



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

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN) (11)
CỤC SỞ HỮU TRÍ TUỆ



1-0044713

(51)^{2020.01} H04N 19/29; H04N 19/17; H04N 19/174 (13) B

(21) 1-2021-04701

(22) 09/01/2020

(86) PCT/US2020/012970 09/01/2020

(87) WO2020/146662 16/07/2020

(30) 62/790,207 09/01/2019 US

(45) 25/04/2025 445

(43) 25/11/2021 404A

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

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

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

(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP GIẢI MÃ, PHƯƠNG PHÁP MÃ HÓA, VẬT GHI ĐỌC ĐƯỢC
BẰNG MÁY TÍNH, BỘ GIẢI MÃ, BỘ MÃ HÓA, VÀ PHƯƠNG TIỆN LƯU TRỮ

(21) 1-2021-04701

(57) Sáng chế đề cập đến cơ chế tạo mã video. Cơ chế bao gồm việc thu luồng bit bao gồm tập hợp thông số chuỗi (SPS), một hoặc nhiều ảnh con được phân chia từ ảnh, và một hoặc nhiều phần đầu lát được liên kết với một hoặc nhiều lát. SPS được phân tích để nhận được các bộ nhận dạng (các ID) ảnh con cho một hoặc nhiều ảnh con. Các phần đầu lát được phân tích để nhận được ID ảnh con hiện tại được liên kết với ảnh con hiện tại. ID ảnh con hiện tại chỉ thị các lát được bao gồm trong ảnh con hiện tại của một hoặc nhiều ảnh con. Ảnh con hiện tại được giải mã dựa vào ID ảnh con hiện tại để tạo chuỗi video. Chuỗi video được chuyển tiếp cho việc hiển thị.

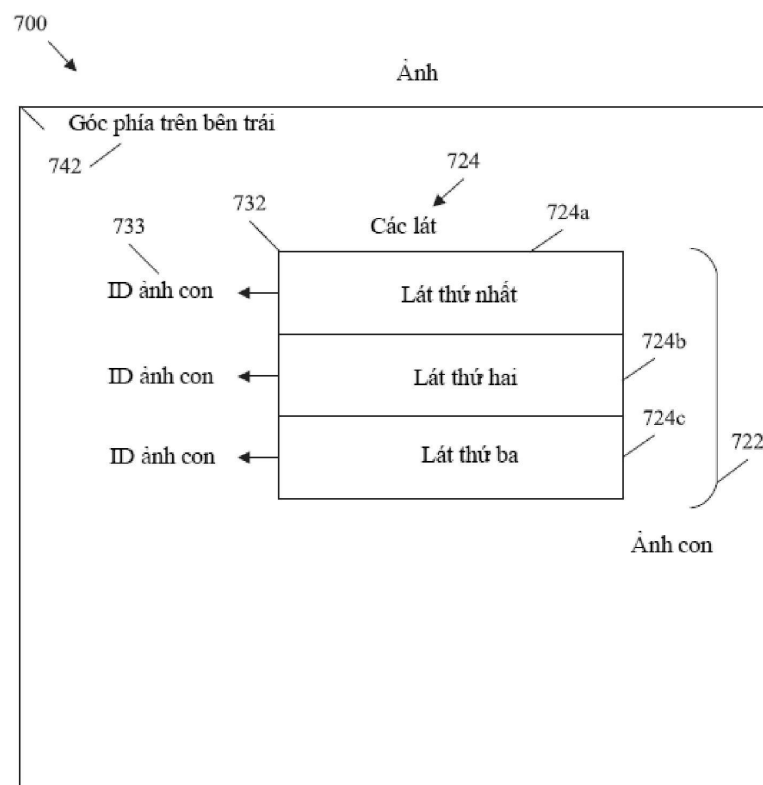


FIG. 7

Lĩnh vực kỹ thuật được đề cập

Sáng chế này đề cập chung đến việc tạo mã video, và cụ thể hơn là đề cập đến việc quản lý ảnh con trong việc tạo mã video.

Tình trạng kỹ thuật của sáng chế

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

Bản chất kỹ thuật của sáng chế

Theo một phương án, sáng chế bao gồm phương pháp được thực hiện ở bộ giải mã, phương pháp này bao gồm các bước: thu, bởi bộ thu của bộ giải mã, luồng bit bao gồm tập hợp thông số chuỗi (sequence parameter set - SPS), một hoặc nhiều ảnh con được phân chia từ ảnh, và một hoặc nhiều phần đầu lát (slice) được liên kết với một hoặc nhiều lát; phân tích, bởi bộ xử lý của bộ giải mã, SPS để nhận được các bộ nhận dạng (identifier - ID) ảnh con cho một hoặc nhiều ảnh con; phân tích, bởi bộ xử lý, các phần đầu lát để nhận được ID ảnh con hiện tại được liên kết với ảnh con hiện tại, ID ảnh con hiện tại chỉ thị các lát được bao gồm trong ảnh con hiện tại từ một hoặc nhiều ảnh con; giải mã, bởi bộ xử lý, ảnh con hiện tại dựa vào ID ảnh con hiện tại để tạo chuỗi video; và chuyển tiếp, bởi bộ xử lý, chuỗi video cho việc hiển thị. Một số hệ thống tạo mã video báo hiệu các vị trí ảnh con so với các ảnh con khác. Điều này gây ra vấn đề nếu các ảnh con bị mất trong chuyển tiếp hoặc được

phát riêng biệt cho mục đích được trích xuất riêng biệt. Trong ví dụ này, tập hợp đầy đủ của các ID ảnh con được báo hiệu trong SPS. Ngoài ra, các phần đầu lát bao gồm ID ảnh con chỉ thị ảnh con mà chứa các lát tương ứng với các phần đầu lát. Bằng việc thiết kế mỗi ảnh con theo ID, các ảnh con có thể được định vị và được tạo kích thước mà không tham chiếu tới các ảnh con khác. Điều này đến lượt nó hỗ trợ việc hiệu chỉnh lỗi cũng như các ứng dụng mà chỉ trích xuất một số ảnh con và tránh việc phát các ảnh con khác. Danh sách đầy đủ của tất cả các ID ảnh con có thể được gửi trong SPS cùng với thông tin kích thước tương ứng. Mỗi phần đầu lát có thể chứa ID ảnh con chỉ thị ảnh con mà bao gồm lát tương ứng. Theo cách này, các ảnh con và các lát tương ứng có thể được trích xuất và được định vị mà không tham chiếu tới các ảnh con khác. Do đó, các ID ảnh con hỗ trợ chức năng được nâng cao và/hoặc hiệu suất mã hóa tăng, mà làm giảm việc sử dụng của các tài nguyên mạng, các tài nguyên bộ nhớ, và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó SPS còn bao gồm kích thước ảnh con cho mỗi ảnh con được phân chia từ ảnh.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó SPS còn bao gồm vị trí ảnh con cho mỗi ảnh con được phân chia từ ảnh.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó bước giải mã ảnh con hiện tại bao gồm: so khớp các lát với ảnh con hiện tại dựa vào ID ảnh con hiện tại; và định vị các lát dựa vào kích thước ảnh con và vị trí ảnh con đối với ảnh con hiện tại.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó vị trí ảnh con bao gồm khoảng cách dịch vị giữa mẫu phía trên cùng bên trái của ảnh con hiện tại và mẫu phía trên cùng bên trái của ảnh.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó kích thước ảnh con bao gồm chiều cao ảnh con trong các mẫu độ chói và chiều rộng ảnh con trong các mẫu độ chói.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó SPS còn bao gồm việc chỉ thị rằng một hoặc nhiều ảnh con là các ảnh con được ràng buộc chuyển động.

Theo một phương án, sáng chế bao gồm phương pháp được thực hiện ở bộ mã hóa, phương pháp này bao gồm các bước: phân chia, bởi bộ xử lý của bộ mã hóa, ảnh thành nhiều ảnh con; phân chia, bởi bộ xử lý, mỗi ảnh con trong số nhiều ảnh con thành một hoặc nhiều lát; mã hóa thành luồng bit, bởi bộ xử lý, SPS bao gồm các ID ảnh con cho nhiều ảnh con được phân chia từ ảnh; mã hóa thành luồng bit, bởi bộ xử lý, các phần đầu lát bao gồm các ID ảnh con chỉ thị các lát được bao gồm trong các ảnh con được chỉ thị; mã hóa, bởi bộ xử lý, một hoặc nhiều ảnh con trong số nhiều ảnh con vào luồng bit; và lưu trữ, trong bộ nhớ của bộ mã hóa, luồng bit cho việc truyền thông về phía bộ giải mã. Một số hệ thống tạo mã video báo hiệu các vị trí ảnh con so với các ảnh con khác. Điều này gây ra vấn đề nếu các ảnh con bị mất trong chuyển tiếp hoặc được phát riêng biệt cho mục đích được trích xuất riêng biệt. Trong ví dụ này, tập hợp đầy đủ của các ID ảnh con được báo hiệu trong SPS. Ngoài ra, các phần đầu lát bao gồm ID ảnh con chỉ thị ảnh con mà chứa các lát tương ứng với các phần đầu lát. Bằng việc thiết kế mỗi ảnh con theo ID, các ảnh con có thể được định vị và được tạo kích thước mà không tham chiếu tới các ảnh con khác. Điều này đến lượt nó hỗ trợ việc hiệu chỉnh lỗi cũng như các ứng dụng mà chỉ trích xuất một số ảnh con và tránh việc phát các ảnh con khác. Danh sách đầy đủ của tất cả các ID ảnh con có thể được gửi trong SPS cùng với thông tin kích thước tương ứng. Mỗi phần đầu lát có thể chứa ID ảnh con chỉ thị ảnh con mà bao gồm lát tương ứng. Theo cách này, các ảnh con và các lát tương ứng có thể được trích xuất và được định vị mà không tham chiếu tới các ảnh con khác. Do đó, các ID ảnh con hỗ trợ chức năng được nâng cao và/hoặc hiệu suất mã hóa tăng, mà làm giảm việc sử dụng của các tài nguyên mạng, các tài nguyên bộ nhớ, và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó SPS còn bao gồm kích thước ảnh con cho mỗi ảnh con trong số nhiều ảnh con.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó SPS còn bao gồm vị trí ảnh con cho mỗi ảnh con trong số nhiều ảnh con.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó vị trí ảnh con bao gồm khoảng cách dịch vị giữa mẫu phía trên cùng bên trái của ảnh con tương ứng và mẫu phía trên cùng bên trái của ảnh.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó kích thước ảnh con bao gồm chiều cao ảnh con trong các mẫu độ chói và chiều rộng ảnh con trong các mẫu độ chói.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó SPS còn bao gồm việc chỉ thị rằng một hoặc nhiều ảnh con là các ảnh con được ràng buộc chuyển động.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó sự kết hợp của các ảnh con đó bao trùm toàn bộ diện tích của ảnh mà không có khoảng hở và không có khoảng chồng lấp.

Theo một phương án, sáng chế bao gồm thiết bị tạo mã video bao gồm: bộ xử lý, bộ nhớ, bộ thu được ghép nối với bộ xử lý, và bộ phát được ghép nối với bộ xử lý, bộ xử lý, bộ nhớ, bộ thu, và bộ phát được tạo cấu hình để thực hiện phương pháp theo khía cạnh bất kỳ trong số các khía cạnh nêu trên.

Theo một phương án, sáng chế bao gồm phương tiện phi chuyển tiếp đọc được bằng máy tính bao gồm sản phẩm chương trình máy tính để sử dụng bởi thiết bị tạo mã video, sản phẩm chương trình máy tính này bao gồm các lệnh thực thi được bằng máy tính được lưu trữ trên phương tiện phi chuyển tiếp đọc được bằng máy tính này, để khi được thực thi bởi bộ xử lý thì khiến thiết bị tạo mã video này thực hiện phương pháp theo bất kỳ trong số các khía cạnh nêu trên.

Theo một phương án, sáng chế bao gồm bộ giải mã bao gồm: phương tiện thu để thu luồng bit bao gồm SPS, một hoặc nhiều ảnh con được phân chia từ ảnh, và một hoặc nhiều phần đầu lát được liên kết với một hoặc nhiều lát; phương tiện phân tích để: phân tích SPS để nhận được các bộ nhận dạng (các ID) ảnh con cho một hoặc nhiều ảnh con; và phân tích các phần đầu lát để nhận được ID ảnh con hiện tại được liên kết với ảnh con hiện tại, ID ảnh con hiện tại chỉ thị các lát được bao gồm trong ảnh con hiện tại từ một hoặc nhiều ảnh con; phương tiện giải mã để giải mã ảnh con hiện tại dựa vào ID ảnh con hiện tại để tạo chuỗi video; và phương tiện chuyển tiếp để chuyển tiếp chuỗi video cho việc hiển thị.

Tùy ý, theo bất kỳ trong số các khía cạnh nêu trên, thì cách thức thực hiện khác của khía cạnh đó đề xuất rằng bộ giải mã còn được tạo cấu hình để thực hiện phương pháp theo bất kỳ trong số các khía cạnh nêu trên.

Theo một phương án, sáng chế bao gồm bộ mã hóa bao gồm: phương tiện phân chia để: phân chia ảnh thành nhiều ảnh con; và phân chia mỗi ảnh con trong số nhiều ảnh con thành một hoặc nhiều lát; phương tiện mã hóa để: mã hóa thành luồng bit SPS bao gồm các ID ảnh con cho nhiều ảnh con được phân chia từ ảnh; mã hóa thành luồng bit các phần đầu lát bao gồm các ID ảnh con chỉ thị các lát được bao gồm trong các ảnh con được chỉ thị; và mã hóa một hoặc nhiều ảnh con trong số nhiều ảnh con vào luồng bit; và phương tiện lưu trữ để lưu trữ luồng bit cho việc truyền thông về phía bộ giải mã.

Tùy chọn là, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một cách thực hiện khác của khía cạnh đề xuất, trong đó bộ mã hóa còn được tạo cấu hình để thực hiện phương pháp theo khía cạnh bất kỳ trong số các khía cạnh nêu trên.

Để cho rõ ràng, thì phương án bất kỳ trong số các phương án nêu trên là có thể được kết hợp với một hoặc nhiều phương án bất kỳ trong số các phương án nêu trên để tạo ra phương án mới trong phạm vi của sáng chế.

Các dấu hiệu này và các dấu hiệu khác sẽ được hiểu rõ hơn từ phần mô tả chi tiết sau đây dựa vào các hình vẽ kèm theo và các điểm yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

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

Fig.1 là lưu đồ của phương pháp ví dụ để tạo mã tín hiệu video.

Fig.2 là hình vẽ sơ lược của hệ thống tạo mã và giải mã (coding and decoding - codec) ví dụ để tạo mã video.

