\text{slice_height_in_tiles_minus1}[i]$ có thể được xác định theo giá trị của NumTileRows . Ví dụ, điều này có thể là được xác định như được thể hiện trong phương trình sau đây.

Phương trình 2

$$\text{slice_height_in_tiles_minus1}[i] = \text{NumTileRows} - 1 \quad ? \quad 0 : \text{slice_height_in_tiles_minus1}[i-1]$$

Thêm vào đó, đối với sự hợp cách của luồng bit, khi ô xếp thứ nhất của lát hình chữ nhật thứ i là ô xếp cuối cùng của hàng ô xếp, giá trị của $\text{slice_height_in_tiles_minus1}[i]$ có thể được

cường ép là 0.

Phương pháp mã hóa và giải mã

Ở dưới đây, phương pháp mã hóa và giải mã ảnh được thực hiện bởi thiết bị mã hóa ảnh và thiết bị giải mã ảnh theo một phương án sẽ được mô tả.

Đầu tiên, hoạt động của thiết bị giải mã sẽ được mô tả. Thiết bị giải mã ảnh theo một phương án có thể bao gồm bộ nhớ và bộ xử lý, và thiết bị giải mã có thể thực hiện việc giải mã bằng hoạt động của bộ xử lý. Fig.32 là hình thể hiện phương án về phương pháp giải mã theo một phương án.

Thiết bị giải mã theo một phương án có thể thu nhận phần tử cú pháp `no_pic_partition_flag` chỉ ra tính khả dụng của việc phân vùng hình ảnh hiện tại từ luồng bit. Thiết bị giải mã có thể xác định tính khả dụng của việc phân vùng hình ảnh hiện tại dựa trên giá trị của `no_pic_partition_flag` như được mô tả trên đây (S3210).

Khi việc phân vùng hình ảnh hiện tại là khả dụng, thiết bị giải mã có thể thu nhận, từ luồng bit, phần tử cú pháp `num_exp_tile_rows_minus1` chỉ ra số lượng hàng ô xếp phân vùng hình ảnh hiện tại và phần tử cú pháp `num_exp_tile_columns_minus1` chỉ ra số lượng cột ô xếp, và xác định số lượng hàng và cột ô xếp từ chúng như được mô tả trên đây (S3220).

Dựa trên số lượng cột ô xếp, thiết bị giải mã có thể thu nhận, từ luồng bit, phần tử cú pháp `tile_column_width_minus1` chỉ ra chiều rộng của mỗi cột ô xếp phân vùng hình ảnh hiện tại, và xác định chiều rộng của mỗi cột ô xếp từ nó như được mô tả trên đây (S3230).

Dựa trên số lượng hàng ô xếp, thiết bị giải mã có thể thu nhận, từ luồng bit, phần tử cú pháp `tile_row_height_minus1[i]` chỉ ra chiều cao của mỗi hàng ô xếp phân vùng hình ảnh hiện tại, và xác định chiều cao của mỗi hàng ô xếp từ nó (S3240). Thêm vào đó, thiết bị giải mã có thể tính số lượng ô xếp phân vùng hình ảnh hiện tại bằng tích số của số lượng cột ô xếp và số lượng hàng ô xếp.

Tiếp theo, thiết bị giải mã có thể thu nhận phần tử cú pháp `rect_slice_flag` chỉ ra liệu hình ảnh hiện tại có được phân vùng thành các lát hình chữ nhật hay không dựa trên việc liệu số

lượng ô xếp phân vùng hình ảnh hiện tại có lớn hơn 1 hay không, và xác định xem liệu hình ảnh hiện tại có được phân vùng thành các lát hình chữ nhật hay không từ giá trị của nó như được mô tả trên đây (S3250).

Tiếp theo, thiết bị giải mã có thể thu nhận, từ luồng bit, phần tử cú pháp `num_slices_in_pic_minus1` chỉ ra số lượng lát phân vùng hình ảnh hiện tại dựa trên việc liệu hình ảnh hiện tại có được phân vùng thành các lát hình chữ nhật hay không, và xác định số lượng lát phân vùng hình ảnh hiện tại từ nó như được mô tả trên đây (S3260).

Tiếp theo, thiết bị giải mã có thể thu nhận, từ luồng bit, thông tin kích thước chỉ ra kích thước của mỗi lát phân vùng hình ảnh hiện tại từ luồng bit nhiều bằng số lượng lát phân vùng hình ảnh hiện tại (S3270).

Ở đây, thông tin kích thước có thể bao gồm phần tử cú pháp `slice_width_in_tiles_minus1[i]` mà nó là thông tin chiều rộng chỉ ra chiều rộng của lát và phần tử cú pháp `slice_height_in_tiles_minus1[i]` mà nó là thông tin chiều cao chỉ ra chiều cao của lát. `slice_width_in_tiles_minus1[i]` có thể chỉ ra chiều rộng của lát theo các đơn vị là các cột ô xếp và `slice_height_in_tiles_minus1[i]` có thể chỉ ra chiều cao của lát theo các đơn vị là các hàng ô xếp.

Ở đây, khi thiết bị giải mã thu nhận thông tin kích thước của lát hiện tại (ví dụ, lát thứ *i*) từ luồng bit, thiết bị giải mã có thể thu nhận `slice_width_in_tiles_minus1[i]` từ luồng bit dựa trên việc liệu ô xếp trên cùng bên trái của lát hiện tại có thuộc về cột ô xếp cuối cùng của hình ảnh hiện tại hay không.

Ví dụ, khi chỉ số ô xếp trên cùng bên trái (ví dụ, `SliceTopLeftTileIdx`) của lát hiện tại không là chỉ số ô xếp tương ứng với cột cuối cùng của cột ô xếp thuộc về hình ảnh hiện tại, `slice_width_in_tiles_minus1[i]` có thể được thu nhận từ luồng bit. Tuy nhiên, khi chỉ số ô xếp trên cùng bên trái của lát hiện tại là chỉ số ô xếp tương ứng với cột cuối cùng của cột ô xếp thuộc về hình ảnh hiện tại, `slice_width_in_tiles_minus1[i]` có thể không được thu nhận từ luồng bit và có thể được xác định là 0.

Tương tự, thiết bị giải mã có thể thu nhận `slice_height_in_tiles_minus1[i]` từ luồng bit dựa trên việc liệu ô xếp trên cùng bên trái của lát hiện tại có thuộc về hàng ô xếp cuối cùng của hình ảnh hiện tại hay không.

Ví dụ, khi chỉ số ô xếp trên cùng bên trái của lát hiện tại không là chỉ số ô xếp tương ứng với hàng cuối cùng của hàng ô xếp thuộc về hình ảnh hiện tại, `slice_height_in_tiles_minus1[i]` có thể được thu nhận từ luồng bit. Tuy nhiên, khi chỉ số ô xếp trên cùng bên trái của lát hiện tại là chỉ số ô xếp tương ứng với hàng cuối cùng của hàng ô xếp thuộc về hình ảnh hiện tại, `slice_height_in_tiles_minus1[i]` có thể không được thu nhận từ luồng bit và có thể được xác định là 0.

Tiếp theo, thiết bị giải mã có thể xác định kích thước của mỗi lát phân vùng hình ảnh hiện tại dựa trên thông tin kích thước và giải mã lát được xác định này, qua đó giải mã ảnh. Ví dụ, CTU được chứa trong lát có kích thước được xác định có thể được giải mã nhờ sử dụng việc dự đoán liên hoặc nội, qua đó giải mã lát này (S3280).

Trong khi đó, `SliceTopLeftTileIdx` có thể là biến số chỉ ra ô xếp trên cùng bên trái của lát và có thể được xác định bởi thuật toán trên Fig.33 và Fig.34. Thuật toán trên Fig.33 và Fig.34 là một thuật toán liên tiếp.

Tiếp theo, hoạt động của thiết bị mã hóa sẽ được mô tả. Thiết bị mã hóa ảnh theo một phương án có thể bao gồm bộ nhớ và bộ xử lý, và thiết bị mã hóa có thể thực hiện việc mã hóa theo cách thức tương ứng với việc giải mã của thiết bị giải mã bằng hoạt động của bộ xử lý này. Ví dụ, như được thể hiện trên Fig.35, thiết bị mã hóa có thể mã hóa hình ảnh hiện tại. Đầu tiên, thiết bị mã hóa có thể xác định cột ô xếp và hàng ô xếp cho hình ảnh hiện tại (S3510). Tiếp theo, lát phân vùng ảnh có thể được xác định (S3520). Tiếp theo, thiết bị mã hóa có thể tạo ra luồng bit bao gồm thông tin được xác định trước bao gồm thông tin kích thước của lát (S3530). Ví dụ, thiết bị mã hóa có thể tạo ra luồng bit bao gồm `no_pic_partition_flag`, `num_exp_tile_rows_minus1`, `num_exp_tile_columns_minus1`, `tile_column_width_minus1`, `tile_row_height_minus1[i]`, `rect_slice_flag`, `num_slices_in_pic_minus1`,

$\text{slice_width_in_tiles_minus1}[i]$ và $\text{slice_height_in_tiles_minus1}[i]$ mà chúng là các phần tử cú pháp được thu nhận từ luồng bit bởi thiết bị giải mã.

Tại thời điểm này, thông tin kích thước có thể được đưa vào trong luồng bit dựa trên việc liệu lát hiện tại thuộc về cột ô xếp cuối cùng hay hàng ô xếp cuối cùng của hình ảnh hiện tại. Ví dụ, thiết bị mã hóa có thể mã hóa luồng bit dựa trên việc liệu ô xếp trên cùng bên trái của lát hiện tại có là cột ô xếp cuối cùng và/hoặc hàng ô xếp cuối cùng hay không, sao cho $\text{slice_width_in_tiles_minus1}[i]$ và $\text{slice_height_in_tiles_minus1}[i]$ tương ứng với phần mô tả của thiết bị giải mã. Ở đây, lát hiện tại có thể là lát hình chữ nhật.

Phương án ứng dụng

Trong khi các phương pháp làm ví dụ của sáng chế được mô tả trên đây được thể hiện dưới dạng chuỗi các hoạt động nhằm làm rõ phần mô tả, điều này không được dự định để giới hạn thứ tự theo đó các bước được thực hiện, và các bước có thể được thực hiện đồng thời hoặc theo thứ tự khác nếu cần. Để thi hành phương pháp theo sáng chế, các bước được mô tả có thể còn bao gồm các bước khác, có thể bao gồm các bước còn lại ngoại trừ một số bước, hoặc có thể bao gồm các bước bổ sung khác ngoại trừ một số bước.

Trong sáng chế này, thiết bị mã hóa ảnh hoặc thiết bị giải mã ảnh mà nó thực hiện hoạt động (bước) được xác định trước có thể thực hiện hoạt động (bước) để khẳng định điều kiện hoặc tình huống thực thi hoạt động (bước) tương ứng. Ví dụ, nếu có được mô tả rằng hoạt động được xác định trước được thực hiện khi điều kiện được xác định trước được thỏa mãn thì thiết bị mã hóa ảnh hoặc thiết bị giải mã ảnh có thể thực hiện hoạt động được xác định trước này sau khi xác định xem liệu điều kiện được xác định trước này có được thỏa mãn hay không.

Các phương án khác nhau của sáng chế không phải là danh sách của tất cả những sự kết hợp khả dĩ và được nhằm để mô tả các khía cạnh đại diện của sáng chế, và các vấn đề được mô tả theo các phương án khác nhau này có thể được áp dụng theo cách độc lập hoặc theo sự kết hợp của hai hay hơn hai phương án.

Các phương án khác nhau của sáng chế có thể được thi hành trong phần cứng, phần sụn,

phần mềm hoặc sự kết hợp của chúng. Trong trường hợp thi hành sáng chế bằng phần cứng, sáng chế có thể được thi hành bằng các mạch tích hợp chuyên dụng (application specific integrated circuit, ASIC), các bộ xử lý tín hiệu số (digital signal processor, DSP), các thiết bị xử lý tín hiệu số (digital signal processing device, DSPD), các thiết bị logic lập trình được (programmable logic device, PLD), các mảng công lập trình được dạng trường (field programmable gate array, FPGA), các bộ xử lý chung, các bộ điều khiển, các bộ vi điều khiển, các bộ vi xử lý, v.v.

Thêm vào đó, thiết bị giải mã ảnh và thiết bị mã hóa ảnh, mà các phương án của sáng chế được áp dụng với chúng, có thể được chứa trong thiết bị truyền và nhận phát quảng bá đa phương tiện, thiết bị đầu cuối truyền thông di động, thiết bị video rạp chiếu phim gia đình, thiết bị video rạp chiếu phim số, camera giám sát, thiết bị trò chuyện video, thiết bị truyền thông theo thời gian thực chẳng hạn như truyền thông video, thiết bị tạo luồng di động, phương tiện lưu trữ, máy quay video, thiết bị cung cấp dịch vụ video theo yêu cầu (Video on demand, VoD), thiết bị video trên cùng (OTT, over the top), thiết bị cung cấp dịch vụ tạo luồng Internet, thiết bị video ba chiều (three-dimensional, 3D), thiết bị video điện thoại video, thiết bị video y tế, và loại tương tự, và có thể được sử dụng để xử lý các tín hiệu video hoặc các tín hiệu dữ liệu. Ví dụ, các thiết bị video OTT có thể bao gồm bảng điều khiển trò chơi, máy đọc đĩa Blu-ray, TV truy cập Internet, hệ thống rạp hát gia đình, điện thoại thông minh, PC dạng bảng, đầu ghi video số (Digital video recorder, DVR), hoặc loại tương tự.

Fig.36 là hình thể hiện hệ thống tạo luồng nội dung, mà phương án của sáng chế có thể áp dụng được với nó.

Như được thể hiện trên Fig.36, hệ thống tạo luồng nội dung, mà phương án của sáng chế này được áp dụng với nó, có thể phần lớn bao gồm máy chủ mã hóa, máy chủ tạo luồng, máy chủ web, bộ lưu trữ phương tiện, thiết bị người dùng, và thiết bị đầu vào đa phương tiện.

Máy chủ mã hóa nén nội dung được nhập vào từ các thiết bị đầu vào đa phương tiện như điện thoại thông minh, camera, máy quay video, v.v. thành dữ liệu số để tạo ra luồng bit và

truyền luồng bit này tới máy chủ tạo luồng. Như là một ví dụ khác, khi các thiết bị đầu vào đa phương tiện như điện thoại thông minh, camera, máy quay video, v.v. trực tiếp tạo ra luồng bit, máy chủ mã hóa có thể được lược bỏ.

Luồng bit có thể được tạo ra bởi phương pháp mã hóa ảnh hoặc thiết bị mã hóa ảnh mà phương án của sáng chế này được áp dụng với chúng, và máy chủ tạo luồng có thể lưu trữ tạm thời luồng bit trong quy trình truyền hoặc nhận luồng bit.

Máy chủ tạo luồng truyền dữ liệu đa phương tiện đến thiết bị người dùng dựa trên yêu cầu của người dùng qua máy chủ web, và máy chủ web đóng vai trò như là phương tiện để thông báo cho người dùng về dịch vụ. Khi người dùng yêu cầu dịch vụ mong muốn từ máy chủ web, máy chủ web có thể phân phát nó đến máy chủ tạo luồng, và máy chủ tạo luồng có thể truyền dữ liệu đa phương tiện đến người dùng. Trong trường hợp này, hệ thống tạo luồng nội dung có thể bao gồm máy chủ điều khiển riêng biệt. Trong trường hợp này, máy chủ điều khiển này đóng vai trò điều khiển mệnh lệnh/phản hồi giữa các thiết bị trong hệ thống tạo luồng nội dung.

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

Các ví dụ về thiết bị người dùng có thể bao gồm điện thoại di động, điện thoại thông minh, máy tính xách tay, thiết bị đầu cuối phát quảng bá kỹ thuật số, thiết bị trợ giúp số cá nhân (personal digital assistant, PDA), thiết bị trình diễn đa phương tiện cầm tay (portable multimedia player, PMP), thiết bị định vị, PC dạng phiên, các PC dạng bảng, các máy tính siêu di động (ultrabook), các thiết bị đeo được (ví dụ, các đồng hồ thông minh, các kính thông minh, các thiết bị hiển thị gắn trên đầu), các TV kỹ thuật số, các máy tính để bàn, bảng ký hiệu kỹ thuật số, và dạng tương tự.

Mỗi máy chủ trong hệ thống tạo luồng nội dung có thể được vận hành dưới dạng máy

chủ phân tán, mà trong trường hợp đó dữ liệu được nhận từ mỗi máy chủ có thể được phân tán.

Phạm vi của sáng chế bao gồm phần mềm hoặc các mệnh lệnh có thể thực thi được bằng máy (ví dụ, hệ điều hành, ứng dụng, phần sụn, chương trình, v.v.) để cho phép các hoạt động theo các phương pháp của các phương án khác nhau được thực thi trên thiết bị hoặc máy tính, phương tiện phi chuyển tiếp đọc được bằng máy tính có phần mềm hoặc các mệnh lệnh như vậy được lưu trữ trên đó và có thể thực thi được trên thiết bị hoặc máy tính này.

Khả năng ứng dụng công nghiệp

Các phương án của sáng chế có thể được sử dụng để mã hóa hoặc giải mã ảnh.

YÊU CẦU BẢO HỘ

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

thu nhận thông tin kích thước chỉ ra kích thước của lát hiện tại tương ứng với ít nhất một phần của hình ảnh hiện tại từ luồng bit; và

xác định kích thước của lát hiện tại dựa trên thông tin kích thước này,

trong đó thông tin kích thước này bao gồm thông tin chiều rộng chỉ ra chiều rộng của lát hiện tại theo các đơn vị là các cột ô xếp và thông tin chiều cao chỉ ra chiều cao của lát hiện tại theo các đơn vị là các hàng ô xếp, và

trong đó bước thu nhận thông tin kích thước được thực hiện dựa trên việc liệu lát hiện tại thuộc về cột ô xếp cuối cùng hay hàng ô xếp cuối cùng của hình ảnh hiện tại,

trong đó thông tin chiều cao của lát hiện tại không được thu nhận từ luồng bit dựa trên việc ô xếp trên cùng bên trái của lát hiện tại thuộc về hàng ô xếp cuối cùng của hình ảnh hiện tại.

2. Phương pháp giải mã ảnh theo điểm 1, trong đó thông tin chiều cao của lát hiện tại được thu nhận từ luồng bit dựa trên việc ô xếp trên cùng bên trái của lát hiện tại không thuộc về hàng ô xếp cuối cùng của hình ảnh hiện tại.

3. Phương pháp giải mã ảnh theo điểm 1, trong đó thông tin chiều cao của lát hiện tại không được thu nhận từ luồng bit và được xác định là giá trị được xác định trước dựa trên việc ô xếp trên cùng bên trái của lát hiện tại thuộc về hàng ô xếp cuối cùng của hình ảnh hiện tại.

4. Phương pháp giải mã ảnh theo điểm 3, trong đó giá trị được xác định trước đã nêu chỉ ra một hàng ô xếp.

5. Phương pháp giải mã ảnh theo điểm 3, trong đó lát hiện tại là lát hình chữ nhật.

6. Phương pháp giải mã ảnh theo điểm 1,

trong đó bước thu nhận thông tin kích thước được thực hiện dựa trên số lượng lát phân vùng hình ảnh hiện tại,

trong đó số lượng lát phân vùng hình ảnh hiện tại được xác định bằng cách:

xác định tính khả dụng của việc phân vùng hình ảnh hiện tại;

xác định số lượng hàng ô xếp và số lượng cột ô xếp, mà chúng đều phân vùng hình ảnh hiện tại, dựa trên tính khả dụng của việc phân vùng hình ảnh hiện tại;

xác định chiều rộng của mỗi cột ô xếp phân vùng hình ảnh hiện tại dựa trên số lượng cột ô xếp;

xác định chiều cao của mỗi hàng ô xếp phân vùng hình ảnh hiện tại dựa trên số lượng hàng ô xếp;

xác định xem liệu hình ảnh hiện tại có được phân vùng thành các lát hình chữ nhật hay không dựa trên số lượng ô xếp phân vùng hình ảnh hiện tại; và

thu nhận số lượng lát phân vùng hình ảnh hiện tại từ luồng bit dựa trên việc liệu hình ảnh hiện tại có được phân vùng thành các lát hình chữ nhật hay không.

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

xác định lát hiện tại tương ứng với ít nhất một phần của hình ảnh hiện tại; và

tạo ra luồng bit bao gồm thông tin kích thước của lát hiện tại,

trong đó thông tin kích thước này bao gồm thông tin chiều rộng chỉ ra chiều rộng của lát hiện tại theo các đơn vị là các cột ô xếp và thông tin chiều cao chỉ ra chiều cao của lát hiện tại theo các đơn vị là các hàng ô xếp, và

trong đó bước tạo ra luồng bit được thực hiện dựa trên việc liệu lát hiện tại thuộc về cột ô xếp cuối cùng hay hàng ô xếp cuối cùng của hình ảnh hiện tại,

trong đó thông tin chiều cao của lát hiện tại không được mã hóa trong luồng bit dựa trên việc ô xếp trên cùng bên trái của lát hiện tại thuộc về hàng ô xếp cuối cùng của hình ảnh hiện tại.