Fig.3 là hình vẽ sơ lược minh họa bộ mã hóa video ví dụ.

Fig.4 là hình vẽ sơ lược minh họa bộ giải mã video ví dụ.

Fig.5 là hình vẽ sơ lược minh họa luồng bit ví dụ và luồng bit con được trích xuất từ luồng bit.

Fig.6 là hình vẽ sơ lược minh họa ảnh ví dụ được phân chia thành các ảnh con.

Fig.7 là hình vẽ sơ lược minh họa cơ chế ví dụ cho các lát liên quan tới bố cục ảnh con.

Fig.8 là hình vẽ sơ lược minh họa ảnh ví dụ khác được phân chia thành các ảnh con.

Fig.9 là hình vẽ sơ lược của thiết bị tạo mã video ví dụ.

Fig.10 là lưu đồ của phương pháp ví dụ để mã hóa các bộ nhận dạng ảnh con (các ID) trong luồng bit để hỗ trợ việc giải mã của các ảnh con.

Fig.11 là lưu đồ của phương pháp ví dụ của việc giải mã luồng bit của các ảnh con dựa vào các ID ảnh con.

Fig.12 là hình vẽ sơ lược của hệ thống ví dụ để báo hiệu các ID ảnh con thông qua luồng bit.

Mô tả chi tiết sáng chế

Cần hiểu rằng tuy cách thức thực hiện làm minh họa của một hoặc nhiều phương án được cung cấp dưới đây, nhưng các hệ thống và/hoặc các phương pháp được bộc lộ có thể được thực hiện nhờ sử dụng số lượng kỹ thuật bất kỳ, cho dù là hiện đã biết hoặc đang tồn tại. Sáng chế không chỉ giới hạn ở những cách thức thực hiện minh họa đó, các hình vẽ đó, và các kỹ thuật được thể hiện dưới đây, bao gồm các thiết kế và những cách thức thực hiện ví dụ được thể hiện và được mô tả ở đây, mà có thể được cải biến trong phạm vi của các điểm yêu cầu bảo hộ kèm theo cùng với đầy đủ phạm vi của các phương án tương đương của chúng.

Các cụm viết tắt khác nhau được sử dụng ở đây, chẳng hạn như khối cây tạo mã (coding tree block - CTB), đơn vị cây tạo mã (coding tree unit - CTU), đơn vị tạo mã (coding unit - CU), chuỗi video được tạo mã (coded video sequence - CVS), nhóm hợp tác các chuyên gia video (Joint Video Experts Team - JVET), tập hợp ô (tile) được ràng buộc chuyển động (motion constrained tile set - MCTS), đơn vị vận chuyển lớn nhất (maximum transfer unit - MTU), lớp trừu tượng mạng (network abstraction layer - NAL), số đếm thứ tự ảnh (picture order count - POC), phần tải hữu ích chuỗi byte thô (raw byte sequence payload - RBSP), tập hợp thông số chuỗi (sequence parameter set - SPS), mã hóa video vạn năng (versatile video coding - VVC), và dự thảo làm việc (working draft - WD).

Nhiều kỹ thuật nén video có thể được sử dụng để giảm kích thước của các tệp tin video với sự tổn thất dữ liệu nhỏ nhất. Ví dụ, các kỹ thuật nén video có thể bao gồm việc thực hiện việc dự đoán về không gian (ví dụ, nội ảnh) và/hoặc việc dự đoán về thời gian (ví dụ, liên ảnh) để giảm hoặc loại bỏ sự trùng lặp dữ liệu trong các chuỗi video. Đối với việc

tạo mã video dựa trên khối, thì lát video (ví dụ, ảnh video hoặc một phần của ảnh video) có thể được phân chia thành các khối video, mà cũng có thể được gọi là các khối cây, các khối cây tạo mã (CTB), các đơn vị cây tạo mã (CTU), các đơn vị tạo mã (CU), và/hoặc các nút tạo mã. Các khối video trong lát được nội tạo mã (I) của ảnh là được tạo mã nhờ sử dụng việc dự đoán trong không gian dựa vào các mẫu tham chiếu trong các khối lân cận trong cùng một ảnh. Các khối video trong lát dự đoán một chiều (P) hoặc lát dự đoán hai chiều (B) được liên tạo mã của ảnh là có thể được tạo mã nhờ sử dụng việc dự đoán trong không gian dựa vào các mẫu tham chiếu trong các khối lân cận trong cùng ảnh hoặc việc dự đoán về mặt thời gian dựa vào các mẫu tham chiếu trong các ảnh tham chiếu khác. Các ảnh có thể được gọi là các khung và/hoặc các hình ảnh, và các ảnh tham chiếu có thể được gọi là các khung tham chiếu và/hoặc các hình ảnh tham chiếu. Việc dự đoán trong không gian hoặc dự đoán về mặt thời gian cho ra khối dự đoán biểu diễn khối hình ảnh. Dữ liệu dư biểu thị những sự khác biệt điểm ảnh giữa khối hình ảnh gốc và khối dự đoán. Theo đó, khối được liên tạo mã là được mã hóa theo vector chuyển động mà trỏ đến khối của các mẫu tham chiếu mà tạo thành khối dự đoán và dữ liệu dư chỉ thị sự khác biệt giữa khối được tạo mã và khối dự đoán. Khối được nội tạo mã là được mã hóa theo chế độ nội tạo mã và dữ liệu dư đó. Để nén hơn nữa, thì dữ liệu dư có thể được biến đổi từ miền điểm ảnh sang miền biến đổi. Những việc này cho ra các hệ số biến đổi dư, mà có thể được lượng tử hóa. Các hệ số biến đổi được lượng tử hóa có thể được sắp xếp lúc đầu trong mảng hai chiều. Các hệ số biến đổi được lượng tử hóa này có thể được quét theo thứ tự để tạo ra vector một chiều của các hệ số biến đổi. Việc tạo mã entropy có thể được áp dụng để đạt được khả năng nén thậm chí hơn nữa. Các kỹ thuật nén video đó được mô tả chi tiết hơn dưới đây.

Để đảm bảo việc video được mã hóa có thể được giải mã chính xác, thì video được mã hóa và được giải mã theo các tiêu chuẩn tạo mã video tương ứng. Các tiêu chuẩn tạo mã video bao gồm tiêu chuẩn ITU (International Telecommunication Union - hiệp hội viễn thông quốc tế) Standardization Sector (bộ phận tiêu chuẩn hóa) (ITU-T) H.261, ISO (International Organization for Standardization - tổ chức tiêu chuẩn hóa quốc tế)/IEC (International Electrotechnical Commission - uỷ ban kỹ thuật điện quốc tế) MPEG (Motion Picture Experts Group - nhóm chuyên gia ảnh động)-1 Part (phần) 2, ITU-T H.262 hoặc ISO/IEC MPEG-2 Part 2, ITU-T H.263, ISO/IEC MPEG-4 Part 2, AVC (Advanced Video Coding - tạo mã video tiên tiến), còn được gọi là ITU-T H.264 hoặc ISO/IEC MPEG-4 Part 10, và HEVC (High Efficiency Video Coding - tạo mã video hiệu quả cao), còn được gọi là

ITU-T H.265 hoặc MPEG-H Part 2. AVC bao gồm các mở rộng chẳng hạn như tạo mã video khả biến (Scalable Video Coding - SCV), tạo mã video đa điểm quan sát (Multiview Video Coding - MVC) và tạo mã video đa điểm quan sát thêm chiều sâu (Multiview Video Coding plus Depth - MVC+D), và AVC ba chiều (3 Dimensional AVC - 3D-AVC). HEVC bao gồm các mở rộng chẳng hạn như HEVC khả biến (Scalable HEVC - SHVC), HEVC đa điểm quan sát (Multiview HEVC - MV-HEVC), và 3D HEVC (3D-HEVC). Nhóm hợp tác chuyên gia video (JVET) của ITU-T và ISO/IEC đã bắt đầu phát triển tiêu chuẩn tạo mã video được gọi là tạo mã video vận năng (VVC). VVC được bao gồm trong dự thảo làm việc (WD) mà bao gồm JVET-L1001-v9.

Để tạo mã hình ảnh video, thì trước hết hình ảnh đó được phân chia, và các phân chia này được tạo mã vào luồng bit. Có các sơ đồ phân chia ảnh khác nhau khả dụng. Ví dụ, hình ảnh có thể được phân chia thành các lát bình thường, các lát phụ thuộc, các ô, và/hoặc theo kỹ thuật xử lý song song mặt sóng (Wavefront Parallel Processing - WPP). Để cho đơn giản, thì HEVC hạn chế các bộ mã hóa sao cho chỉ có các lát bình thường, các lát phụ thuộc, các ô, WPP, và các tổ hợp của chúng, mới có thể được sử dụng khi phân chia lát thành các nhóm CTB để tạo mã video. Việc phân chia đó có thể được áp dụng để hỗ trợ việc so khớp kích thước đơn vị vận chuyển lớn nhất (MTU), xử lý song song, và giảm độ trễ giữa các đầu. MTU biểu thị lượng dữ liệu lớn nhất mà có thể được phát trong một gói đơn. Nếu phần tải hữu ích của gói vượt quá MTU, thì phần tải hữu ích đó được chia thành hai gói thông qua tiến trình được gọi là phân mảnh.

Lát bình thường, còn được gọi đơn giản là lát, là phần được phân chia của hình ảnh mà có thể được tái tạo độc lập với các lát bình thường khác trong cùng ảnh, tuy nhiên vẫn có một số sự phụ thuộc nhau do các hoạt động lọc vòng. Mỗi lát bình thường được gói trong đơn vị lớp trừu tượng mạng (NAL) của riêng nó để phát. Ngoài ra, sự phụ thuộc của việc dự đoán trong ảnh (dự đoán nội mẫu, dự đoán thông tin chuyển động, dự đoán chế độ tạo mã) và việc tạo mã entropy giữa các đường biên lát là có thể được vô hiệu hóa để hỗ trợ việc tái tạo độc lập. Việc tái tạo độc lập đó hỗ trợ việc song song hóa. Ví dụ, việc song song hóa dựa trên lát bình thường sử dụng việc truyền thông liên bộ xử lý hoặc liên lõi tối thiểu. Tuy nhiên, vì mỗi lát bình thường là độc lập, nên mỗi lát được liên kết với phần đầu lát riêng biệt. Việc sử dụng các lát bình thường có thể gây ra phí tổn tạo mã lớn do chi phí bit của phần đầu lát đối với mỗi lát và do việc thiếu sự dự đoán giữa các đường biên lát. Ngoài ra, các lát bình thường có thể được sử dụng để hỗ trợ việc so khớp đối với các yêu cầu kích thước MTU. Cụ

thể là, vì lát bình thường được gói trong đơn vị NAL riêng biệt và có thể được tạo mã độc lập, nên mỗi lát bình thường cần nhỏ hơn MTU trong các sơ đồ MTU để tránh việc phá vỡ lát thành nhiều gói. Như vậy, mục đích của việc song song hóa và mục đích của việc so khớp kích thước MTU có thể đặt ra các yêu cầu mâu thuẫn đối với bố cục lát trong ảnh.

Các lát phụ thuộc là tương tự như các lát bình thường, nhưng có các phần đầu lát được rút ngắn và cho phép phân chia các đường biên khối cây hình ảnh mà không phá vỡ việc dự đoán trong ảnh. Theo đó, các lát phụ thuộc cho phép phân mảnh lát bình thường thành các đơn vị NAL, để giảm độ trễ giữa các đầu bằng cách cho phép gửi một phần của lát bình thường ra ngoài trước khi hoàn tất việc mã hóa toàn bộ lát bình thường.

Ô là phần được phân chia của hình ảnh mà được tạo ra bởi đường biên ngang và đường biên dọc mà tạo ra các cột và các hàng ô. Các ô có thể được tạo mã theo thứ tự quét mảnh (phải sang trái và trên xuống dưới). Thứ tự quét của các CTB là cục bộ trong ô. Theo đó, các CTB trong ô đầu tiên là được tạo mã theo thứ tự quét mảnh, trước khi tiến đến các CTB trong ô tiếp theo. Tương tự như các lát bình thường, thì các ô phá vỡ những sự phụ thuộc dự đoán trong ảnh cũng như những sự phụ thuộc giải mã entropy. Tuy nhiên, các ô có thể không được bao gồm vào các đơn vị NAL riêng lẻ, và do đó, các ô có thể không được dùng để so khớp kích thước MTU. Mỗi ô có thể được xử lý bởi một bộ xử lý/lỗi, và hoạt động truyền thông liên bộ xử lý/liên lỗi mà được sử dụng cho việc dự đoán trong ảnh giữa các đơn vị xử lý mà giải mã các ô lân cận là có thể được giới hạn để mang phần đầu lát dùng chung (khi các ô liên kề ở cùng lát), và thực hiện việc chia sẻ liên quan đến lọc vòng đối với các mẫu được tái tạo và siêu dữ liệu. Khi nhiều hơn một ô được bao gồm trong lát, thì độ dịch vị byte của điểm vào đối với mỗi ô mà không phải là độ dịch vị điểm vào đầu tiên trong lát là có thể được báo hiệu trong phần đầu lát. Đối với mỗi lát và ô, thì ít nhất một trong số các điều kiện sau đây cần được thỏa mãn: 1) tất cả các khối cây được tạo mã trong lát là thuộc về cùng ô; và 2) tất cả các khối cây được tạo mã trong ô là thuộc về cùng lát.

Theo WPP, thì hình ảnh được phân chia thành các hàng đơn gồm các CTB. Việc giải mã entropy và các cơ chế dự đoán có thể sử dụng dữ liệu từ các CTB ở các hàng khác. Việc xử lý song song được làm cho khả thi thông qua việc giải mã song song đối với các hàng CTB. Ví dụ, hàng hiện tại có thể được giải mã song song với hàng đằng trước. Tuy nhiên, việc giải mã hàng hiện tại được làm trễ tính từ tiến trình giải mã các hàng đằng trước là hai CTB. Độ trễ này bảo đảm rằng dữ liệu liên quan đến CTB bên trên và CTB bên trên và sang phải của CTB hiện tại ở hàng hiện tại là khả dụng trước khi CTB hiện tại được tạo mã. Cách

tiếp cận này xuất hiện như mặt sóng khi được biểu diễn bằng đồ thị. Hình ảnh chứa bao nhiêu hàng CTB thì sự bắt đầu so le này cho phép song song hóa với bấy nhiêu bộ xử lý/lỗi. Vì việc dự đoán trong ảnh giữa các hàng khối cây lân cận trong ảnh là được cho phép, nên hoạt động truyền thông liên bộ xử lý/liên lỗi để cho phép dự đoán trong ảnh có thể là lớn. Việc phân chia WPP có tính đến các kích thước đơn vị NAL. Do đó, WPP không hỗ trợ việc so khớp kích thước MTU. Tuy nhiên, các lát bình thường có thể được sử dụng cùng với WPP, với phí tổn tạo mã nhất định, để thực hiện việc so khớp kích thước MTU theo ý muốn.