8. Phương pháp mã hóa ảnh theo điểm 7, trong đó lát hiện tại là lát hình chữ nhật.

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

xác định lát hiện tại tương ứng với ít nhất một phần của hình ảnh hiện tại; và
 tạo ra luồng bit bao gồm thông tin kích thước của lát hiện tại,
 trong đó thông tin kích thước này bao gồm thông tin chiều rộng chỉ ra chiều rộng của lát
 hiện tại theo các đơn vị là các cột ô xếp và thông tin chiều cao chỉ ra chiều cao của lát hiện tại
 theo các đơn vị là các hàng ô xếp, và

trong đó bước tạo ra luồng bit được thực hiện dựa trên việc liệu lát hiện tại thuộc về cột ô
 xếp cuối cùng hay hàng ô xếp cuối cùng của hình ảnh hiện tại,

trong đó thông tin chiều cao của lát hiện tại không được mã hóa trong luồng bit dựa trên
 việc ô xếp trên cùng bên trái của lát hiện tại thuộc về hàng ô xếp cuối cùng của hình ảnh hiện tại.

10. Phương tiện phi chuyển tiếp đọc được bằng máy tính lưu trữ luồng bit được tạo ra bởi phương
 pháp mã hóa ảnh, phương pháp mã hóa ảnh này bao gồm các bước:

xác định lát hiện tại tương ứng với ít nhất một phần của hình ảnh hiện tại; và
 tạo ra luồng bit bao gồm thông tin kích thước của lát hiện tại,
 trong đó thông tin kích thước này bao gồm thông tin chiều rộng chỉ ra chiều rộng của lát
 hiện tại theo các đơn vị là các cột ô xếp và thông tin chiều cao chỉ ra chiều cao của lát hiện tại
 theo các đơn vị là các hàng ô xếp, và

trong đó bước tạo ra luồng bit được thực hiện dựa trên việc liệu lát hiện tại thuộc về cột ô
 xếp cuối cùng hay hàng ô xếp cuối cùng của hình ảnh hiện tại,

trong đó thông tin chiều cao của lát hiện tại không được mã hóa trong luồng bit dựa trên
 việc ô xếp trên cùng bên trái của lát hiện tại thuộc về hàng ô xếp cuối cùng của hình ảnh hiện tại.

FIG.1

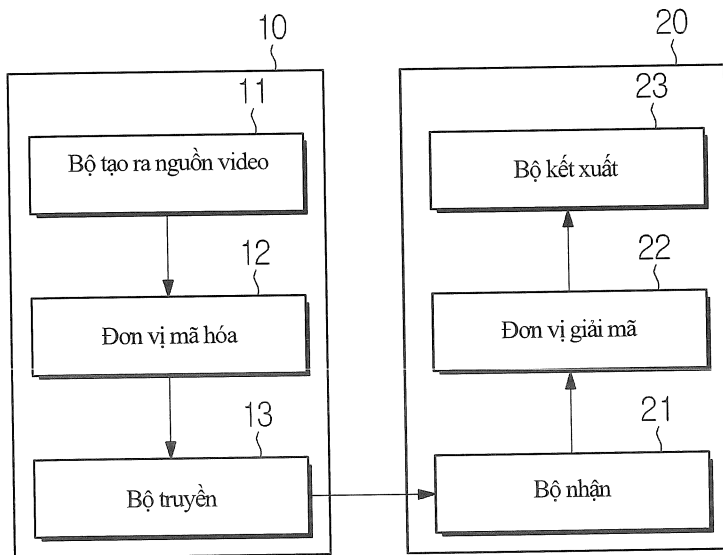


FIG.2

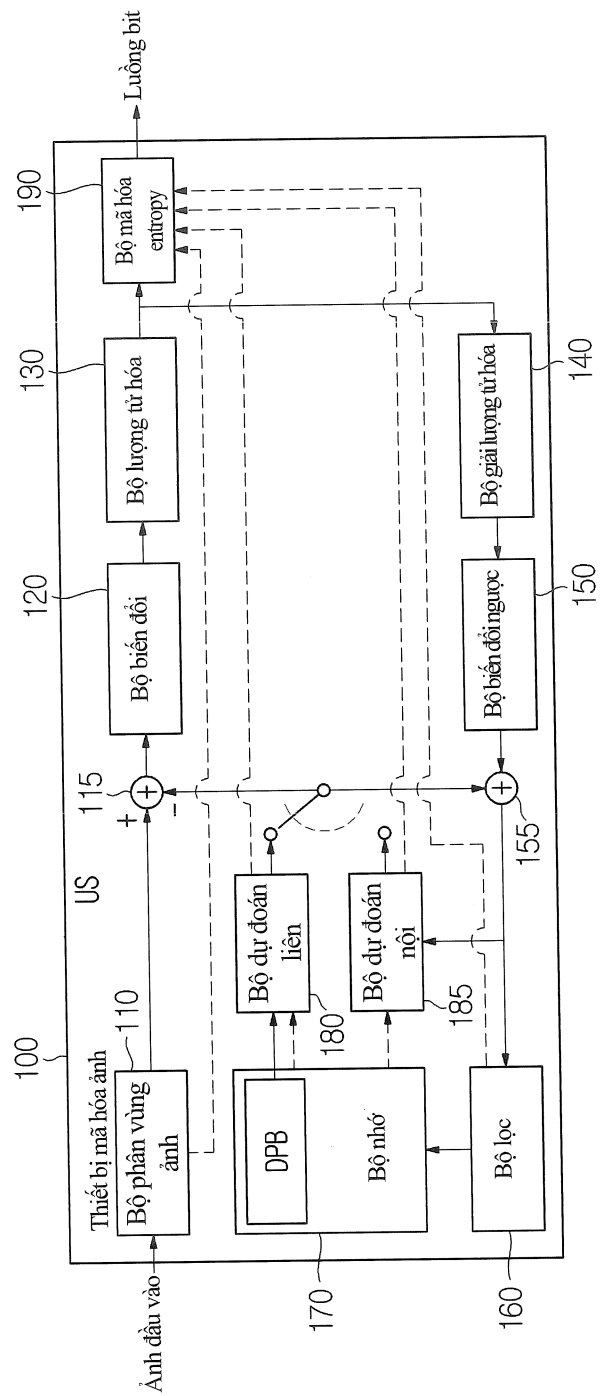


FIG.3

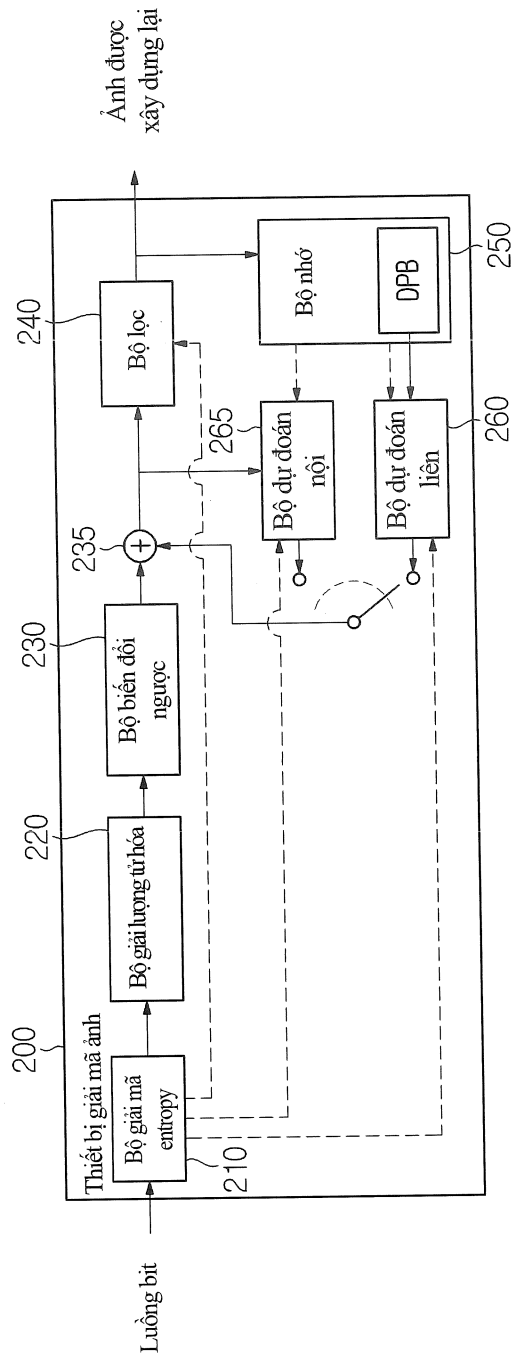
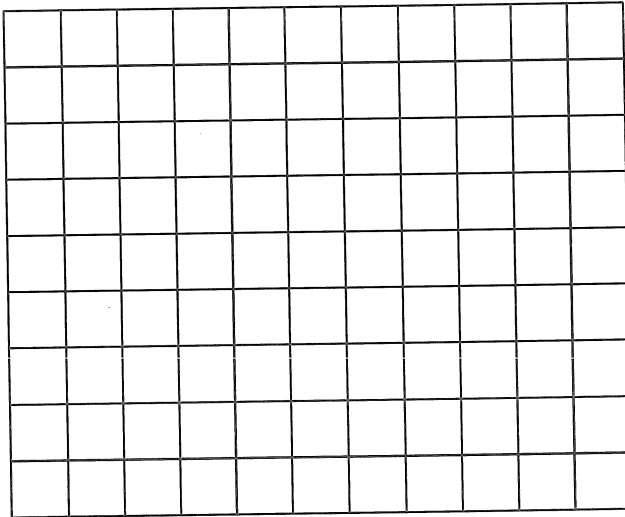
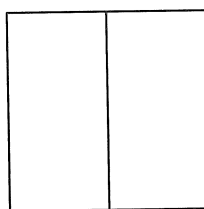
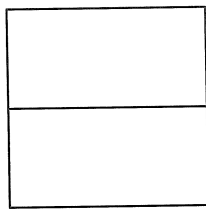
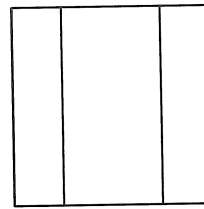


FIG.4**FIG.5**

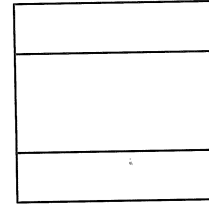
SPLIT_BT_VER



SPLIT_BT_HOR



SPLIT_TT_VER



SPLIT_TT_HOR

FIG.6

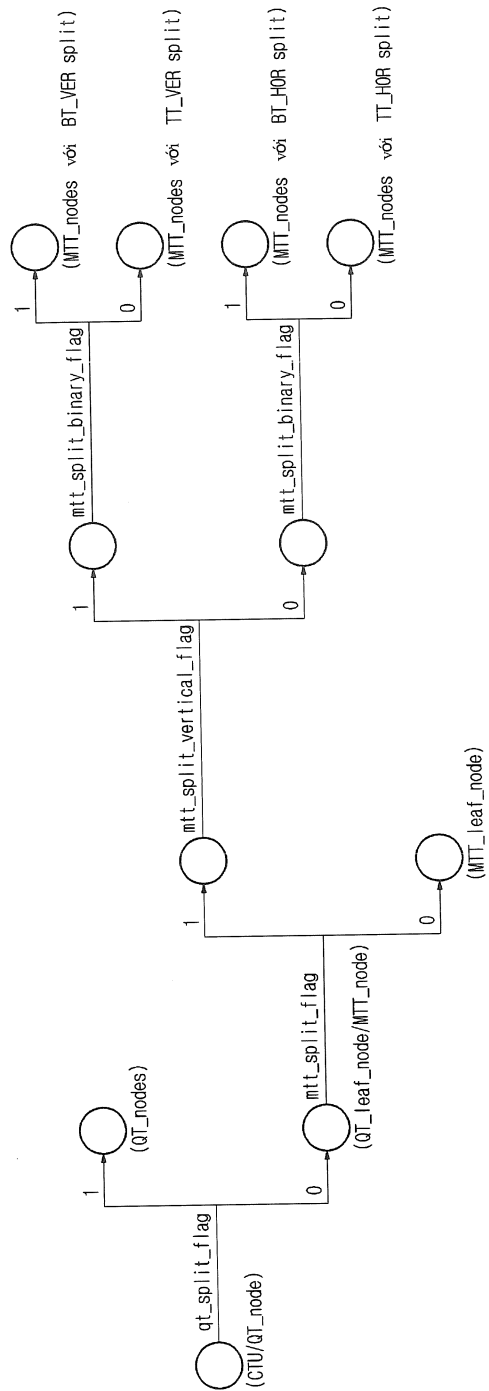


FIG.7

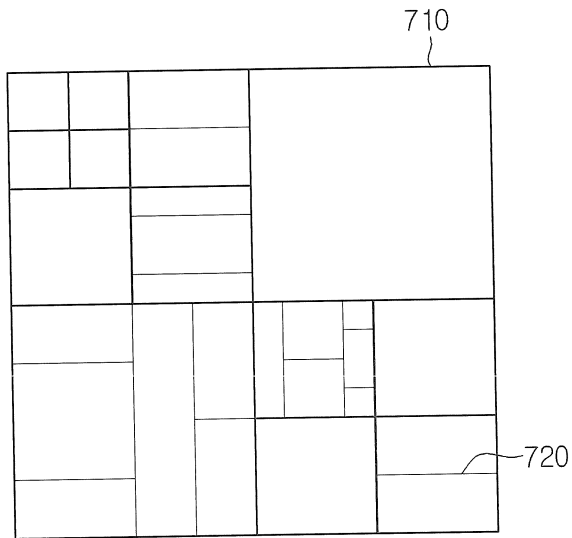


FIG.8

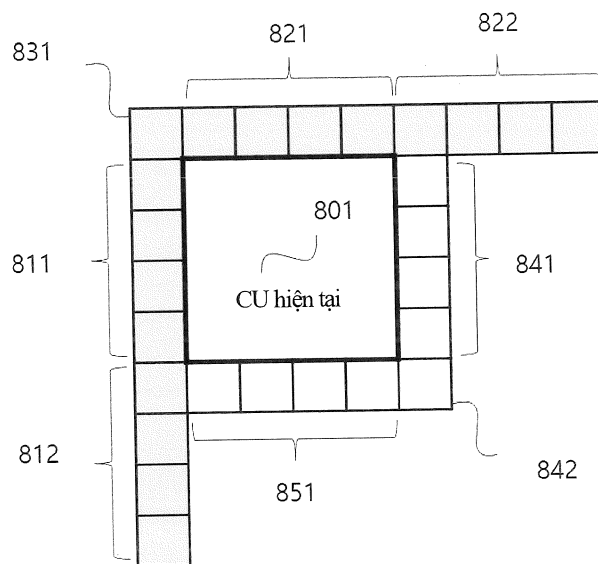


FIG.9

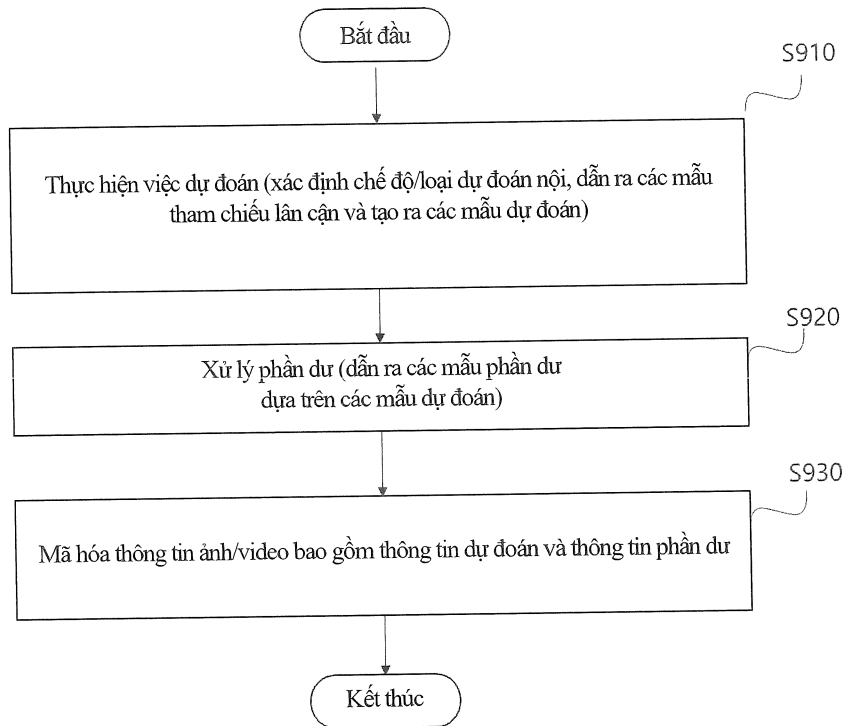