Các ô cũng có thể bao gồm các tập hợp ô được ràng buộc chuyển động. Tập hợp ô được ràng buộc chuyển động (MCTS) là tập hợp ô được thiết kế sao cho các vector chuyển động được kết hợp là được hạn chế để trở đến các vị trí của mẫu đầy đủ bên trong MCTS này và đến các vị trí của phân đoạn mẫu mà chỉ yêu cầu các vị trí của mẫu đầy đủ bên trong MCTS này để nội suy. Ngoài ra, việc sử dụng các ứng viên vector chuyển động cho việc dự đoán vector chuyển động về thời gian mà được suy ra từ các khối bên ngoài MCTS là không được phép. Theo cách này, thì mỗi MCTS có thể được giải mã độc lập mà không cần sự tồn tại của các ô không được bao gồm trong MCTS. Các thông điệp MCTS SEI (Supplemental Enhancement Information - thông tin tăng cường bổ sung) về thời gian có thể được dùng để chỉ thị sự tồn tại của các MCTS trong luồng bit và báo hiệu các MCTS này. Thông điệp MCTS SEI cung cấp thông tin bổ sung mà có thể được sử dụng trong quá trình trích xuất luồng bit con MCTS (được quy định như một phần ngữ nghĩa của thông điệp SEI) để tạo ra luồng bit hợp cách cho tập hợp MCTS. Thông tin này bao gồm một số lượng tập hợp thông tin trích xuất, mỗi tập hợp xác định một số lượng tập hợp MCTS và chứa các byte phần tải hữu ích của chuỗi byte thô (RBSP) của các tập hợp thông số video (Video Parameter Set - VPS) thay thế, các tập hợp thông số chuỗi (Sequence Parameter Set - SPS), và các tập hợp thông số ảnh (Picture Parameter Set - PPS) cần được sử dụng trong tiến trình trích xuất luồng bit con MCTS. Khi trích xuất luồng bit con theo tiến trình trích xuất luồng bit con MCTS, thì các tập hợp thông số (VPS, SPS, và PPS) có thể được viết lại hoặc được thay thế, và các phần đầu lát có thể được cập nhật bởi vì một hoặc tất cả trong số các phân tử cú pháp liên quan đến địa chỉ lát (bao gồm `first_slice_segment_in_pic_flag` và `slice_segment_address`) có thể sử dụng các giá trị khác nhau trong luồng bit con trích xuất được.

Ảnh cũng có thể được phân chia thành một hoặc nhiều ảnh con. Ảnh con là tập hợp hình chữ nhật của các nhóm ô/các lát mà bắt đầu với nhóm ô mà có `tile_group_address` bằng không. Mỗi ảnh con có thể là để chỉ PPS riêng biệt và có thể do đó có việc phân chia ô riêng

biệt. Các ảnh con có thể được xử lý như các ảnh trong tiến trình giải mã. Các ảnh con tham chiếu để giải mã ảnh con hiện tại được tạo ra bằng việc trích xuất vùng cùng vị trí với ảnh con hiện tại từ các ảnh tham chiếu trong bộ đệm ảnh được giải mã. Vùng được trích xuất được xử lý như ảnh con được giải mã. Việc liên dự báo có thể diễn ra giữa các ảnh con có cùng kích thước và cùng vị trí trong ảnh. Nhóm ô, cũng được gọi là lát, là chuỗi của các ô liên quan trong ảnh hoặc ảnh con. Một số mục có thể được suy ra để xác định vị trí của ảnh con trong ảnh. Ví dụ, mỗi ảnh con hiện thời có thể được định vị trong vị trí không được chiếm giữ tiếp theo trong thứ tự quét màng CTU trong ảnh mà lớn đủ để chứa ảnh con hiện tại trong các đường biên ảnh.

Ngoài ra, việc phân chia ảnh có thể dựa vào các ô cấp độ ảnh và các ô mức chuỗi. Các ô mức chuỗi có thể bao gồm chức năng của MCTS, và có thể được thực hiện như các ảnh con. Ví dụ, ô cấp độ ảnh được xác định là vùng hình chữ nhật gồm các khối cây tạo mã trong cột ô cụ thể và hàng ô cụ thể trong ảnh. Ô mức chuỗi có thể được xác định như tập hợp của các vùng chữ nhật của các khối cây tạo mã được bao gồm trong các khung khác nhau trong đó mỗi vùng chữ nhật còn bao gồm một hoặc nhiều ô cấp độ ảnh và tập hợp của các vùng chữ nhật của các khối cây tạo mã là có thể giải mã một cách độc lập từ tập hợp bất kỳ khác của các vùng chữ nhật tương tự. Tập hợp nhóm ô mức chuỗi (sequence level tile group set - STGPS) là nhóm của các ô mức chuỗi như vậy. STGPS có thể được báo hiệu trong đơn vị NAL lớp tạo mã không phải video (non-video coding layer - VCL) với ký hiệu định danh được kết hợp (ID) trong phần đầu đơn vị NAL.

Sơ đồ phân chia dựa trên ảnh con đằng trước có thể được liên kết với các vấn đề nhất định. Ví dụ, khi các ảnh con được cho phép tạo ô bên trong các ảnh con (việc phân chia của các ảnh con thành các ô) có thể được sử dụng để hỗ trợ tiến trình song song. Việc phân chia ô của các ảnh con cho các mục đích tiến trình song song có thể thay đổi cho tùy từng ảnh (ví dụ, cho các mục đích cân bằng tải tiến trình song song), và do đó có thể được quản lý ở cấp độ ảnh (ví dụ, trong PPS). Tuy nhiên, việc phân chia ảnh con (việc phân chia của các ảnh thành các ảnh con) có thể được sử dụng để hỗ trợ vùng quan tâm (region of interest - ROI) và việc truy cập ảnh trên cơ sở ảnh con. Trong trường hợp như vậy, việc báo hiệu của các ảnh con hoặc MCTS trong PPS là không hiệu quả.

Trong một ví dụ khác, khi ảnh con bất kỳ trong ảnh được tạo mã như ảnh con được ràng buộc chuyển động theo thời gian, tất cả các ảnh con trong ảnh có thể được tạo mã như các ảnh con được ràng buộc chuyển động theo thời gian. Việc phân chia ảnh như vậy có thể

có giới hạn. Ví dụ, việc tạo mã ảnh con như ảnh con được ràng buộc chuyển động theo thời gian có thể làm giảm hiệu suất tạo mã để trao đổi với chức năng bổ sung. Tuy nhiên, trong vùng của các ứng dụng dựa trên sở thích, thường chỉ một hoặc một vài ảnh con sử dụng chức năng dựa trên ảnh con được ràng buộc chuyển động theo thời gian. Do đó, các ảnh con còn lại chịu ảnh hưởng từ hiệu suất tạo mã bị giảm mà không cung cấp lợi ích thực tế bất kỳ.

Trong một ví dụ khác, các phân tử cú pháp để quy định kích thước của ảnh con có thể được quy định trong các đơn vị của các kích thước CTU độ chói. Theo đó, cả chiều rộng và chiều cao ảnh con nên là bội số nguyên của $CtbSizeY$. Cơ chế quy định chiều rộng và chiều cao ảnh con này có thể dẫn đến các vấn đề khác nhau. Ví dụ, việc phân chia ảnh con chỉ có thể áp dụng cho các ảnh với chiều rộng ảnh và/hoặc chiều cao ảnh mà là bội số nguyên của $CtbSizeY$. Điều này kết xuất việc phân chia ảnh con như không khả dụng cho các ảnh mà chứa các kích thước mà không phải là bội số nguyên của $CtbSizeY$. Nếu việc phân chia ảnh con được áp dụng cho chiều rộng và/hoặc chiều cao ảnh khi kích thước ảnh không phải là bội số nguyên của $CtbSizeY$, việc suy ra của chiều rộng ảnh con và/hoặc chiều cao ảnh con trong các mẫu độ chói cho ảnh con ngoài cùng bên phải và ảnh con dưới cùng sẽ là không chính xác. Việc suy ra không chính xác như vậy sẽ gây ra các kết quả lỗi trong một số công cụ tạo mã.

Trong một ví dụ khác, vị trí của ảnh con trong ảnh có thể không được báo hiệu. Vị trí thay vào đó được suy ra sử dụng quy tắc sau đây. Ảnh con hiện tại được định vị ở vị trí không được chiếm giữ như vậy bên cạnh trong thứ tự quét mảng CTU trong ảnh mà lớn đủ để chứa ảnh con trong các đường biên ảnh. Việc suy ra các vị trí ảnh con theo cách như vậy có thể gây ra các lỗi trong một số trường hợp. Ví dụ, nếu ảnh con bị mất trong khi phát, thì các vị trí của các ảnh con khác được suy ra không chính xác và các mẫu được giải mã được đặt ở các vị trí lỗi. Cùng vấn đề tác động khi các ảnh con đến nơi theo thứ tự sai.

Trong một ví dụ khác, việc giải mã ảnh con có thể yêu cầu việc trích xuất của các ảnh con cùng vị trí trong các ảnh tham chiếu. Điều này có thể gây nên độ phức tạp thêm và dẫn đến các gánh nặng xét về việc sử dụng tài nguyên bộ nhớ và bộ xử lý.

Trong một ví dụ khác, khi ảnh con được chỉ định như ảnh con được ràng buộc chuyển động theo thời gian, các bộ lọc vòng mà duyệt đường biên ảnh con được vô hiệu. Điều này xảy ra bất kể việc liệu các bộ lọc vòng mà duyệt các đường biên ô có được kích hoạt hay

không. Sự giới hạn như vậy có thể là quá chặt chẽ và có thể dẫn đến các giả tượng trực quan cho các ảnh video đang sử dụng nhiều ảnh con.

Trong một ví dụ khác, mối quan hệ giữa SPS, STGPS, PPS và các phần đầu nhóm ô là như sau đây. STGPS là để chỉ SPS, PPS là để chỉ STGPS, và các phần đầu nhóm ô/các phần đầu lát là để chỉ PPS. Tuy nhiên, STGPS và PPS nên trực giao thay vì PPS tham chiếu đến STGPS. Việc bố trí đằng trước cũng có thể không cho phép tất cả các nhóm ô của cùng ảnh khởi việc tham chiếu đến cùng PPS.

Trong một ví dụ khác, mỗi STGPS có thể chứa các ID cho bốn phía của ảnh con. Các ID như vậy được sử dụng để nhận dạng các ảnh con mà dùng chung cùng biên giới sao cho mối quan hệ về không gian tương đối của chúng có thể được xác định. Tuy nhiên, thông tin như vậy có thể là không đủ để suy ra thông tin vị trí và kích thước cho tập hợp nhóm ô lớp chuỗi trong một số trường hợp. Trong các trường hợp khác, việc báo hiệu thông tin vị trí và kích thước có thể là dư thừa.

Trong một ví dụ khác, ID STGPS có thể được báo hiệu trong phần đầu đơn vị NAL của đơn vị NAL VCL sử dụng tám bit. Điều này có thể hỗ trợ với việc trích xuất ảnh con. Việc báo hiệu như vậy có thể làm tăng một cách không cần thiết chiều dài của phần đầu đơn vị NAL. Một vấn đề khác là ở chỗ trừ khi các tập hợp nhóm ô mức chuỗi được ràng buộc để ngăn ngừa các khoảng chồng lấp, một nhóm ô có thể được liên kết với nhiều tập hợp nhóm ô mức chuỗi.

Được bộc lộ ở đây là các cơ chế khác nhau để giải quyết một hoặc nhiều vấn đề được đề cập ở trên. Trong ví dụ thứ nhất, thông tin bố cục cho các ảnh con được bao gồm trong SPS thay vì PPS. Thông tin bố cục ảnh con bao gồm vị trí ảnh con và kích thước ảnh con. Vị trí ảnh con là độ dịch vị giữa mẫu phía trên cùng bên trái của ảnh con và mẫu phía trên cùng bên trái của ảnh. Kích thước ảnh con là chiều cao và chiều rộng của ảnh con như được đo trong các mẫu độ chói. Như được lưu ý ở trên, một số hệ thống bao gồm thông tin tạo ô trong PPS do các ô có thể thay đổi cho tùy từng ảnh. Tuy nhiên, các ảnh con có thể được sử dụng để hỗ trợ các ứng dụng ROI và việc truy cập trên cơ sở ảnh con. Các chức năng này không thay đổi trên cơ sở mỗi ảnh. Ngoài ra, chuỗi video có thể bao gồm SPS đơn (hoặc một cho mỗi đoạn video), và có thể bao gồm nhiều đến một PPS mỗi ảnh. Việc đặt thông tin bố cục cho các ảnh con trong SPS đảm bảo rằng bố cục chỉ được báo hiệu một lần cho chuỗi/đoạn thay vì được báo hiệu một cách dư thừa cho mỗi PPS. Theo đó, việc báo hiệu bố

cục ảnh con trong SPS làm tăng hiệu suất tạo mã và do đó làm giảm việc sử dụng của các tài nguyên mạng, các tài nguyên bộ nhớ, và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã. Ngoài ra, một số hệ thống có thông tin ảnh con được suy ra bởi bộ giải mã. Việc báo hiệu thông tin ảnh con làm giảm xác suất lỗi trong trường hợp các gói bị mất và hỗ trợ chức năng bổ sung xét về việc trích xuất các ảnh con. Theo đó, việc báo hiệu bố cục ảnh con trong SPS cải thiện chức năng của bộ mã hóa và/hoặc bộ giải mã.