FIG.10

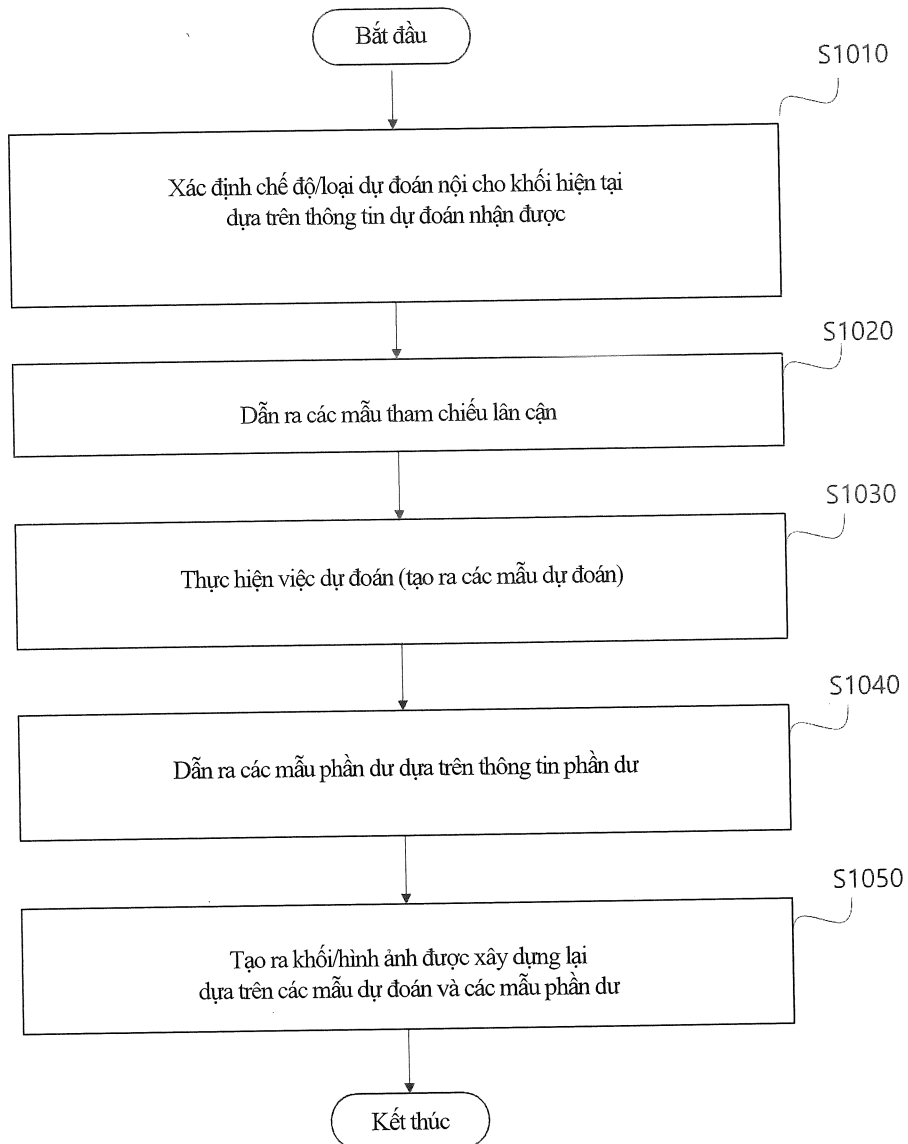


FIG.11

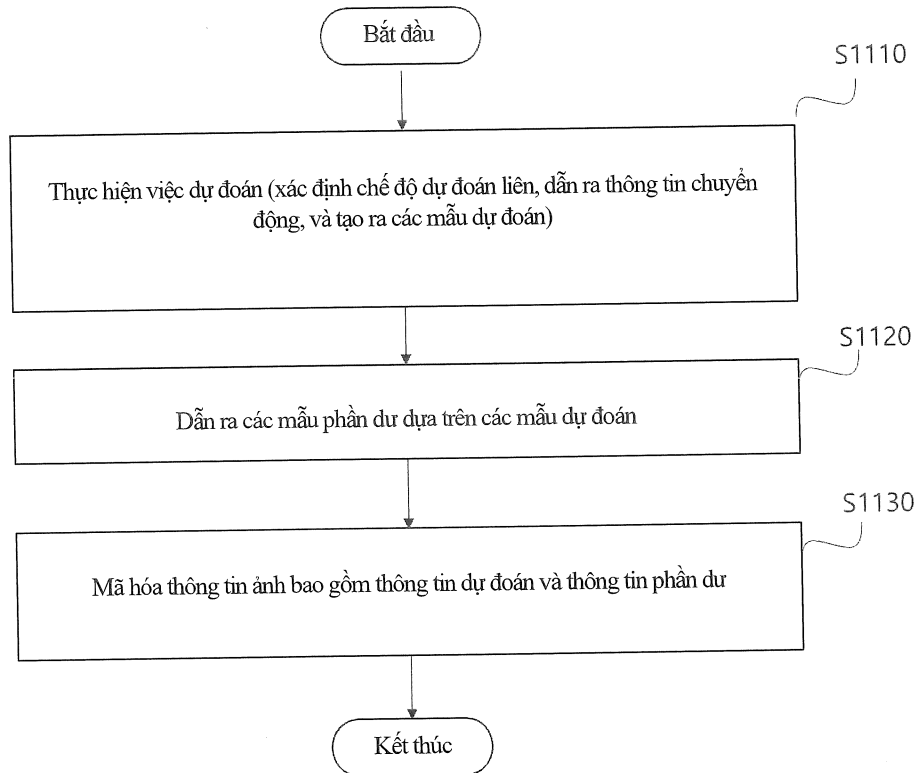


FIG.12

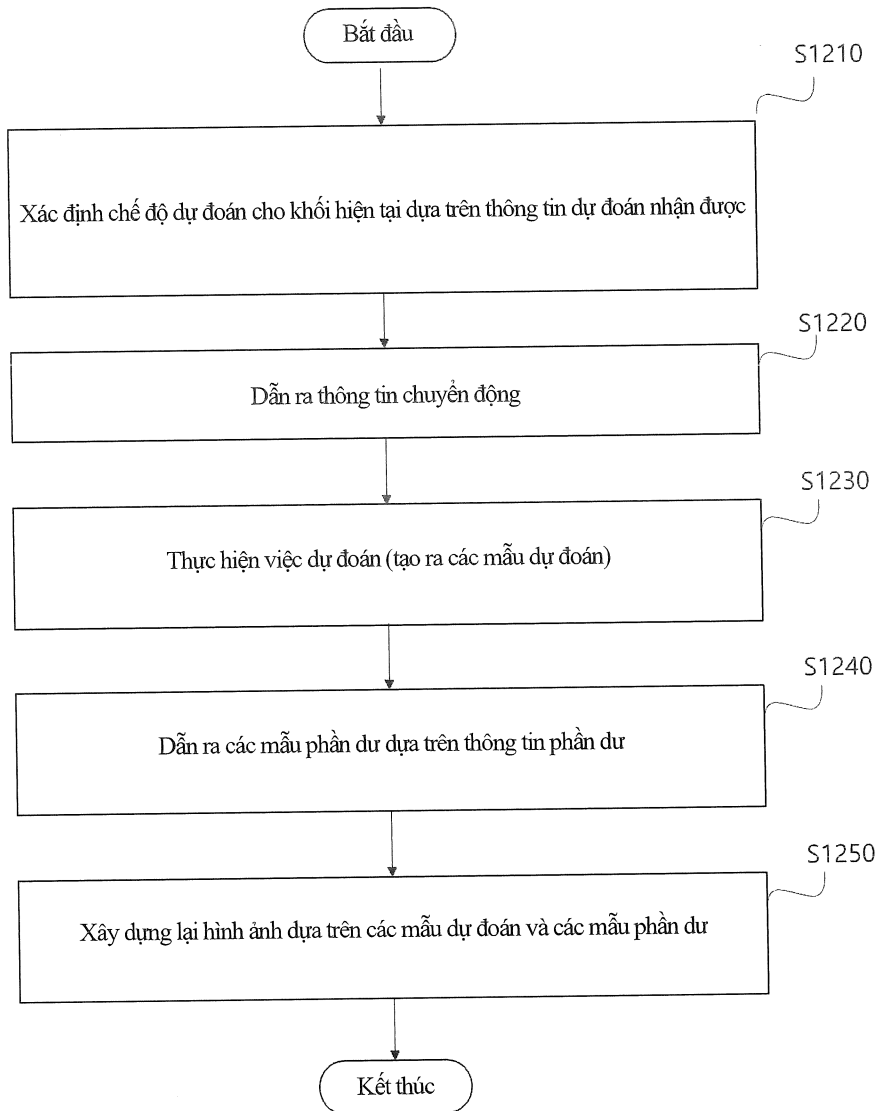


FIG.13

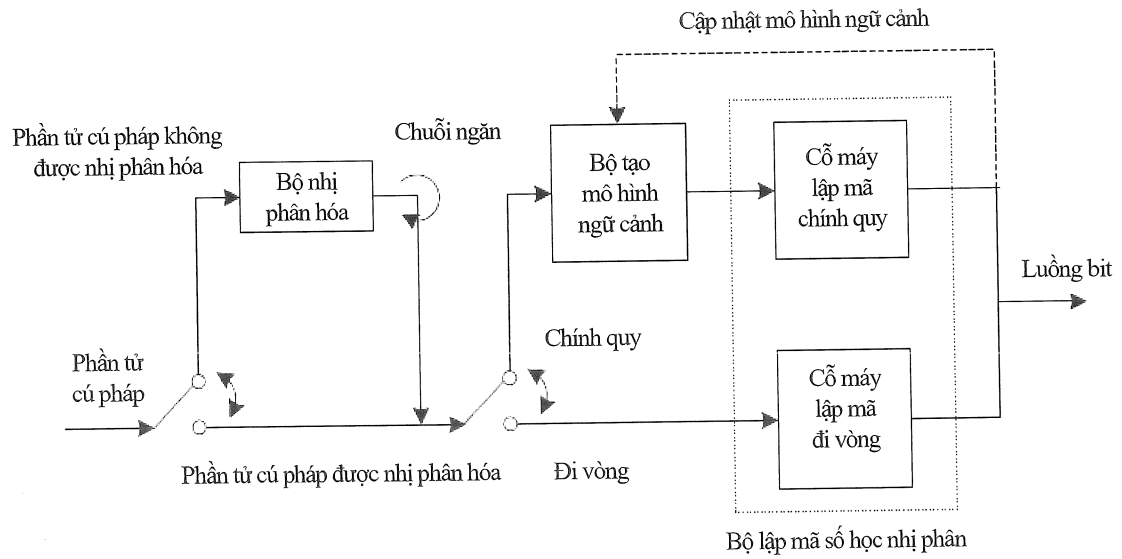


FIG.14

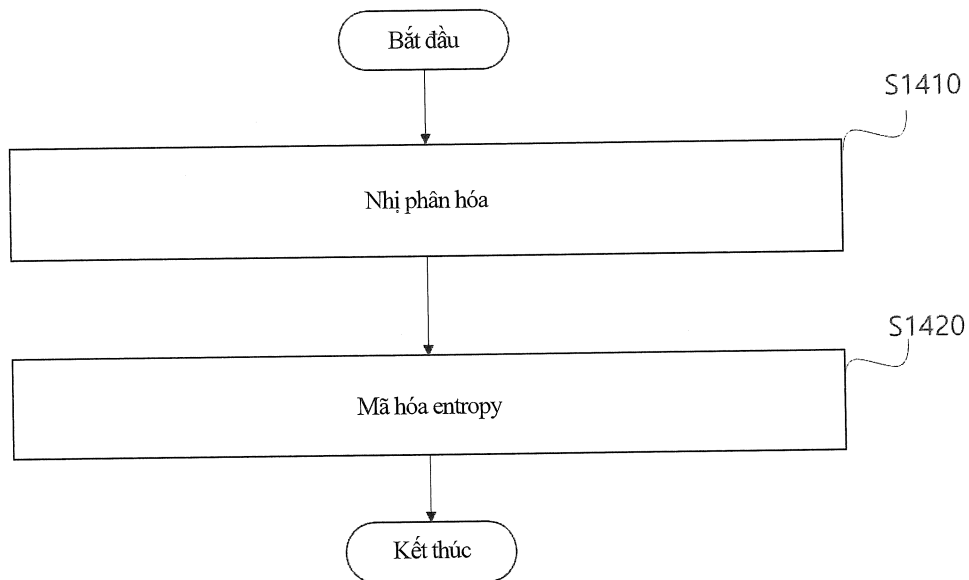


FIG.15

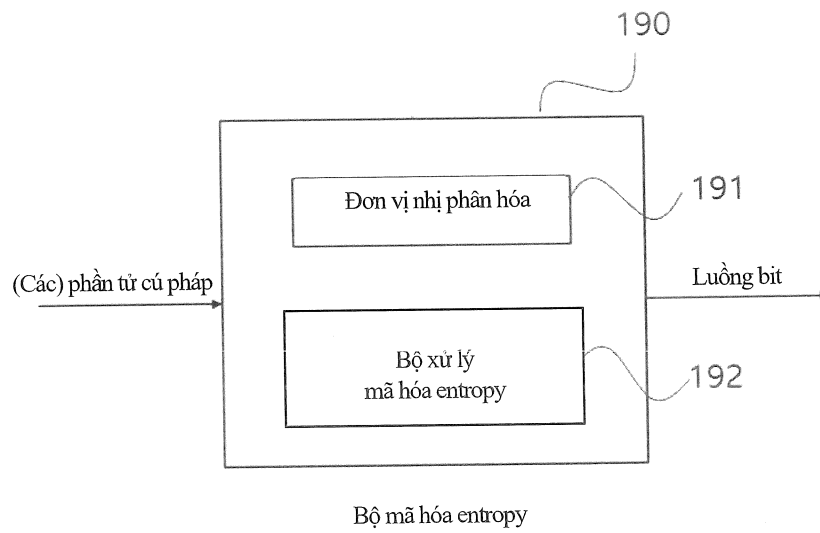


FIG.16

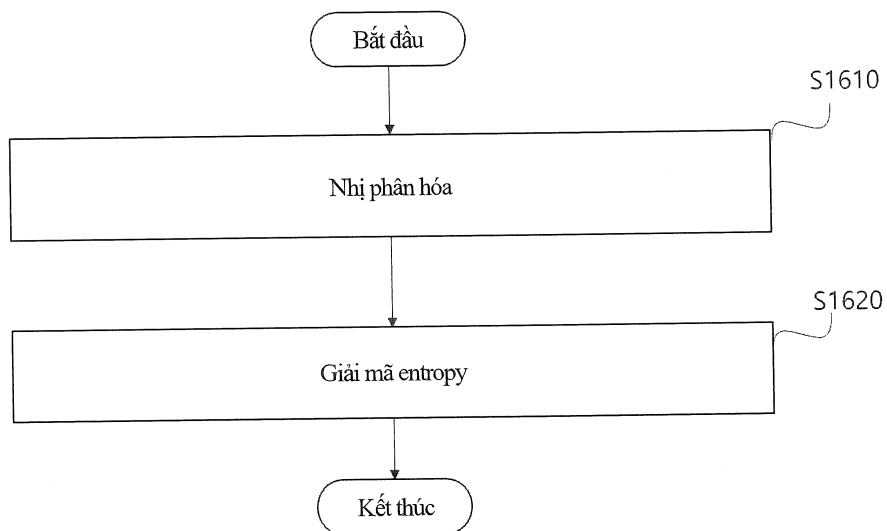


FIG.17

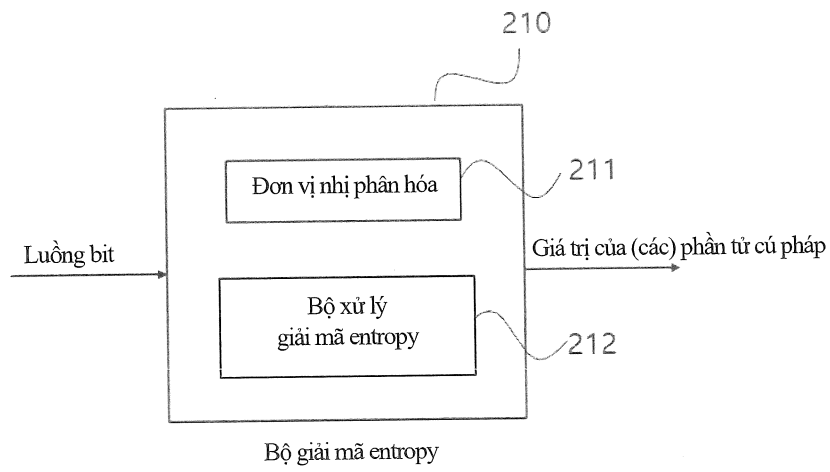


FIG.18

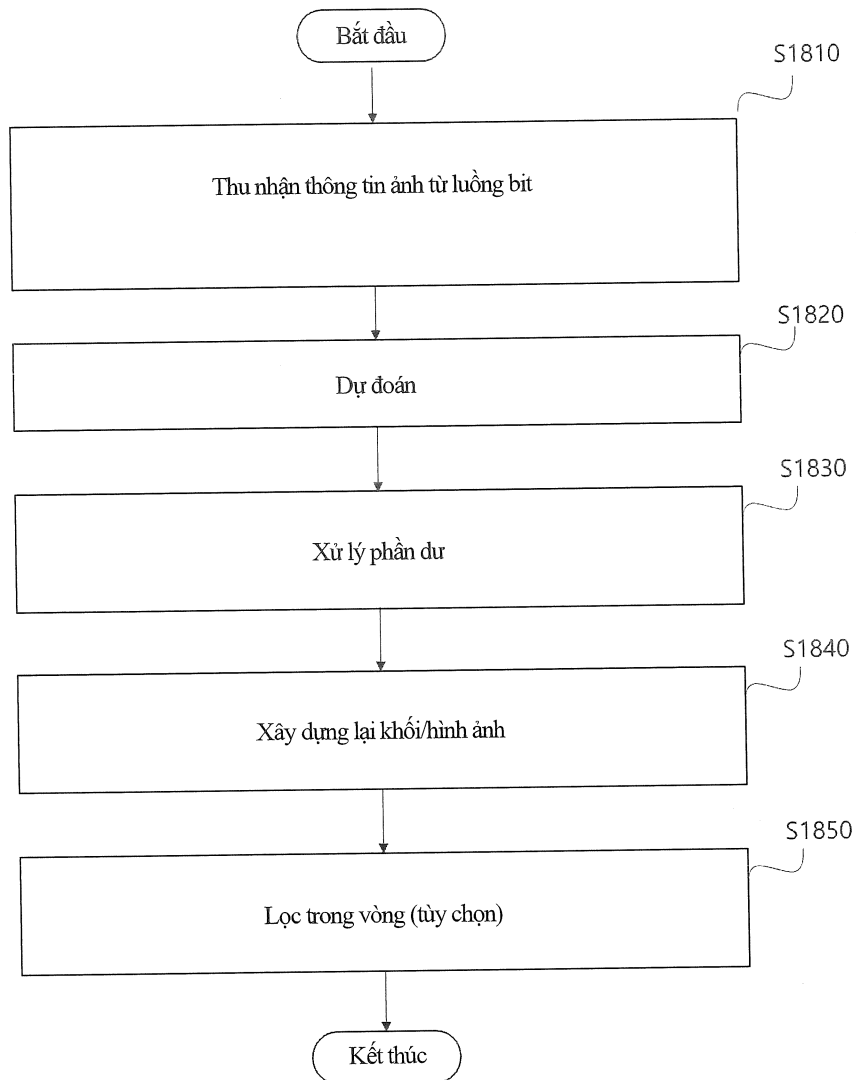


FIG.19

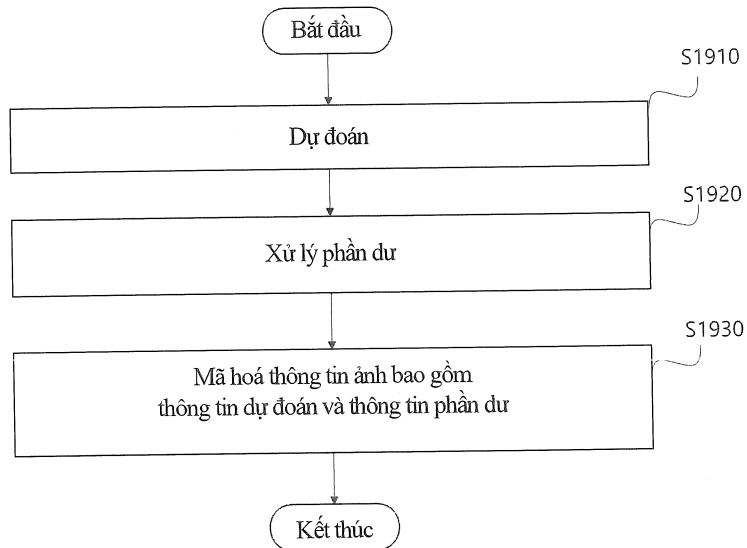


FIG.20

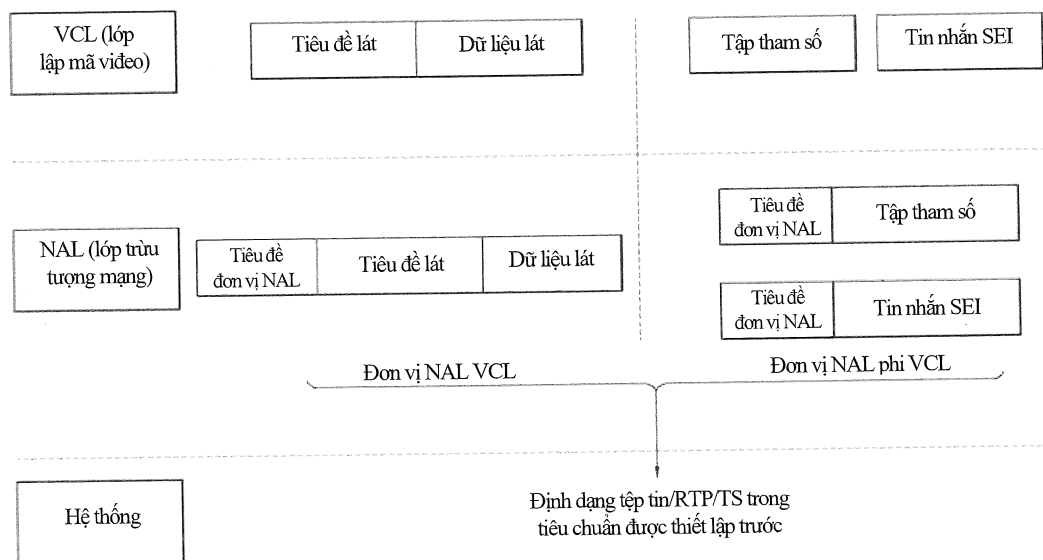
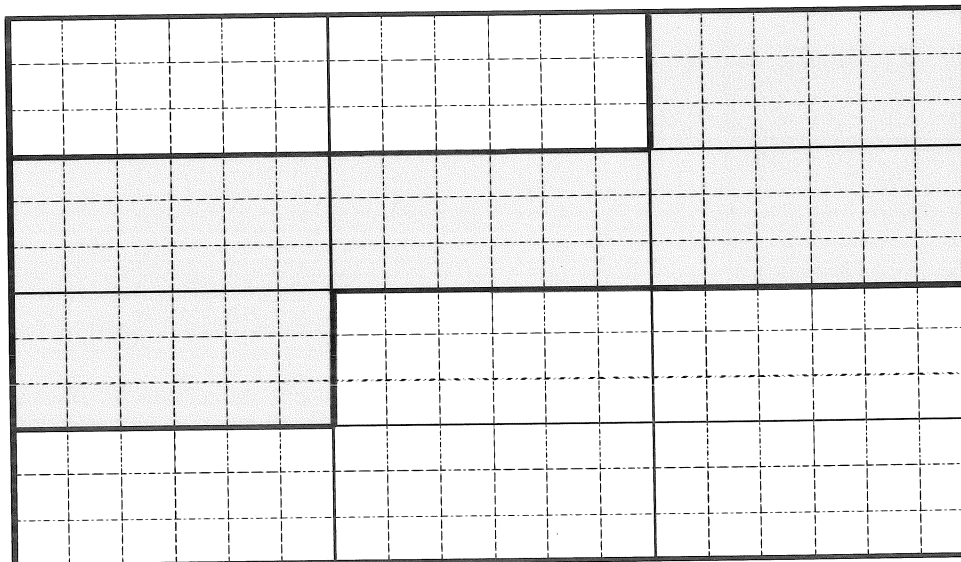