Trong ví dụ thứ hai, các chiều rộng ảnh con và các chiều cao ảnh con được ràng buộc là các bội số của kích thước CTU. Tuy nhiên, các ràng buộc này được loại bỏ khi ảnh con được định vị ở ranh giới bên phải của ảnh hoặc ranh giới phía dưới của ảnh, một cách tương ứng. Như được lưu ý ở trên, một số hệ thống video có thể giới hạn các ảnh con để bao gồm các chiều cao và các chiều rộng mà các các bội của kích thước CTU. Điều này ngăn các ảnh con không hoạt động một cách chính xác với nhiều bố cục ảnh. Bằng việc cho phép các ảnh con phía dưới và bên phải bao gồm các chiều cao và các chiều rộng, một cách tương ứng, mà không phải là các bội của kích thước CTU, các ảnh con có thể được sử dụng với ảnh bất kỳ mà không gây ra các lỗi giải mã. Điều này dẫn đến việc làm tăng chức năng bộ mã hóa và bộ giải mã. Ngoài ra, chức năng được tăng cho phép bộ mã hóa tạo mã các ảnh một cách hiệu quả hơn, mà làm giảm việc sử dụng của các tài nguyên mạng, các tài nguyên bộ nhớ, và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Trong ví dụ thứ ba, các ảnh con được ràng buộc để bao trùm ảnh mà không có khoảng trống hoặc khoảng chồng lấp. Như được lưu ý ở trên, một số hệ thống tạo mã video cho phép các ảnh con bao gồm các khoảng trống và các khoảng chồng lấp. Điều này tạo ra tiềm năng cho các nhóm ô/các lát được liên kết với nhiều ảnh con. Nếu điều này được cho phép ở bộ mã hóa, các bộ mã hóa phải được xây dựng để hỗ trợ sơ đồ tạo mã như vậy ngay cả khi sơ đồ giải mã hiếm khi được sử dụng. Bằng việc không cho phép ảnh con các khoảng trống và các khoảng chồng lấp, độ phức tạp của bộ giải mã có thể được giảm do bộ giải mã không được yêu cầu phải tính đến các khoảng trống và các khoảng chồng lấp tiềm năng khi xác định các kích thước và các vị trí ảnh con. Ngoài ra, việc không cho phép các khoảng trống và các khoảng chồng lấp ảnh con làm giảm độ phức tạp của các tiến trình tối ưu hóa tốc độ và độ méo (RDO) ở bộ mã hóa do bộ mã hóa có thể bỏ qua việc xem xét các trường hợp khoảng trống và khoảng chồng lấp khi lựa chọn việc mã hóa cho chuỗi video. Theo đó, việc tránh các khoảng trống và các khoảng chồng lấp có thể làm giảm việc sử dụng của các tài nguyên bộ nhớ và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Trong ví dụ thứ tư, cờ có thể được báo hiệu trong SPS để chỉ thị khi ảnh con là ảnh con được ràng buộc chuyển động theo thời gian. Như được lưu ý ở trên, một số hệ thống có thể thiết đặt một cách chọn lọc tất cả các ảnh con là các ảnh con ràng buộc chuyển động theo thời gian hoặc hoàn toàn không cho phép việc sử dụng của các ảnh con ràng buộc chuyển động theo thời gian. Các ảnh con ràng buộc chuyển động theo thời gian như vậy cung cấp chức năng trích xuất độc lập ở cái giá là hiệu suất tạo mã giảm. Tuy nhiên, trong vùng của các ứng dụng dựa trên sở thích, vùng quan tâm nên được tạo mã cho việc trích xuất độc lập trong khi các vùng bên ngoài của vùng quan tâm không cần chức năng như vậy. Do đó, các ảnh con còn lại chịu ảnh hưởng từ hiệu suất tạo mã bị giảm mà không cung cấp lợi ích thực tế bất kỳ. Theo đó, cờ cho phép việc trộn lẫn của các ảnh con ràng buộc chuyển động theo thời gian mà cung cấp chức năng trích xuất độc lập và các ảnh con được ràng buộc không chuyển động cho hiệu suất mã hóa tăng khi việc trích xuất độc lập là không được mong muốn. Do đó, cờ cho phép chức năng được tăng và/hoặc hiệu suất mã hóa được tăng, mà làm giảm việc sử dụng của các tài nguyên mạng, các tài nguyên bộ nhớ, và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Trong ví dụ thứ năm, tập hợp đầy đủ của các ID ảnh con được báo hiệu trong SPS, và các phần đầu lát bao gồm ID ảnh con chỉ thị ảnh con mà chứa các lát tương ứng. Như được lưu ý ở trên, một số hệ thống báo hiệu các vị trí ảnh con so với các ảnh con khác. Điều này gây ra vấn đề nếu các ảnh con bị mất hoặc được trích xuất riêng biệt. Bằng việc thiết kế mỗi ảnh con theo ID, các ảnh con có thể được định vị và được tạo kích thước mà không tham chiếu tới các ảnh con khác. Điều này đến lượt nó hỗ trợ việc hiệu chỉnh lỗi cũng như các ứng dụng mà chỉ trích xuất một số ảnh con và tránh việc phát các ảnh con khác. Danh sách đầy đủ của tất cả các ID ảnh con có thể được gửi trong SPS cùng với thông tin kích thước tương ứng. Mỗi phần đầu lát có thể chứa ID ảnh con chỉ thị ảnh con mà bao gồm lát tương ứng. Theo cách này, các ảnh con và các lát tương ứng có thể được trích xuất và được định vị mà không tham chiếu tới các ảnh con khác. Do đó, các ID ảnh con hỗ trợ chức năng được nâng cao và/hoặc hiệu suất mã hóa tăng, mà làm giảm việc sử dụng của các tài nguyên mạng, các tài nguyên bộ nhớ, và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Trong ví dụ thứ sáu, các mức được báo hiệu cho mỗi ảnh con. Trong một số hệ thống tạo mã video các mức được báo hiệu cho các ảnh. Mức chỉ thị các tài nguyên phần cứng cần thiết để giải mã ảnh. Như được lưu ý ở trên, các ảnh con khác nhau có thể có chức năng khác nhau trong một số trường hợp và do đó có thể được xử lý một cách khác nhau trong suốt tiến

trình tạo mã. Theo đó, mức trên cơ sở ảnh có thể là không hữu dụng để giải mã một số ảnh con. Do đó, phân bậc lộ này bao gồm các mức cho mỗi ảnh con. Theo cách này, mỗi ảnh con có thể được tạo mã một cách độc lập của các ảnh con khác mà không cần yêu cầu quá mức một cách không cần thiết bộ giải mã bằng việc thiết đặt các yêu cầu giải mã quá cao cho các ảnh con được tạo mã theo các cơ chế ít phức tạp hơn. Thông tin cấp độ ảnh con được báo hiệu hỗ trợ chức năng được tăng và/hoặc hiệu suất mã hóa được tăng, mà làm giảm việc sử dụng của các tài nguyên mạng, các tài nguyên bộ nhớ, và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Fig.1 là lưu đồ của phương pháp vận hành 100 ví dụ của việc tạo mã tín hiệu video. Cụ thể là, tín hiệu video được mã hóa ở bộ mã hóa. Tiến trình mã hóa nén tín hiệu video này bằng cách sử dụng các cơ chế khác nhau để giảm kích thước tệp tin video. Kích thước tệp tin nhỏ hơn sẽ cho phép phát tệp tin video được nén đó về phía người dùng, trong khi giảm được phí tổn băng thông liên quan. Sau đó, bộ giải mã giải mã tệp tin video được nén này để tái tạo tín hiệu video gốc để hiển thị cho người dùng cuối. Tiến trình giải mã thường là đối xứng gương với tiến trình mã hóa để cho phép bộ giải mã tái tạo tín hiệu video một cách nhất quán.

Ở bước 101, tín hiệu video được nhập vào bộ mã hóa. Ví dụ, tín hiệu video này có thể là tệp tin video không được nén mà được lưu trữ trong bộ nhớ. Theo ví dụ khác, thì tệp tin video này có thể được ghi bởi thiết bị ghi video, chẳng hạn camera video, và được mã hóa để hỗ trợ việc tạo luồng trực tiếp (live streaming) đối với video này. Tệp tin video này có thể bao gồm cả thành phần audio và thành phần video. Thành phần video chứa hàng loạt khung hình ảnh mà khi được xem theo trình tự thì cho cảm nhận thị giác về sự chuyển động. Các khung này chứa các điểm ảnh mà được biểu diễn về mặt ánh sáng, ở đây được gọi là các thành phần độ chói (hay các mẫu độ chói), và màu sắc, mà được gọi là các thành phần sắc độ (hay các mẫu màu). Theo một số ví dụ, thì các khung này cũng có thể chứa các giá trị chiều sâu để hỗ trợ xem ba chiều.

Ở bước 103, video được phân chia thành các khối. Việc phân chia bao gồm việc chia tiếp các điểm ảnh trong mỗi khung thành các khối hình vuông và/hoặc hình chữ nhật để nén. Ví dụ, theo tiêu chuẩn tạo mã video hiệu quả cao (High Efficiency Video Coding - HEVC) (còn được gọi là H.265 và MPEG-H Part 2), thì khung có thể trước hết được chia thành các đơn vị cây tạo mã (CTU), tức là các khối có kích thước được định trước (ví dụ, sáu mươi tư điểm ảnh nhân sáu mươi tư điểm ảnh). Các CTU này chứa cả các mẫu độ chói và các mẫu

sắc độ. Các cây tạo mã có thể được sử dụng để chia các CTU thành các khối và sau đó chia tiếp các khối này theo cách đệ quy cho đến khi đạt được các cấu hình hỗ trợ việc mã hóa tiếp. Ví dụ, các thành phần độ chói của khung có thể được chia tiếp cho đến khi các khối riêng lẻ chứa các giá trị sáng tương đối đồng nhất. Ngoài ra, các thành phần sắc độ của khung có thể được chia tiếp cho đến khi các khối riêng lẻ chứa các giá trị màu tương đối đồng nhất. Theo đó, các cơ chế phân chia là biến thiên tùy theo nội dung của các khung video.

Ở bước 105, các cơ chế nén khác nhau được sử dụng để nén các khối ảnh được phân chia ở bước 103. Ví dụ, cơ chế liên dự đoán và/hoặc nội dự đoán có thể được sử dụng. Cơ chế liên dự đoán được thiết kế để tận dụng thực tế rằng các đối tượng trong khung cảnh chung có xu hướng xuất hiện trong các khung liên tiếp. Theo đó, khối thể hiện đối tượng trong khung tham chiếu không cần phải được mô tả lặp lại trong các khung liền kề. Cụ thể là, một đối tượng, chẳng hạn chiếc bàn, có thể vẫn nằm ở vị trí không đổi trên nhiều khung. Do đó, chiếc bàn đó được mô tả một lần và các khung liền kề có thể tham chiếu trở lại khung tham chiếu. Các cơ chế so khớp mẫu có thể được sử dụng để so khớp các đối tượng trên nhiều khung. Ngoài ra, các đối tượng chuyển động có thể được biểu diễn trên nhiều khung, ví dụ, do sự chuyển động của đối tượng hoặc sự chuyển động của camera. Theo một ví dụ cụ thể, thì một video có thể thể hiện chiếc ô tô di chuyển qua màn hình trên nhiều khung. Các vector chuyển động có thể được sử dụng để mô tả sự chuyển động đó. Vector chuyển động là vector hai chiều mà cung cấp độ dịch vị từ các tọa độ của đối tượng trong khung đến các tọa độ của đối tượng trong khung tham chiếu. Như vậy, cơ chế liên dự đoán có thể mã hóa khối ảnh trong khung hiện tại dưới dạng tập hợp các vector chuyển động chỉ thị độ dịch vị từ khối tương ứng trong khung tham chiếu.

Cơ chế nội dự đoán mã hóa các khối trong khung chung. Cơ chế nội dự đoán tận dụng thực tế là các thành phần độ chói và sắc độ có xu hướng xếp thành nhóm trong khung. Ví dụ, một mảng màu xanh lục ở một phần của một cái cây thì có xu hướng được đặt kề với các mảng màu xanh lục tương tự. Cơ chế nội dự đoán sử dụng nhiều chế độ dự đoán có hướng (ví dụ, ba mươi ba chế độ theo HEVC), chế độ phẳng, và chế độ dòng điện một chiều (Direct Current - DC). Các chế độ có hướng này chỉ thị rằng khối hiện tại là tương tự/giống với các mẫu của khối lân cận theo hướng tương ứng. Chế độ phẳng chỉ thị rằng một loạt khối dọc theo hàng/cột (ví dụ, mặt phẳng) là có thể được nội suy dựa trên các khối lân cận tại các mép của hàng. Trên thực tế, chế độ phẳng chỉ thị sự chuyển tiếp trơn của ánh sáng/màu sắc trên hàng/cột bằng cách sử dụng độ dốc gần như không đổi trong việc thay đổi các giá trị. Chế

độ DC được sử dụng để làm nhãn đường biên và chỉ thị rằng khối là tương tự/giống với giá trị trung bình được liên kết với các mẫu của tất cả các khối lân cận mà được liên kết với các hướng xiên góc của các chế độ dự đoán có hướng. Theo đó, các khối nội dự đoán có thể biểu thị các khối ảnh dưới dạng các giá trị của chế độ dự đoán có quan hệ khác nhau thay vì các giá trị thực tế. Ngoài ra, các khối liên dự đoán có thể biểu thị các khối ảnh dưới dạng các giá trị vector chuyển động thay vì các giá trị thực tế. Trong trường hợp nào trong số hai trường hợp đó, thì các khối dự đoán cũng có thể không biểu thị một cách chính xác các khối hình ảnh trong một số trường hợp. Những sự khác biệt bất kỳ là được lưu trữ ở các khối dư. Các phép biến đổi có thể được áp dụng cho các khối dư để nén tệp tin hơn nữa.

Ở bước 107, các kỹ thuật lọc khác nhau có thể được áp dụng. Theo HEVC, thì các bộ lọc là được áp dụng theo sơ đồ lọc trong vòng lặp. Việc dự đoán dựa trên khối nêu trên có thể dẫn đến việc tạo ra các hình ảnh có khối tại bộ giải mã. Ngoài ra, sơ đồ dự đoán dựa trên khối có thể mã hóa khối và sau đó tái tạo khối được mã hóa để về sau sử dụng làm khối tham chiếu. Sơ đồ lọc trong vòng lặp này áp dụng lặp đi lặp lại các bộ lọc khử nhiễu, các bộ lọc khử khối, các bộ lọc vòng thích ứng, và các bộ lọc dịch vị thích ứng mẫu (Sample Adaptive Offset - SAO) cho các khối/các khung. Các bộ lọc này giảm bớt các thành phần giả tạo khối đó để tệp tin được mã hóa có thể được tái tạo chính xác. Ngoài ra, các bộ lọc này giảm bớt các thành phần giả ở các khối tham chiếu tái tạo được, để các thành phần giả ít có khả năng tạo ra các thành phần giả bổ sung ở các khối sau đó mà được mã hóa dựa trên các khối tham chiếu tái tạo được.

Khi tín hiệu video đã được phân chia, được nén, và được lọc, thì dữ liệu thu được được mã hóa trong luồng bit ở bước 109. Luồng bit này bao gồm dữ liệu nêu trên cũng như dữ liệu báo hiệu bất kỳ mà được mong muốn để hỗ trợ việc tái tạo tín hiệu video phù hợp tại bộ giải mã. Ví dụ, dữ liệu đó có thể bao gồm dữ liệu phân chia, dữ liệu dự đoán, các khối dư, và các cờ khác nhau mà cung cấp các lệnh tạo mã cho bộ giải mã. Luồng bit này có thể được lưu trữ trong bộ nhớ để phát về phía bộ giải mã khi có yêu cầu. Luồng bit này cũng có thể được phát quảng bá và/hoặc phát đa điểm về phía các bộ giải mã. Việc tạo ra luồng bit này là tiến trình lặp đi lặp lại. Theo đó, các bước 101, 103, 105, 107, và 109 có thể xuất hiện liên tục và/hoặc đồng thời trên nhiều khung và khối. Thứ tự được thể hiện trên Fig.1 là được thể hiện để cho việc mô tả được rõ ràng và dễ dàng, chứ không nhằm giới hạn tiến trình tạo mã video ở thứ tự cụ thể nào.