FIG.21

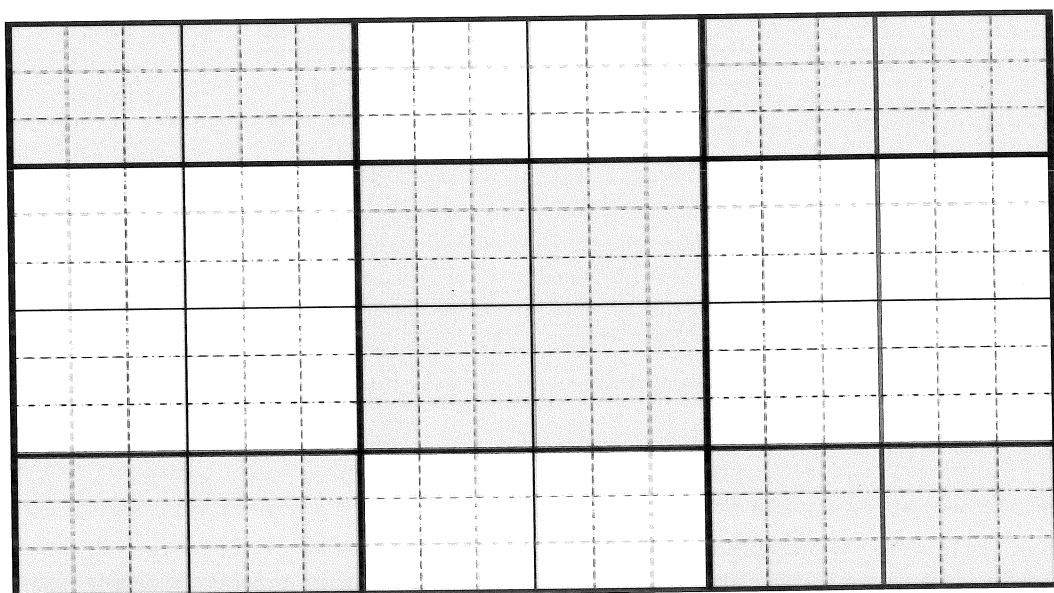


CTU

Ô xếp

Lát

FIG. 22



CTU

Ô xếp

Lát

FIG.23

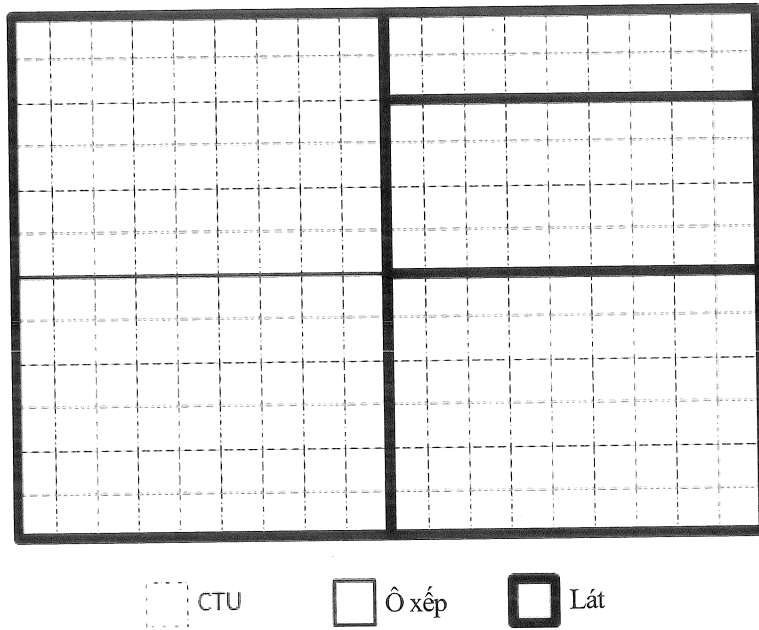


FIG.24

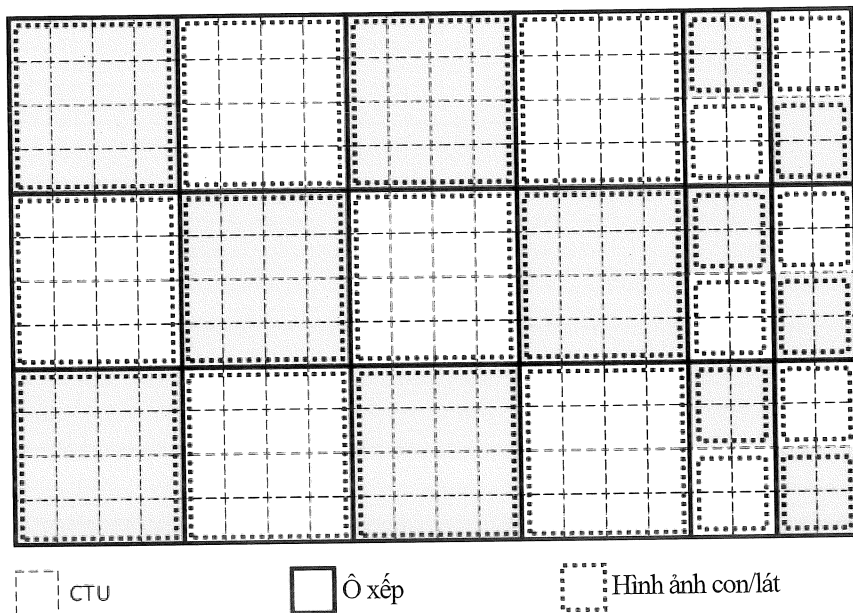


FIG.25

	Descriptor
seq_parameter_set_rbsp() {	
...	
subpic_info_present_flag	u(1)
if(subpic_info_present_flag) {	
sps_num_subpics_minus1	ue(v)
sps_independent_subpics_flag	u(1)
for(i = 0; sps_num_subpics_minus1 > 0 && i <= sps_num_subpics_minus1; i++) {	
if(i > 0 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_ctu_top_left_x[i]	u(v)
if(i > 0 && pic_height_max_in_luma_samples > CtbSizeY) {	
subpic_ctu_top_left_y[i]	u(v)
if(i < sps_num_subpics_minus1 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_width_minus1[i]	u(v)
if(i < sps_num_subpics_minus1 && pic_height_max_in_luma_samples > CtbSizeY)	
subpic_height_minus1[i]	u(v)
if(!sps_independent_subpics_flag) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
}	
...	
}	

FIG.26

	Descriptor
pic parameter set rbsp() {	
...	
no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	
pps_log2_ctu_size_minus5	u(2)
num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
if(NumTilesInPic > 1)	
rect_slice_flag	u(1)
if(rect_slice_flag)	
single_slice_per_subpic_flag	u(1)
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
if(num_slices_in_pic_minus1 > 0)	
tile_idx_delta_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
if(NumTileColumns > 1)	
slice_width_in_tiles_minus1[i]	ue(v)
if(NumTileRows > 1 && (tile_idx_delta_present_flag	
SliceTopLeftTileIdx[i] % NumTileColumns == 0))	
slice_height_in_tiles_minus1[i]	ue(v)
if(slice_width_in_tiles_minus1[i] == 0 &&	
slice_height_in_tiles_minus1[i] == 0 &&	
RowHeight[SliceTopLeftTileIdx[i] / NumTileColumns] > 1) {	
num_exp_slices_in_tile[i]	ue(v)
for(j = 0; j < num_exp_slices_in_tile[i]; j++)	
exp_slice_height_in_ctus_minus1[i][j]	ue(v)
i += NumSlicesInTile[i] - 1	
}	
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1)	
tile_idx_delta[i]	se(v)
}	
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
}	
...	
}	

FIG.27

slice_header() {	Descripto r
...	
if(subpic_info_present_flag)	
slice_subpic_id	u(v)
if((rect_slice_flag && NumSlicesInSubpic[CurrSubpicIdx] > 1) (!rect_slice_flag && NumTilesInPic > 1))	
slice_address	u(v)
for(i = 0; i < NumExtraShBits; i++)	
sh_extra_bit[i]	u(1)
if(!rect_slice_flag && NumTilesInPic > 1)	
num_tiles_in_slice_minus1	ue(v)
...	
}	

FIG.28

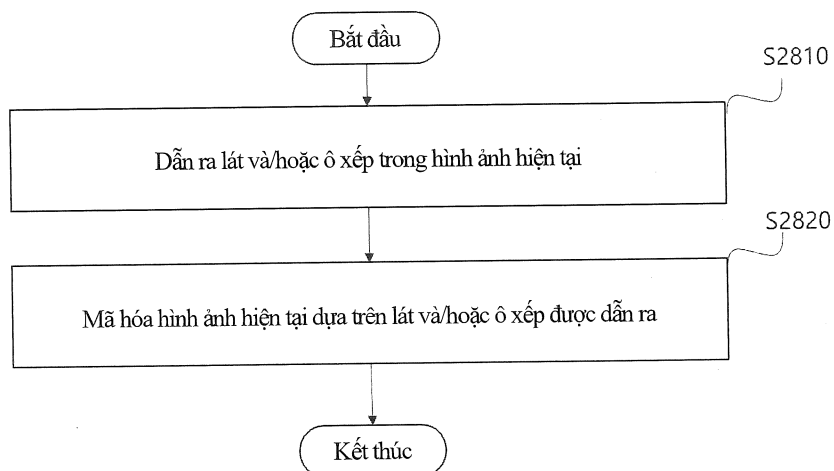


FIG.29

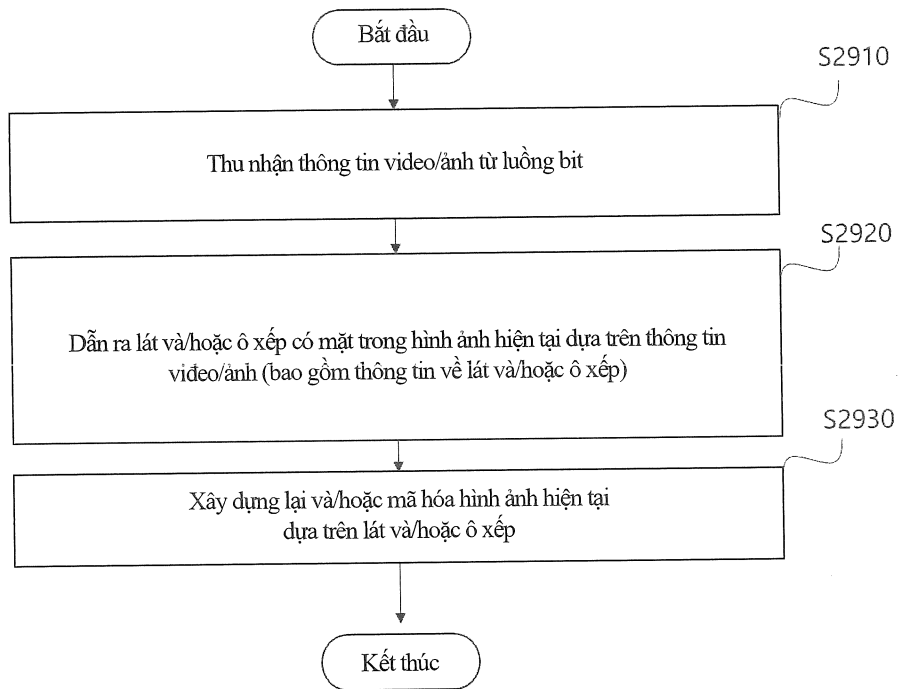


FIG.30

pic_parameter_set_rbsp() {	Descriptor
...	
no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	
...	
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
if(num_slices_in_pic_minus1 > 0)	
tile_idx_delta_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
if(NumTileColumns > 1 && (SliceTopLeftTileIdx[i] % NumTileColumns != NumTileColumns - 1))	
slice_width_in_tiles_minus1[i]	ue(v)
if(NumTileRows > 1 &&(tile_idx_delta_present_flag SliceTopLeftTileIdx[i] % NumTileColumns == 0) && (SliceTopLeftTileIdx[i] / NumTileColumns != NumTileRows - 1))	
slice_height_in_tiles_minus1[i]	ue(v)
...	
}	
}	
...	
}	
...	
}	