Bộ giải mã thu luồng bit này và bắt đầu tiến trình giải mã ở bước 111. Cụ thể là, bộ giải mã sử dụng sơ đồ giải mã entropy để chuyển đổi luồng bit này thành dữ liệu cú pháp và dữ liệu video tương ứng. Bộ giải mã sử dụng dữ liệu cú pháp này từ luồng bit để xác định các phần chia đối với các khung ở bước 111. Việc phân chia này cần khớp với các kết quả của việc phân chia khối ở bước 103. Việc mã hóa/giải mã entropy như được sử dụng ở bước 111 sẽ được mô tả. Bộ mã hóa chọn nhiều sự lựa chọn trong tiến trình nén, chẳng hạn chọn các sơ đồ phân chia khối từ một số sự lựa chọn khả thi dựa trên sự định vị trong không gian của các giá trị ở (các) hình ảnh đầu vào. Việc báo hiệu những sự lựa chọn chính xác này có thể sử dụng một lượng lớn các bin. Như được sử dụng ở đây, bin là giá trị nhị phân mà được coi như biến số (ví dụ, giá trị bit mà có thể biến thiên tùy theo ngữ cảnh). Việc tạo mã entropy cho phép bộ mã hóa loại bỏ các tùy chọn bất kỳ mà rõ ràng là không khả thi cho trường hợp cụ thể, để lại tập hợp các tùy chọn có thể được phép. Sau đó, mỗi tùy chọn có thể được phép được ấn định từ mã. Độ dài của các từ mã là dựa trên số lượng tùy chọn có thể được phép (ví dụ, một bin cho hai tùy chọn, hai bin cho ba đến bốn tùy chọn, v.v.). Bộ mã hóa sau đó mã hóa từ mã cho tùy chọn được chọn lựa. Sơ đồ này giảm kích thước của các từ mã vì các từ mã là lớn theo ý muốn để chỉ thị duy nhất một sự lựa chọn từ tập hợp con cỡ nhỏ gồm các tùy chọn có thể được phép, ngược lại so với việc chỉ thị duy nhất sự lựa chọn đó từ tập hợp có thể là lớn gồm tất cả các tùy chọn khả thi. Sau đó, bộ giải mã giải mã sự lựa chọn này bằng cách xác định tập hợp các tùy chọn có thể được phép theo cách tương tự như bộ mã hóa. Bằng cách xác định tập hợp các tùy chọn có thể được phép, thì bộ giải mã có thể đọc từ mã đó và xác định sự lựa chọn mà được chọn bởi bộ mã hóa.

Ở bước 113, bộ giải mã thực hiện việc giải mã khối. Cụ thể là, bộ giải mã sử dụng các phép biến đổi ngược để tạo ra các khối dư. Sau đó, bộ giải mã sử dụng các khối dư đó và các khối dự đoán tương ứng để tái tạo các khối ảnh theo việc phân chia. Các khối dự đoán này có thể bao gồm cả các khối nội dự đoán và các khối liên dự đoán như được tạo ra tại bộ mã hóa ở bước 105. Sau đó, các khối ảnh được tái tạo được đặt vào các khung của tín hiệu video tái tạo được theo dữ liệu phân chia xác định được ở bước 111. Cú pháp cho bước 113 cũng có thể được báo hiệu trong luồng bit thông qua việc tạo mã entropy như đã mô tả trên đây.

Ở bước 115, việc lọc được thực hiện trên các khung của tín hiệu video tái tạo được theo cách tương tự như bước 107 tại bộ mã hóa. Ví dụ, các bộ lọc khử nhiễu, các bộ lọc khử khối, các bộ lọc vòng thích ứng, và các bộ lọc SAO có thể được áp dụng cho các khung để

loại bỏ các thành phần giả tạo khỏi. Khi các khung được lọc, thì tín hiệu video có thể được xuất đến thiết bị hiển thị ở bước 117 để người dùng cuối xem.

Fig.2 là hình vẽ sơ lược của hệ thống tạo mã và giải mã (codec) 200 ví dụ để tạo mã video. Cụ thể là, hệ thống codec 200 cung cấp chức năng để hỗ trợ việc thực hiện phương pháp vận hành 100. Hệ thống codec 200 được tổng quát hóa để thể hiện các thành phần được sử dụng ở cả bộ mã hóa và bộ giải mã. Hệ thống codec 200 thu và phân chia tín hiệu video như được mô tả dựa vào các bước 101 và 103 ở phương pháp vận hành 100, để thu được tín hiệu video được phân chia 201. Sau đó, hệ thống codec 200 nén tín hiệu video được phân chia 201 này vào luồng bit được tạo mã khi hoạt động như bộ mã hóa như được mô tả dựa vào các bước 105, 107, và 109 ở phương pháp 100. Khi hoạt động như bộ giải mã thì hệ thống codec 200 tạo ra tín hiệu video đầu ra từ luồng bit này như được mô tả dựa vào các bước 111, 113, 115, và 117 ở phương pháp vận hành 100. Hệ thống codec 200 bao gồm thành phần điều khiển bộ tạo mã chung 211, thành phần biến đổi, thay đổi tỷ lệ và lượng tử hóa 213, thành phần ước lượng nội ảnh 215, thành phần dự đoán nội ảnh 217, thành phần bù chuyển động 219, thành phần ước lượng chuyển động 221, thành phần thay đổi tỷ lệ và biến đổi ngược 229, thành phần phân tích điều khiển bộ lọc 227, thành phần bộ lọc trong vòng lặp 225, thành phần bộ đệm ảnh được giải mã 223, và thành phần định dạng phần đầu và tạo mã số học nhị phân thích ứng ngữ cảnh (Context Adaptive Binary Arithmetic Coding - CABAC) 231. Các thành phần đó được ghép nối như được thể hiện. Trên Fig.2, thì các đường màu đen chỉ thị sự di chuyển của dữ liệu cần được mã hóa/giải mã, còn các đường nét đứt thì chỉ thị sự di chuyển của dữ liệu điều khiển mà điều khiển sự hoạt động của các thành phần khác. Các thành phần của hệ thống codec 200 có thể đều có mặt ở bộ mã hóa. Bộ giải mã có thể bao gồm tập hợp con của các thành phần của hệ thống codec 200. Ví dụ, bộ giải mã có thể bao gồm thành phần dự đoán nội ảnh 217, thành phần bù chuyển động 219, thành phần thay đổi tỷ lệ và biến đổi ngược 229, thành phần bộ lọc trong vòng lặp 225, và thành phần bộ đệm ảnh được giải mã 223. Các thành phần này sau đây sẽ được mô tả.

Tín hiệu video được phân chia 201 là chuỗi video ghi được mà đã được phân chia thành các khối điểm ảnh bởi cây tạo mã. Cây tạo mã sử dụng các chế độ chia khác nhau để chia tiếp khối các điểm ảnh thành các khối các điểm ảnh nhỏ hơn. Sau đó, các khối này có thể còn được chia tiếp thành các khối nhỏ hơn. Các khối này có thể được gọi là các nút trên cây tạo mã. Các nút mẹ lớn hơn được chia thành các nút con nhỏ hơn. Số lần nút được chia nhỏ được gọi là chiều sâu của nút/cây tạo mã. Các khối được chia có thể được bao gồm trong

các đơn vị tạo mã (CU) trong một số trường hợp. Ví dụ, CU có thể là phần con của CTU mà chứa khối độ chói, (các) khối sắc độ chênh lệch đỏ (Cr), và (các) khối sắc độ chênh lệch xanh lam (Cb) cùng với các lệnh cú pháp tương ứng dành cho CU đó. Các chế độ chia có thể bao gồm cây nhị phân (Binary Tree - BT), cây tam phân (Triple Tree - TT), và cây tứ phân (Quad Tree - QT) được sử dụng để phân chia nút lần lượt thành hai, ba, hoặc bốn nút con, với các hình dạng biến thiên tùy theo các chế độ chia được sử dụng. Tín hiệu video được phân chia 201 được chuyển tiếp đến thành phần điều khiển bộ tạo mã chung 211, thành phần biến đổi, thay đổi tỷ lệ và lượng tử hóa 213, thành phần ước lượng nội ảnh 215, thành phần phân tích điều khiển bộ lọc 227, và thành phần ước lượng chuyển động 221 để nén.

Thành phần điều khiển bộ tạo mã chung 211 được tạo cấu hình để đưa ra những quyết định liên quan đến việc tạo mã các hình ảnh của chuỗi video vào luồng bit theo các ràng buộc ứng dụng. Ví dụ, thành phần điều khiển bộ tạo mã chung 211 quản lý việc tối ưu hóa tốc độ bit/kích thước luồng bit so với chất lượng tái tạo. Những quyết định đó có thể được đưa ra dựa trên sự khả dụng của không gian lưu trữ/băng thông và các yêu cầu về độ phân giải hình ảnh. Thành phần điều khiển bộ tạo mã chung 211 còn quản lý việc sử dụng bộ đệm dựa vào tốc độ phát để giảm bớt vấn đề chạy dưới mức bộ đệm và vấn đề tràn bộ đệm. Để quản lý các vấn đề này, thì thành phần điều khiển bộ tạo mã chung 211 quản lý việc phân chia, việc dự đoán, và việc lọc của các thành phần còn lại. Ví dụ, thành phần điều khiển bộ tạo mã chung 211 có thể, theo cách động, tăng sự phức tạp nén để tăng độ phân giải và tăng mức sử dụng băng thông, hoặc giảm sự phức tạp nén để giảm độ phân giải và mức sử dụng băng thông. Do đó, thành phần điều khiển bộ tạo mã chung 211 điều khiển các thành phần còn lại của hệ thống codec 200 để cân bằng giữa chất lượng tái tạo tín hiệu video và tốc độ bit. Thành phần điều khiển bộ tạo mã chung 211 tạo ra dữ liệu điều khiển, mà điều khiển sự hoạt động của các thành phần còn lại. Dữ liệu điều khiển này cũng được chuyển tiếp đến thành phần định dạng phần đầu và CABAC 231 để được mã hóa trong luồng bit để báo hiệu các thông số cho việc giải mã tại bộ giải mã.

Tín hiệu video được phân chia 201 cũng được gửi đến thành phần ước lượng chuyển động 221 và thành phần bù chuyển động 219 để liên dự đoán. Khung hoặc lát của tín hiệu video được phân chia 201 có thể được chia thành nhiều khối video. Thành phần ước lượng chuyển động 221 và thành phần bù chuyển động 219 thực hiện việc tạo mã liên dự đoán đối với khối video thu được so với một hoặc nhiều khối trong một hoặc nhiều khung tham chiếu

để cung cấp việc dự đoán về mặt thời gian. Hệ thống codec 200 có thể thực hiện nhiều chẳng tạo mã, ví dụ, để chọn chế độ tạo mã thích hợp cho mỗi khối của dữ liệu video.

Thành phần ước lượng chuyển động 221 và thành phần bù chuyển động 219 có thể được tích hợp ở mức chiều cao, nhưng được thể hiện riêng biệt nhằm mục đích minh họa. Việc ước lượng chuyển động, mà được thực hiện bởi thành phần ước lượng chuyển động 221, là tiến trình tạo ra các vectơ chuyển động, để ước lượng sự chuyển động cho các khối video. Ví dụ, vectơ chuyển động có thể chỉ thị sự dịch chuyển của đối tượng được tạo mã so với khối dự đoán. Khối dự đoán là khối mà được tìm thấy là gần như khớp với khối cần được tạo mã, xét về sự khác biệt điểm ảnh. Khối dự đoán cũng có thể được gọi là khối tham chiếu. Sự khác biệt điểm ảnh đó có thể được xác định bởi tổng chênh lệch tuyệt đối (Sum of Absolute Difference - SAD), tổng chênh lệch bình phương (Sum of Square Difference - SSD), hoặc các kết quả đo (metric - mét-ric) chênh lệch khác. HEVC sử dụng một số đối tượng được tạo mã bao gồm CTU, các khối cây tạo mã (CTB), và các CU. Ví dụ, CTU có thể được chia thành các CTB, mà sau đó có thể được chia thành các CB để được chứa trong các CU. CU có thể được mã hóa dưới dạng đơn vị dự đoán (Prediction Unit - PU) chứa dữ liệu dự đoán và/hoặc đơn vị biến đổi (Transform Unit - TU) chứa dữ liệu dư được biến đổi dành cho CU đó. Thành phần ước lượng chuyển động 221 tạo ra các vectơ chuyển động, các PU, và các TU nhờ sử dụng việc phân tích tốc độ - độ méo như một phần của tiến trình tối ưu hóa tốc độ và độ méo. Ví dụ, thành phần ước lượng chuyển động 221 có thể xác định nhiều khối tham chiếu, nhiều vectơ chuyển động, v.v., dành cho khối/khung hiện tại, và có thể chọn các khối tham chiếu, các vectơ chuyển động, v.v., mà có các đặc điểm tốc độ - độ méo tốt nhất. Các đặc điểm tốc độ - độ méo tốt nhất này cân bằng cả chất lượng của việc tái tạo video (ví dụ, lượng mất mát dữ liệu do việc nén) với hiệu quả tạo mã (ví dụ, kích thước của việc mã hóa cuối cùng).

Theo một số ví dụ, thì hệ thống codec 200 có thể tính toán các giá trị dành cho các vị trí điểm ảnh nguyên con của các ảnh tham chiếu được lưu trữ trong thành phần bộ đệm ảnh được giải mã 223. Ví dụ, hệ thống codec video 200 có thể nội suy các giá trị của các vị trí phân tử điểm ảnh, các vị trí phân tám điểm ảnh, hoặc các vị trí phân đoạn điểm ảnh khác của ảnh tham chiếu. Do đó, thành phần ước lượng chuyển động 221 có thể thực hiện việc tìm kiếm chuyển động so với các vị trí điểm ảnh đầy đủ và các vị trí phân đoạn điểm ảnh và xuất ra vectơ chuyển động với độ chính xác của phân đoạn điểm ảnh. Thành phần ước lượng chuyển động 221 tính toán vectơ chuyển động dành cho PU của khối video trong lát được

liên tạo mã bằng cách so sánh vị trí của PU với vị trí của khối dự đoán của ảnh tham chiếu. Thành phần ước lượng chuyển động 221 xuất vector chuyển động tính toán được dưới dạng dữ liệu chuyển động đến thành phần định dạng phần đầu và CABAC 231 để mã hóa và sự chuyển động đến thành phần bù chuyển động 219.

Việc bù chuyển động, mà được thực hiện bởi thành phần bù chuyển động 219, là có thể bao gồm việc tìm nạp hoặc tạo ra khối dự đoán dựa trên vector chuyển động được xác định bởi thành phần ước lượng chuyển động 221. Như đã nêu, thành phần ước lượng chuyển động 221 và thành phần bù chuyển động 219 có thể được tích hợp về mặt chức năng, theo một số ví dụ. Khi thu được vector chuyển động dành cho PU của khối video hiện tại, thì thành phần bù chuyển động 219 có thể định vị khối dự đoán mà vector chuyển động này trỏ đến đó. Sau đó, khối video dư được tạo ra bằng cách trừ đi các giá trị điểm ảnh của khối dự đoán từ các giá trị điểm ảnh của khối video hiện tại đang được tạo mã, để tạo ra các giá trị chênh lệch điểm ảnh. Nói chung, thì thành phần ước lượng chuyển động 221 thực hiện việc ước lượng chuyển động so với các thành phần độ chói, và thành phần bù chuyển động 219 sử dụng các vector chuyển động tính được dựa trên các thành phần độ chói này cho cả các thành phần sắc độ và các thành phần độ chói. Khối dự đoán và khối dư này được chuyển tiếp đến thành phần biến đổi, thay đổi tỷ lệ và lượng tử hóa 213.

Tín hiệu video được phân chia 201 cũng được gửi đến thành phần ước lượng nội ảnh 215 và thành phần dự đoán nội ảnh 217. Cũng như đối với thành phần ước lượng chuyển động 221 và thành phần bù chuyển động 219, thì thành phần ước lượng nội ảnh 215 và thành phần dự đoán nội ảnh 217 có thể được tích hợp mức chiều cao, nhưng được thể hiện riêng biệt nhằm mục đích minh họa. Thành phần ước lượng nội ảnh 215 và thành phần dự đoán nội ảnh 217 nội dự đoán khối hiện tại so với các khối trong khung hiện tại, thay vì việc liên dự đoán mà được thực hiện bởi thành phần ước lượng chuyển động 221 và thành phần bù chuyển động 219 giữa các khung, như đã mô tả trên đây. Cụ thể là, thành phần ước lượng nội ảnh 215 xác định chế độ nội dự đoán để sử dụng để mã hóa khối hiện tại. Theo một số ví dụ, thì thành phần ước lượng nội ảnh 215 chọn chế độ nội dự đoán thích hợp để mã hóa khối hiện tại từ nhiều chế độ nội dự đoán đã được kiểm tra. Sau đó, các chế độ nội dự đoán được chọn được chuyển tiếp đến thành phần định dạng phần đầu và CABAC 231 để mã hóa.

Ví dụ, thành phần ước lượng nội ảnh 215 tính toán các giá trị tốc độ - độ méo nhờ sử dụng việc phân tích tốc độ - độ méo đối với các chế độ nội dự đoán khác nhau đã được kiểm tra, và chọn chế độ nội dự đoán có các đặc điểm tốc độ - độ méo tốt nhất trong số các chế độ

đã được kiểm tra đó. Việc phân tích tốc độ - độ méo thường xác định lượng méo (hoặc sai số) giữa khối được mã hóa và khối gốc chưa được mã hóa mà đã được mã hóa để tạo ra khối được mã hóa, cũng như tốc độ bit (ví dụ, số lượng bit) được sử dụng để tạo ra khối được mã hóa. Thành phần ước lượng nội ảnh 215 tính toán các tỷ số từ những sự méo và các tốc độ đối với các khối được mã hóa khác nhau để xác định xem chế độ nội dự đoán nào có giá trị tốc độ - độ méo tốt nhất cho khối đó. Ngoài ra, thành phần ước lượng nội ảnh 215 có thể được tạo cấu hình để tạo mã các khối chiều sâu của bản đồ chiều sâu nhờ sử dụng chế độ dựng mô hình chiều sâu (Depth Modeling Mode - DMM) dựa trên sự tối ưu hóa tốc độ - độ méo (Rate-Distortion Optimization - RDO).

Thành phần dự đoán nội ảnh 217 có thể tạo ra khối dư từ khối dự đoán dựa trên các chế độ nội dự đoán được chọn mà được xác định bởi thành phần ước lượng nội ảnh 215 khi được thực hiện trên bộ mã hóa, hoặc đọc khối dư từ luồng bit khi được thực hiện trên bộ giải mã. Khối dư này bao gồm sự chênh lệch về các giá trị giữa khối dự đoán và khối gốc, được biểu diễn dưới dạng ma trận. Sau đó, khối dư này được chuyển tiếp đến thành phần biến đổi, thay đổi tỷ lệ và lượng tử hóa 213. Thành phần ước lượng nội ảnh 215 và thành phần dự đoán nội ảnh 217 có thể hoạt động trên cả thành phần độ chói và thành phần sắc độ.

Thành phần biến đổi, thay đổi tỷ lệ và lượng tử hóa 213 được tạo cấu hình để nén tiếp khối dư. Thành phần biến đổi, thay đổi tỷ lệ và lượng tử hóa 213 áp dụng phép biến đổi, chẳng hạn biến đổi cosin rời rạc (Discrete Cosine Transform - DCT), biến đổi sin rời rạc (Discrete Sine Transform - DST), hoặc phép biến đổi tương tự về mặt khái niệm, đối với khối dư, để tạo ra khối video bao gồm các giá trị hệ số biến đổi dư. Phép biến đổi wavelet, biến đổi nguyên, biến đổi dải con hoặc các loại phép biến đổi khác cũng có thể được sử dụng. Phép biến đổi này có thể chuyển đổi thông tin dư từ miền giá trị điểm ảnh sang miền biến đổi, chẳng hạn miền tần số. Thành phần biến đổi, thay đổi tỷ lệ và lượng tử hóa 213 cũng được tạo cấu hình để thay đổi tỷ lệ thông tin dư đã được biến đổi, ví dụ, dựa trên tần số. Việc thay đổi tỷ lệ đó bao gồm việc áp dụng hệ số tỷ lệ cho thông tin dư sao cho các thông tin tần số khác nhau được lượng tử hóa tại các độ chi tiết khác nhau, điều này có thể ảnh hưởng đến chất lượng hiển thị cuối cùng của video tái tạo được. Thành phần biến đổi, thay đổi tỷ lệ và lượng tử hóa 213 cũng được tạo cấu hình để lượng tử hóa các hệ số biến đổi để giảm tốc độ bit hơn nữa. Tiến trình lượng tử hóa có thể giảm chiều sâu bit được liên kết với một số hoặc tất cả trong số các hệ số đó. Mức độ lượng tử hóa có thể được biến đổi bằng cách điều chỉnh thông số lượng tử hóa. Theo một số ví dụ, thì sau đó thành phần biến đổi, thay đổi tỷ lệ và

lượng tử hóa 213 có thể thực hiện việc quét ma trận bao gồm các hệ số biến đổi đã được lượng tử hóa. Các hệ số biến đổi đã được lượng tử hóa được chuyển tiếp đến thành phần định dạng phần đầu và CABAC 231 để được mã hóa trong luồng bit.

Thành phần thay đổi tỷ lệ và biến đổi ngược 229 áp dụng thao tác ngược của thành phần biến đổi, thay đổi tỷ lệ và lượng tử hóa 213 để hỗ trợ việc ước lượng chuyển động. Thành phần thay đổi tỷ lệ và biến đổi ngược 229 áp dụng việc thay đổi tỷ lệ, biến đổi, và/hoặc lượng tử hóa ngược để tái tạo khối dư trong miền điểm ảnh, ví dụ, để sử dụng về sau làm khối tham chiếu mà có thể trở thành khối dự đoán đối với khối hiện tại khác. Thành phần ước lượng chuyển động 221 và/hoặc thành phần bù chuyển động 219 có thể tính toán khối tham chiếu bằng cách cộng khối dư trở lại khối dự đoán tương ứng để sử dụng trong việc ước lượng chuyển động của khối/khung về sau. Các bộ lọc được áp dụng cho các khối tham chiếu tái tạo được để giảm bớt các thành phần giả được tạo ra trong quá trình thay đổi tỷ lệ, lượng tử hóa, và biến đổi. Các thành phần giả đó vốn có thể gây ra việc dự đoán không chính xác (và tạo ra các thành phần giả bổ sung) khi các khối sau đó được dự đoán.

Thành phần phân tích điều khiển bộ lọc 227 và thành phần bộ lọc trong vòng lặp 225 áp dụng các bộ lọc cho các khối dư và/hoặc cho các khối ảnh được tái tạo. Ví dụ, khối dư đã được biến đổi từ thành phần thay đổi tỷ lệ và biến đổi ngược 229 có thể được kết hợp với khối dự đoán tương ứng từ thành phần dự đoán nội ảnh 217 và/hoặc thành phần bù chuyển động 219 để tái tạo khối hình ảnh gốc. Sau đó, các bộ lọc này có thể được áp dụng cho khối hình ảnh được tái tạo. Theo một số ví dụ thì thay vào đó, các bộ lọc này có thể được áp dụng cho các khối dư. Cũng như đối với các thành phần khác trên Fig.2, thì thành phần phân tích điều khiển bộ lọc 227 và thành phần bộ lọc trong vòng lặp 225 là được tích hợp mức chiều cao và có thể được thực hiện cùng nhau, nhưng được thể hiện riêng biệt nhằm mục đích minh họa. Các bộ lọc mà được áp dụng cho các khối tham chiếu tái tạo được là được áp dụng cho các vùng không gian cụ thể và bao gồm nhiều thông số để điều chỉnh việc các bộ lọc đó được áp dụng như thế nào. Thành phần phân tích điều khiển bộ lọc 227 phân tích các khối tham chiếu tái tạo được để xác định xem các bộ lọc đó nên được áp dụng ở đâu và thiết đặt các thông số tương ứng. Dữ liệu đó được chuyển tiếp đến thành phần định dạng phần đầu và CABAC 231 dưới dạng dữ liệu điều khiển bộ lọc để mã hóa. Thành phần bộ lọc trong vòng lặp 225 áp dụng các bộ lọc đó dựa trên dữ liệu điều khiển bộ lọc này. Các bộ lọc đó có thể bao gồm bộ lọc khử khối, bộ lọc khử nhiễu, bộ lọc SAO, và bộ lọc vòng thích ứng. Các bộ

lọc đó có thể được áp dụng trong miền không gian/miền điểm ảnh (ví dụ, trên khối điểm ảnh tái tạo được) hoặc trong miền tần số, tùy theo ví dụ.

Khi hoạt động như bộ mã hóa, thì khối hình ảnh được tái tạo đã được lọc, khối dư, và/hoặc khối dự đoán là được lưu trữ ở thành phần bộ đệm ảnh được giải mã 223 để sử dụng về sau trong quá trình ước lượng chuyển động như đã mô tả trên đây. Khi hoạt động như bộ giải mã, thì thành phần bộ đệm ảnh được giải mã 223 lưu trữ và chuyển tiếp các khối được tái tạo và được lọc về phía thiết bị hiển thị như một phần của tín hiệu video đầu ra. Thành phần bộ đệm ảnh được giải mã 223 có thể là thiết bị nhớ bất kỳ mà có khả năng lưu trữ các khối dự đoán, các khối dư, và/hoặc các khối ảnh được tái tạo.

Thành phần định dạng phần đầu và CABAC 231 thu dữ liệu từ các thành phần khác nhau của hệ thống codec 200 và mã hóa dữ liệu đó vào luồng bit được tạo mã để phát về phía bộ giải mã. Cụ thể là, thành phần định dạng phần đầu và CABAC 231 tạo ra các phần đầu khác nhau để mã hóa dữ liệu điều khiển, chẳng hạn dữ liệu điều khiển tổng và dữ liệu điều khiển bộ lọc. Ngoài ra, dữ liệu dự đoán, bao gồm dữ liệu nội dự đoán và dữ liệu chuyển động, cũng như dữ liệu dư dưới dạng dữ liệu hệ số biến đổi đã được lượng tử hóa, đều được mã hóa trong luồng bit này. Luồng bit cuối cùng bao gồm tất cả thông tin cần thiết cho bộ giải mã để tái tạo tín hiệu video được phân chia gốc 201. Thông tin đó cũng có thể bao gồm các bảng chỉ số chế độ nội dự đoán (còn được gọi là các bảng ánh xạ từ mã), các định nghĩa về các ngữ cảnh mã hóa dành cho các khối khác nhau, các chỉ thị về các chế độ nội dự đoán khả dĩ nhất, chỉ thị về thông tin về phân chia, v.v.. Dữ liệu như vậy có thể được mã hóa bằng việc thực hiện tạo mã entropy. Ví dụ, thông tin này có thể được mã hóa bằng cách sử dụng kỹ thuật tạo mã độ dài biến thiên thích ứng ngữ cảnh (Context Adaptive Variable Length Coding - CAVLC), CABAC, tạo mã số học nhị phân thích ứng ngữ cảnh dựa trên cú pháp (Syntax-based context-adaptive Binary Arithmetic Coding - SBAC), tạo mã entropy phân chia khoảng xác suất (Probability Interval Partitioning Entropy - PIPE), hoặc kỹ thuật tạo mã entropy khác. Sau khi tạo mã entropy, thì luồng bit được tạo mã có thể được phát đến thiết bị khác (ví dụ, bộ giải mã video) hoặc được lưu trữ để phát hoặc truy hồi về sau.

Fig.3 là hình vẽ sơ lược minh họa khối của bộ mã hóa video 300 ví dụ. Bộ mã hóa video 300 có thể được sử dụng để thực hiện các chức năng mã hóa của hệ thống codec 200 và/hoặc thực hiện các bước 101, 103, 105, 107, và/hoặc 109 của phương pháp vận hành 100. Bộ mã hóa 300 phân chia tín hiệu video đầu vào, để thu được tín hiệu video được phân chia 301, mà về cơ bản là tương tự như tín hiệu video được phân chia 201. Sau đó, tín hiệu video

được phân chia 301 được nén và được mã hóa thành luồng bit bởi các thành phần của bộ mã hóa 300.