FIG.31

pic_parameter_set_rbsp() {	Descript or
...	
no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	
...	
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
if(num_slices_in_pic_minus1 > 0)	
tile_idx_delta_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
if(SliceTopLeftTileIdx[i] % NumTileColumns != NumTileColumns - 1)	
slice_width_in_tiles_minus1[i]	ue(v)
if((SliceTopLeftTileIdx[i] / NumTileColumns != NumTileRows - 1) &&(tile_idx_delta_present_flag SliceTopLeftTileIdx[i] % NumTileColumns == 0))	
slice_height_in_tiles_minus1[i]	ue(v)
...	
}	
}	
...	
}	
...	
}	

FIG.32

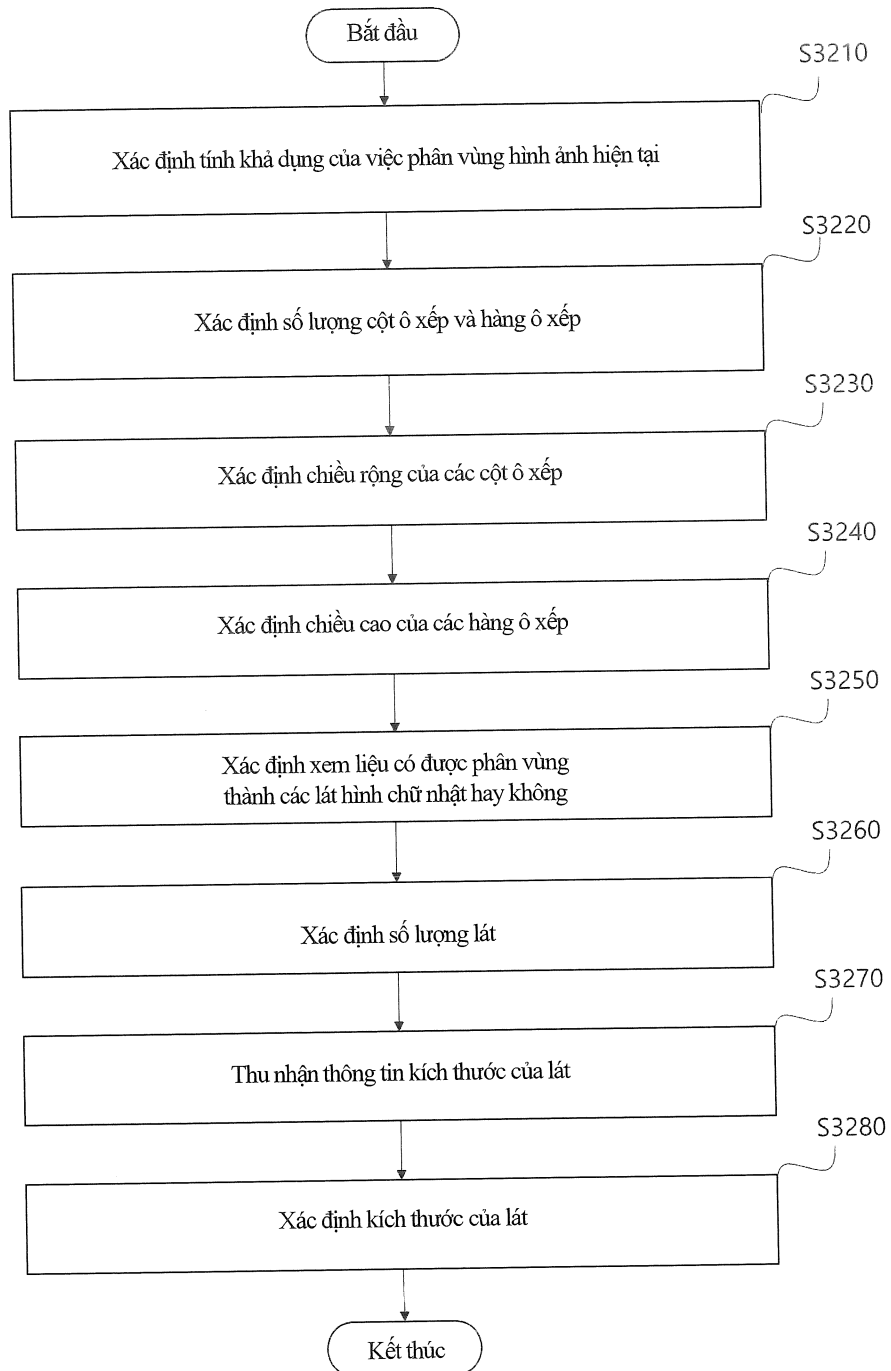


FIG.33

```

tileIdx = 0
for( i = 0; i <= pps_num_slices_in_pic_minus1; i++ )
    NumCtusInSlice[ i ] = 0
for( i = 0; i <= pps_num_slices_in_pic_minus1; i++ ) {
    SliceTopLeftTileIdx[ i ] = tileIdx
    tileX = tileIdx % NumTileColumns
    tileY = tileIdx / NumTileColumns
    if( i < pps_num_slices_in_pic_minus1 ) {
        sliceWidthInTiles[ i ] = pps_slice_width_in_tiles_minus1[ i ] + 1
        sliceHeightInTiles[ i ] = pps_slice_height_in_tiles_minus1[ i ] + 1
    } else {
        sliceWidthInTiles[ i ] = NumTileColumns - tileX
        sliceHeightInTiles[ i ] = NumTileRows - tileY
        NumSlicesInTile[ i ] = 1
    }
    if( sliceWidthInTiles[ i ] == 1 && sliceHeightInTiles[ i ] == 1 ) {
        if( pps_num_exp_slices_in_tile[ i ] == 0 ) {
            NumSlicesInTile[ i ] = 1
            sliceHeightInCtus[ i ] = RowHeightVal[ SliceTopLeftTileIdx[ i ] / NumTileColumns ]
        } else {
            remainingHeightInCtbsY = RowHeightVal[ SliceTopLeftTileIdx[ i ] / NumTileColumns ]
            for( j = 0; j < pps_num_exp_slices_in_tile[ i ]; j++ ) {
                sliceHeightInCtus[ i + j ] = pps_exp_slice_height_in_ctus_minus1[ i ][ j ] + 1
                remainingHeightInCtbsY -= sliceHeightInCtus[ i + j ]
            }
            uniformSliceHeight = sliceHeightInCtus[ i + j - 1 ]
            while( remainingHeightInCtbsY >= uniformSliceHeight ) {
                sliceHeightInCtus[ i + j ] = uniformSliceHeight
                remainingHeightInCtbsY -= uniformSliceHeight
                j++
            }
            if( remainingHeightInCtbsY > 0 ) {
                sliceHeightInCtus[ i + j ] = remainingHeightInCtbsY
                j++
            }
        }
    }
}

```

FIG.34

```

        NumSlicesInTile[ i ] = j
    }
    ctbY = TileRowBdVal[ tileY ]
    for( j = 0; j < NumSlicesInTile[ i ]; j++ ) {
        AddCtbsToSlice( i + j, TileColBdVal[ tileX ], TileColBdVal[ tileX + 1 ],
            ctbY, ctbY + sliceHeightInCtus[ i + j ] )
        ctbY += sliceHeightInCtus[ i + j ]
        sliceWidthInTiles[ i + j ] = 1
        sliceHeightInTiles[ i + j ] = 1
    }
    i += NumSlicesInTile[ i ] - 1
} else
    for( j = 0; j < sliceHeightInTiles[ i ]; j++ )
        for( k = 0; k < sliceWidthInTiles[ i ]; k++ )
            AddCtbsToSlice( i, TileColBdVal[ tileX + k ], TileColBdVal[ tileX + k + 1 ],
                TileRowBdVal[ tileY + j ], TileRowBdVal[ tileY + j + 1 ] )
if( i < pps_num_slices_in_pic_minus1 ) {
    if( pps_tile_idx_delta_present_flag )
        tileIdx += pps_tile_idx_delta_val[ i ]
    else {
        tileIdx += sliceWidthInTiles[ i ]
        if( tileIdx % NumTileColumns == 0 )
            tileIdx += ( sliceHeightInTiles[ i ] - 1 ) * NumTileColumns
    }
}
}
}

```

FIG.35

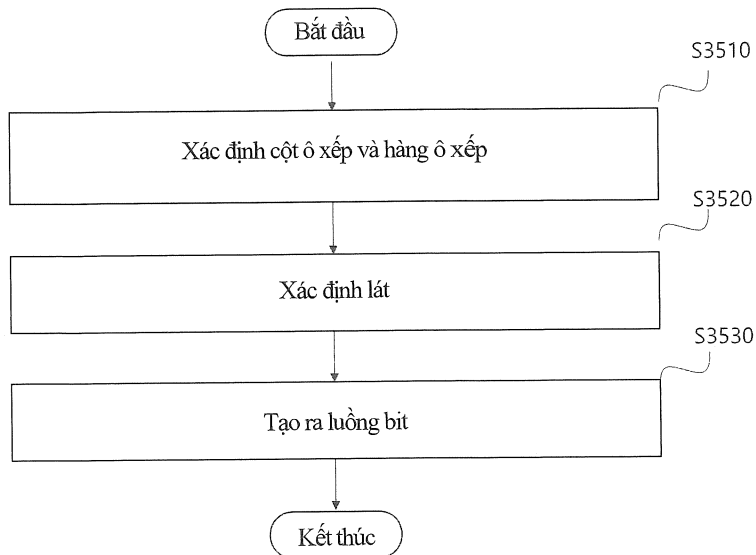


FIG.36