Cụ thể là, tín hiệu video được phân chia 301 được chuyển tiếp đến thành phần dự đoán nội ảnh 317 để nội dự đoán. Thành phần dự đoán nội ảnh 317 có thể về cơ bản là tương tự như thành phần ước lượng nội ảnh 215 và thành phần dự đoán nội ảnh 217. Tín hiệu video được phân chia 301 cũng được chuyển tiếp đến thành phần bù chuyển động 321 để liên dự đoán dựa trên các khối tham chiếu ở thành phần bộ đệm ảnh được giải mã 323. Thành phần bù chuyển động 321 có thể về cơ bản là tương tự như thành phần ước lượng chuyển động 221 và thành phần bù chuyển động 219. Các khối dự đoán và các khối dư từ thành phần dự đoán nội ảnh 317 và thành phần bù chuyển động 321 được chuyển tiếp đến thành phần biến đổi và lượng tử hóa 313 để biến đổi và lượng tử hóa các khối dư. Thành phần biến đổi và lượng tử hóa 313 có thể về cơ bản là tương tự như thành phần biến đổi, thay đổi tỷ lệ và lượng tử hóa 213. Các khối dư đã được biến đổi và lượng tử hóa và các khối dự đoán tương ứng (cùng với dữ liệu điều khiển liên quan) được chuyển tiếp đến thành phần tạo mã entropy 331 để tạo mã vào luồng bit. Thành phần tạo mã entropy 331 có thể về cơ bản là tương tự như thành phần định dạng phân đầu và CABAC 231.

Các khối dư đã được biến đổi và lượng tử hóa và/hoặc các khối dự đoán tương ứng cũng được chuyển tiếp từ thành phần biến đổi và lượng tử hóa 313 đến thành phần biến đổi và lượng tử hóa ngược 329 để tái tạo thành các khối tham chiếu để thành phần bù chuyển động 321 sử dụng. Thành phần biến đổi và lượng tử hóa ngược 329 có thể về cơ bản là tương tự như thành phần thay đổi tỷ lệ và biến đổi ngược 229. Các bộ lọc trong vòng lặp ở thành phần bộ lọc trong vòng lặp 325 cũng được áp dụng cho các khối dư và/hoặc các khối tham chiếu tái tạo được, tùy theo ví dụ. Thành phần bộ lọc trong vòng lặp 325 có thể về cơ bản là tương tự như thành phần phân tích điều khiển bộ lọc 227 và thành phần bộ lọc trong vòng lặp 225. Thành phần bộ lọc trong vòng lặp 325 có thể bao gồm nhiều bộ lọc như được mô tả đối với thành phần bộ lọc trong vòng lặp 225. Sau đó, các khối được lọc được lưu trữ ở thành phần bộ đệm ảnh được giải mã 323 để thành phần bù chuyển động 321 sử dụng làm các khối tham chiếu. Thành phần bộ đệm ảnh được giải mã 323 có thể về cơ bản là tương tự như thành phần bộ đệm ảnh được giải mã 223.

Fig.4 là hình vẽ sơ lược minh họa khối của bộ giải mã video 400 ví dụ. Bộ giải mã video 400 có thể được sử dụng để thực hiện các chức năng giải mã của hệ thống codec 200 và/hoặc thực hiện các bước 111, 113, 115, và/hoặc 117 của phương pháp vận hành 100. Bộ

giải mã 400 thu luồng bit, ví dụ, từ bộ mã hóa 300, và tạo ra tín hiệu video đầu ra được tái tạo dựa trên luồng bit này để hiển thị cho người dùng cuối.

Luồng bit này được thu bởi thành phần giải mã entropy 433. Thành phần giải mã entropy 433 được tạo cấu hình để thực hiện sơ đồ giải mã entropy, chẳng hạn CAVLC, CABAC, SBAC, tạo mã PIPE, hoặc các kỹ thuật tạo mã entropy khác. Ví dụ, thành phần giải mã entropy 433 có thể sử dụng thông tin phân đầu để cung cấp ngữ cảnh để diễn dịch dữ liệu bổ sung mà được mã hóa dưới dạng các từ mã trong luồng bit. Thông tin được giải mã bao gồm thông tin mong muốn bất kỳ để giải mã tín hiệu video, chẳng hạn dữ liệu điều khiển tổng, dữ liệu điều khiển bộ lọc, thông tin về phân chia, dữ liệu chuyển động, dữ liệu dự đoán, và các hệ số biến đổi đã được lượng tử hóa từ các khối dư. Các hệ số biến đổi đã được lượng tử hóa được chuyển tiếp đến thành phần biến đổi và lượng tử hóa ngược 429 để tái tạo thành các khối dư. Thành phần biến đổi và lượng tử hóa ngược 429 có thể tương tự như thành phần biến đổi và lượng tử hóa ngược 329.

Các khối dư được tái tạo và/hoặc các khối dự đoán được chuyển tiếp đến thành phần dự đoán nội ảnh 417 để tái tạo thành các khối ảnh dựa trên các hoạt động nội dự đoán. Thành phần dự đoán nội ảnh 417 có thể tương tự như thành phần ước lượng nội ảnh 215 và thành phần dự đoán nội ảnh 217. Cụ thể là, thành phần dự đoán nội ảnh 417 sử dụng các chế độ dự đoán để định vị khối tham chiếu trong khung và áp dụng khối dư cho kết quả để tái tạo các khối ảnh được nội dự đoán. Các khối hình ảnh được nội dự đoán tái tạo được và/hoặc các khối dư và dữ liệu liên dự đoán tương ứng được chuyển tiếp đến thành phần bộ đệm ảnh được giải mã 423 thông qua thành phần bộ lọc trong vòng lặp 425, mà có thể về cơ bản là tương tự lần lượt như thành phần bộ đệm ảnh được giải mã 223 và thành phần bộ lọc trong vòng lặp 225. Thành phần bộ lọc trong vòng lặp 425 lọc các khối hình ảnh được tái tạo, các khối dư và/hoặc các khối dự đoán, và thông tin đó được lưu trữ trong thành phần bộ đệm ảnh được giải mã 423. Các khối ảnh được tái tạo đến từ thành phần bộ đệm ảnh được giải mã 423 là được chuyển tiếp đến thành phần bù chuyển động 421 để liên dự đoán. Thành phần bù chuyển động 421 có thể về cơ bản là tương tự như thành phần ước lượng chuyển động 221 và/hoặc thành phần bù chuyển động 219. Cụ thể là, thành phần bù chuyển động 421 sử dụng các vector chuyển động từ khối tham chiếu để tạo ra khối dự đoán và áp dụng khối dư cho kết quả để tái tạo khối ảnh. Các khối được tái tạo thu được cũng có thể được chuyển tiếp qua thành phần bộ lọc trong vòng lặp 425 đến thành phần bộ đệm ảnh được giải mã 423. Thành phần bộ đệm ảnh được giải mã 423 tiếp tục lưu trữ thêm các khối ảnh tái tạo được,

mà có thể được tái tạo thành các khung thông qua thông tin về phân chia. Các khung đó cũng có thể được đặt vào chuỗi. Chuỗi này được xuất về phía thiết bị hiển thị dưới dạng tín hiệu video đầu ra tái tạo được.

Fig.5 là hình vẽ sơ lược minh họa luồng bit ví dụ 500 và luồng bit con 501 được trích xuất từ luồng bit 500. Ví dụ, luồng bit 500 có thể được tạo ra bởi hệ thống codec 200 và/hoặc bộ mã hóa 300 để giải mã bởi hệ thống codec 200 và/hoặc bộ giải mã 400. Theo ví dụ khác, thì luồng bit 500 có thể được tạo ra bởi bộ mã hóa ở bước 109 của phương pháp 100 để bộ giải mã sử dụng ở bước 111.

Luồng bit 500 này bao gồm tập hợp thông số chuỗi (Sequence Parameter Set - SPS) 510, các tập hợp thông số ảnh (Picture Parameter Set - PPS) 512, các phần đầu lát 514, dữ liệu ảnh 520, và một hoặc nhiều thông điệp SEI, SPS 510 chứa dữ liệu chuỗi chung cho tất cả các ảnh trong chuỗi video được chứa trong luồng bit 500 này. Dữ liệu như vậy có thể bao gồm việc tạo kích thước ảnh, chiều sâu bit, các tham số công cụ tạo mã, các hạn chế tốc độ bit, v.v.. PPS 512 chứa các tham số mà là cụ thể cho một hoặc nhiều ảnh tương ứng. Do đó, mỗi ảnh trong chuỗi video có thể là để chỉ một PPS 512. PPS 512 có thể chỉ thị các công cụ tạo mã sẵn có trong các ô trong các ảnh tương ứng, các tham số lượng tử hóa, các độ dịch vị, các tham số công cụ tạo mã dành riêng cho ảnh (ví dụ, các điều khiển bộ lọc), v.v.. Phần đầu lát 514 chứa các tham số mà là cụ thể cho một hoặc nhiều lát tương ứng 524 trong ảnh. Do đó, mỗi lát 524 trong chuỗi video có thể là để chỉ phần đầu lát 514. Phần đầu lát 514 có thể chứa thông tin về loại lát, các số đếm thứ tự ảnh (Picture Order Count - POC), các danh sách ảnh tham chiếu, các trọng số dự đoán, các điểm vào của ô, các thông số khử khối, v.v.. Trong một số ví dụ, các lát 524 có thể là để chỉ các nhóm ô. Trong trường hợp như vậy, phần đầu lát 514 có thể là để chỉ phần đầu nhóm ô. Các thông điệp SEI 515 là các thông điệp tùy chọn mà chứa siêu dữ liệu mà không được yêu cầu cho việc giải mã khối, nhưng có thể được sử dụng cho các mục đích liên quan chẳng hạn như chỉ thị việc định thời xuất ảnh, các thiết đặt hiển thị, việc phát hiện mất mát, che dấu mất mát, v.v..

Dữ liệu ảnh 520 chứa dữ liệu video được mã hóa theo việc liên dự đoán và/hoặc nội dự đoán cũng như dữ liệu dư đã được biến đổi và lượng tử hóa tương ứng. Dữ liệu ảnh 520 đó được sắp xếp theo việc phân chia mà được dùng để phân chia hình ảnh trước khi mã hóa. Ví dụ, chuỗi video được chia thành các ảnh 521. Các ảnh 521 có thể được chia thêm thành các ảnh con 522, mà được chia thành các lát 524. Các lát 524 có thể được chia thêm thành các ô và/hoặc các CTU. Các CTU được chia tiếp thành các khối tạo mã dựa trên các cây tạo

mã. Sau đó, các khối tạo mã này có thể được mã hóa/giải mã theo các cơ chế dự đoán. Ví dụ, ảnh 521 có thể chứa một hoặc nhiều ảnh con 522. Ảnh con 522 có thể chứa một hoặc nhiều lát 524. Ảnh 521 là để chỉ PPS 512 và các lát 524 là để chỉ phần đầu lát 514. Các ảnh con 522 có thể được phân chia một cách nhất quán trên toàn bộ chuỗi video (cũng được gọi là đoạn), và do đó có thể là để chỉ SPS 510. Mỗi lát 524 thể chứa một hoặc nhiều ô. Mỗi lát 524, và do đó ảnh 521 và ảnh con 522, cũng có thể chứa nhiều CTU.

Mỗi ảnh 521 có thể chứa toàn bộ tập hợp của dữ liệu trực quan được liên kết với chuỗi video cho thời điểm tương ứng trong thời gian. Tuy nhiên, các ứng dụng nhất định có thể muốn hiển thị chỉ một phần của ảnh 521 trong một số trường hợp. Ví dụ, hệ thống thực tế ảo (virtual reality - VR) có thể hiển thị vùng được lựa chọn bởi người dùng của ảnh 521, mà tạo ra cảm giác có mặt trong cảnh được mô tả trong ảnh 521. Vùng người dùng có thể muốn quan sát là không được biết khi luồng bit 500 được mã hóa. Theo đó, ảnh 521 có thể chứa mỗi vùng khả thi người dùng có thể quan sát như các ảnh con 522, mà có thể được giải mã và được hiển thị một cách riêng biệt dựa vào đầu vào người dùng. Các ứng dụng khác có thể hiển thị một cách riêng biệt vùng quan tâm. Ví dụ, tivi với ảnh trong ảnh có thể muốn hiển thị vùng cụ thể, và do đó ảnh con 522, từ một chuỗi video trên ảnh 521 của chuỗi video không liên quan. Trong một ví dụ khác nữa, các hệ thống hội nghị từ xa có thể hiển thị toàn bộ ảnh 521 của người dùng mà hiện đang nói và ảnh con 522 của người dùng mà hiện không đang nói. Theo đó, ảnh con 522 có thể chứa vùng được xác định của ảnh 521. Ảnh con 522 mà được ràng buộc chuyển động theo thời gian là có thể giải mã một cách riêng biệt từ phần còn lại của ảnh 521. Cụ thể là, ảnh con được ràng buộc chuyển động theo thời gian được mã hóa mà không tham chiếu tới các mẫu bên ngoài của ảnh con được ràng buộc chuyển động theo thời gian, và do đó chứa đủ thông tin để hoàn thành việc giải mã mà không tham chiếu tới phần còn lại của ảnh 521.

Mỗi lát 524 có thể là hình chữ nhật được xác định bởi CTU tại góc trên bên trái và CTU tại góc dưới cùng bên phải. Trong một số ví dụ, lát 524 bao gồm dãy các ô và/hoặc các CTU theo thứ tự quét mảnh xử lý từ trái sang phải và từ trên xuống dưới. Trong các ví dụ khác, lát 524 là lát hình chữ nhật. Lát hình chữ nhật có thể không duyệt toàn bộ chiều rộng của ảnh theo thứ tự quét mảnh. Thay vào đó, lát hình chữ nhật có thể chứa vùng hình chữ nhật và/hoặc hình vuông của ảnh 521 và/hoặc ảnh con 522 được xác định xét về các hàng CTU và/hoặc ô và các cột CTU và/hoặc ô. Lát 524 là đơn vị nhỏ nhất mà có thể được bộ giải

mã hiển thị một cách riêng biệt. Do đó, các lát 524 từ ảnh 521 có thể được chỉ định cho các ảnh con khác nhau 522 để thể hiện một cách riêng biệt các vùng mong muốn của ảnh 521.

Bộ giải mã có thể hiển thị một hoặc nhiều ảnh con 523 của ảnh 521. Các ảnh con 523 là nhóm nhỏ được xác định trước hoặc được lựa chọn theo người dùng của các ảnh con 522. Ví dụ, ảnh 521 có thể được chia thành chín ảnh con 522, nhưng bộ giải mã có thể chỉ hiển thị ảnh con đơn 523 từ nhóm của các ảnh con 522. Các ảnh con 523 chứa các lát 525, mà là nhóm con được xác định trước hoặc được lựa chọn của các lát 524. Để cho phép việc hiển thị riêng biệt của các ảnh con 523, luồng bit con 501 có thể được trích xuất 529 từ luồng bit 500. Việc trích xuất 529 có thể xảy ra trên phía bộ mã hóa sao cho bộ giải mã chỉ thu luồng bit con 501. Trong các trường hợp khác, toàn bộ luồng bit 500 được phát tới bộ giải mã và bộ giải mã trích xuất 529 luồng bit con 501 cho việc giải mã riêng biệt. Cần lưu ý rằng luồng bit con 501 cũng có thể được gọi chung là luồng bit trong một số trường hợp. Luồng bit con 501 bao gồm SPS 510, PPS 512, các ảnh con 523 được lựa chọn, cũng như các phần đầu lát 514, và các thông điệp SEI 515 mà liên quan tới các ảnh con 523 và/hoặc các lát 525.

Phần bộc lộ này báo hiệu các dữ liệu khác nhau để hỗ trợ việc tạo mã hiệu quả của các ảnh con 522 cho việc lựa chọn và hiển thị của các ảnh con 523 ở bộ giải mã. SPS 510 bao gồm kích thước ảnh con 531, vị trí ảnh con 532, và các ID ảnh con 533 có liên quan tới tập hợp đầy đủ của các ảnh con 522. Kích thước ảnh con 531 bao gồm chiều cao ảnh con trong các mẫu độ chói và chiều rộng ảnh con trong các mẫu độ chói cho ảnh con tương ứng 522. Vị trí ảnh con 532 bao gồm khoảng cách dịch vị giữa mẫu phía trên cùng bên trái của ảnh con tương ứng 522 và mẫu trên cùng bên trái của ảnh 521. Vị trí ảnh con 532 và kích thước ảnh con 531 xác định bố cục của ảnh con 522 tương ứng. ID ảnh con 533 chứa dữ liệu mà nhận dạng duy nhất ảnh con 522 tương ứng. ID ảnh con 533 có thể là chỉ số quét mảnh của ảnh con 522 hoặc giá trị được xác định khác. Do đó, bộ giải mã có thể đọc SPS 510 và xác định kích thước, vị trí, và ID của mỗi ảnh con 522. Trong một số hệ thống tạo mã video, dữ liệu có liên quan tới các ảnh con 522 có thể được bao gồm trong PPS 512 bởi vì ảnh con 522 được phân chia từ ảnh 521. Tuy nhiên, các phần chia được sử dụng để tạo ra các ảnh con 522 có thể được sử dụng bởi các ứng dụng, chẳng hạn như các ứng dụng trên cơ sở ROI, các ứng dụng VR, v.v., mà phụ thuộc vào các phần chia ảnh con 522 nhất quán trên đoạn/chuỗi video. Theo đó, các phần chia ảnh con 522 thường không thay đổi trên cơ sở mỗi ảnh. Việc đặt thông tin bố cục cho các ảnh con 522 trong SPS 510 đảm bảo rằng bố cục chỉ được báo hiệu một lần cho chuỗi/đoạn thay vì được báo hiệu một cách dư thừa cho mỗi PPS

512 (mà có thể được báo hiệu cho mỗi ảnh 521 trong một số trường hợp). Ngoài ra, việc báo hiệu thông tin ảnh con 522, thay vì phụ thuộc vào bộ giải mã để suy ra thông tin như vậy, làm giảm xác suất lỗi trong trường hợp các gói bị mất và hỗ trợ chức năng bổ sung xét về trích xuất các ảnh con 523. Theo đó, việc báo hiệu bố cục ảnh con 522 trong SPS 510 cải thiện chức năng của bộ mã hóa và/hoặc bộ giải mã.

SPS 510 cũng chứa các cờ các ảnh con được ràng buộc chuyển động 534 có liên quan tới tập hợp đầy đủ của các ảnh con 522. Các cờ các ảnh con được ràng buộc chuyển động 534 chỉ thị xem mỗi ảnh con 522 có phải là ảnh con được ràng buộc chuyển động theo thời gian hay không. Do đó, bộ giải mã có thể đọc các cờ các ảnh con được ràng buộc chuyển động 534 và xác định ảnh con nào trong số các ảnh con 522 có thể được trích xuất và được hiển thị một cách riêng biệt mà không phải giải mã các ảnh con khác 522. Điều này cho phép các ảnh con 522 được lựa chọn được tạo mã như các ảnh con ràng buộc chuyển động theo thời gian trong khi cho phép các ảnh con khác 522 được tạo mã mà không có các hạn chế như vậy cho hiệu suất mã hóa tăng.

Các ID ảnh con 533 cũng được bao gồm trong các phần đầu lát 514. Mỗi phần đầu lát 514 chứa dữ liệu liên quan tới tập hợp tương ứng của các lát 524. Theo đó, phần đầu lát 514 chứa chỉ các ID ảnh con 533 tương ứng với các lát 524 được liên kết với phần đầu lát 514. Theo đó, bộ giải mã có thể thu lát 524, nhận ID ảnh con 533 từ phần đầu lát 514, và xác định ảnh con 522 nào chứa lát 524. Bộ giải mã cũng có thể sử dụng ID ảnh con 533 từ phần đầu lát 514 để tương quan với dữ liệu có liên quan trong SPS 510. Theo đó, bộ giải mã có thể xác định làm cách nào định vị các ảnh con 522/523 và các lát 524/525 bằng việc đọc SPS 510 và các phần đầu lát 514 tương ứng. Điều này cho phép các ảnh con 523 và các lát 525 được giải mã ngay cả nếu một số ảnh con 522 bị mất trong khi phát được bỏ qua có chủ đích để làm tăng hiệu suất tạo mã.

Thông điệp SEI 515 cũng có thể chứa cấp độ ảnh con 535. Mức ảnh con 535 chỉ thị các tài nguyên phần cứng cần thiết để giải mã ảnh con tương ứng 522. Theo cách này, mỗi ảnh con 522 có thể được tạo mã một cách độc lập của các ảnh con khác 522. Điều này đảm bảo mỗi ảnh con 522 có thể được phân bổ lượng chính xác của các tài nguyên phần cứng ở bộ giải mã. Không có cấp độ ảnh con 535 như vậy, mỗi ảnh con 522 sẽ được phân bổ với các tài nguyên đủ để giải mã ảnh con 522 phức tạp nhất. Do đó, cấp độ ảnh con 535 ngăn bộ giải mã không phân bổ quá mức các tài nguyên phần cứng nếu các ảnh con 522 được liên kết với các yêu cầu tài nguyên phần cứng khác nhau.

Fig.6 là hình vẽ sơ lược minh họa ảnh ví dụ 600 được phân chia thành các ảnh con 622. Ví dụ, ảnh 600 có thể được mã hóa vào và được giải mã từ luồng bit 500, ví dụ bởi hệ thống codec 200, bộ mã hóa 300, và/hoặc bộ giải mã 400. Ngoài ra, hình ảnh 600 có thể được phân chia và/hoặc được bao gồm trong luồng bit con 501 để hỗ trợ việc mã hóa và giải mã theo phương pháp 100.

Ảnh 600 có thể gần như là tương tự với ảnh 521. Ngoài ra, ảnh 600 có thể được phân chia thành các ảnh con 622, mà gần như là tương tự với các ảnh con 522. Các ảnh con 622 mỗi trong số chúng bao gồm kích thước ảnh con 631, mà có thể được bao gồm trong luồng bit 500 như kích thước ảnh con 531. Kích thước ảnh con 631 bao gồm chiều rộng ảnh con 631a và chiều cao ảnh con 631b. Chiều rộng ảnh con 631a là chiều rộng của ảnh con tương ứng 622 trong các đơn vị của các mẫu độ chói. Chiều cao ảnh con 631b là chiều cao của ảnh con tương ứng 622 trong các đơn vị của các mẫu độ chói. Các ảnh con 622 mỗi trong số chúng bao gồm ID ảnh con 633, mà có thể được bao gồm trong luồng bit 500 như ID ảnh con 633. ID ảnh con 633 có thể là giá trị bất kỳ mà nhận dạng duy nhất mỗi ảnh con 622. Trong ví dụ được thể hiện, ID ảnh con 633 là chỉ số ảnh con 622. Các ảnh con 622 mỗi trong số chúng bao gồm vị trí 632, mà có thể được bao gồm trong luồng bit 500 như vị trí ảnh con 532. Vị trí 632 được thể hiện như độ dịch vị giữa mẫu trên cùng bên trái của ảnh con tương ứng 622 và mẫu trên cùng bên trái 642 của ảnh 600.

Ngoài ra như được thể hiện, một số ảnh con 622 có thể là các ảnh con ràng buộc chuyển động theo thời gian 634 và các ảnh con khác 622 có thể không. Trong ví dụ được thể hiện, ảnh con 622 với ID ảnh con 633 là năm là ảnh con được ràng buộc chuyển động theo thời gian 634. Điều này chỉ thị rằng ảnh con 622 được nhận dạng là năm được tạo mã mà không tham chiếu tới ảnh con 622 bất kỳ khác và có thể do đó được trích xuất và được giải mã một cách riêng biệt mà không xem xét đến dữ liệu từ các ảnh con khác 622. Chỉ thị trong đó các ảnh con 622 là các ảnh con ràng buộc chuyển động theo thời gian 634 có thể được báo hiệu trong luồng bit 500 trong các cờ các ảnh con được ràng buộc chuyển động 534.

Như được thể hiện, các ảnh con 622 có thể được ràng buộc để bao trùm ảnh 600 mà không có khoảng trống hoặc khoảng chồng lấp. Khoảng trống là vùng của ảnh 600 mà không được bao gồm trong ảnh con 622 bất kỳ. Khoảng chồng lấp là vùng của ảnh 600 mà được bao gồm trong nhiều hơn một ảnh con 622. Trong ví dụ được thể hiện trên Fig.6, các ảnh con 622 được phân chia từ ảnh 600 để ngăn ngừa cả các khoảng trống và các khoảng chồng lấp. Các khoảng trống khiến các mẫu ảnh 600 bị bỏ sót trong các ảnh con 622. Các khoảng

chồng lấp khiến các lát được kết hợp được bao gồm trong nhiều ảnh con 622. Do đó, các khoảng trống và các khoảng chồng lấp có thể khiến các mẫu bị tác động bởi các xử lý khác nhau khi các ảnh con 622 được tạo mã một cách khác nhau. Nếu điều này được cho phép ở bộ mã hóa, bộ giải mã phải hỗ trợ sơ đồ tạo mã như vậy ngay cả khi sơ đồ giải mã hiếm khi được sử dụng. Bằng việc không cho phép ảnh con 622 các khoảng trống và các khoảng chồng lấp, độ phức tạp của bộ giải mã có thể được giảm do bộ giải mã không được yêu cầu phải tính đến các khoảng trống và các khoảng chồng lấp tiềm năng khi xác định các kích thước 631 và các vị trí 632 ảnh con. Ngoài ra, việc không cho phép các khoảng trống và các khoảng chồng lấp ảnh con 622 làm giảm độ phức tạp của các tiến trình RDO ở bộ mã hóa. Đó là bởi vì bộ mã hóa có thể bỏ qua việc xem xét các trường hợp khoảng trống và khoảng chồng lấp khi lựa chọn việc mã hóa cho chuỗi video. Theo đó, việc tránh các khoảng trống và các khoảng chồng lấp có thể làm giảm việc sử dụng của các tài nguyên bộ nhớ và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Fig.7 là hình vẽ sơ lược minh họa cơ chế ví dụ 700 cho các lát liên quan 724 tới bộ cục ảnh con 722. Ví dụ, cơ chế 700 có thể được áp dụng cho ảnh 600. Ngoài ra, cơ chế 700 có thể được áp dụng dựa vào dữ liệu trong luồng bit 500, ví dụ bởi hệ thống codec 200, bộ mã hóa 300, và/hoặc bộ giải mã 400. Ngoài ra, cơ chế 700 có thể được sử dụng để hỗ trợ việc mã hóa và giải mã theo phương pháp 100.

Cơ chế 700 có thể được áp dụng cho các lát 724 trong ảnh con 722, chẳng hạn như các lát 524/525 và các ảnh con 522/523, một cách tương ứng. Trong ví dụ được thể hiện, ảnh con 722 bao gồm lát thứ nhất 724a, lát thứ hai 724b, và lát thứ ba 724c. Các phần đầu lát cho mỗi trong số các lát 724 bao gồm ID ảnh con 733 đối với ảnh con 722. Bộ giải mã có thể so khớp ID ảnh con 733 từ phần đầu lát với ID ảnh con 733 trong SPS. Bộ giải mã có thể sau đó xác định vị trí 732 và kích thước của ảnh con 722 từ SPS dựa vào ID ảnh con 733. Sử dụng vị trí 732, ảnh con 722 có thể được đặt tương đối với mẫu trên cùng bên trái ở góc trên cùng bên trái 742 của ảnh. Kích thước có thể được sử dụng để thiết đặt chiều cao và chiều rộng của ảnh con 722 so với vị trí 732. Các lát 724 có thể sau đó được bao gồm trong ảnh con 722. Theo đó, các lát 724 có thể được định vị trong ảnh con 722 chính xác dựa vào ID ảnh con 733 mà không tham chiếu tới các ảnh con khác. Điều này hỗ trợ việc hiệu chỉnh lỗi do các ảnh con bị mất khác không thay thế việc giải mã của ảnh con 722. Điều này cũng hỗ trợ các ứng dụng mà chỉ trích xuất ảnh con 722 và tránh việc phát các ảnh con khác. Do đó, các ID ảnh con 733 hỗ trợ chức năng được nâng cao và/hoặc hiệu suất mã hóa tăng, mà làm

giảm việc sử dụng của các tài nguyên mạng, các tài nguyên bộ nhớ, và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Fig.8 là hình vẽ sơ lược minh họa ảnh ví dụ khác 800 được phân chia thành các ảnh con 822. Ảnh 800 có thể gần như là tương tự với ảnh 600. Ngoài ra, ảnh 800 có thể được mã hóa vào và được giải mã từ luồng bit 500, ví dụ, bởi hệ thống codec 200, bộ mã hóa 300, và/hoặc bộ giải mã 400. Ngoài ra, ảnh 800 có thể được phân chia và/hoặc được bao gồm trong luồng bit con 501 để hỗ trợ mã hóa và giải mã theo phương pháp 100 và/hoặc cơ chế 700.

Ảnh 800 bao gồm các ảnh con 822, mà có thể gần như là tương tự với các ảnh con 522, 523, 622, và/hoặc 722. Các ảnh con 822 được chia thành nhiều CTU 825. CTU 825 là đơn vị tạo mã cơ bản trong các hệ thống tạo mã video được tiêu chuẩn hóa. CTU 825 được chia nhỏ bởi cây tạo mã thành các khối tạo mã, mà được tạo mã theo việc liên dự báo hoặc việc nội dự báo. Như được thể hiện, một số ảnh con 822a được ràng buộc để bao gồm các chiều rộng ảnh con và các chiều cao ảnh con mà các bội của kích thước CTU 825. Trong ví dụ được thể hiện, các ảnh con 822a có chiều cao là sáu CTU 825 và chiều rộng là năm CTU 825. Ràng buộc này được loại bỏ đối với các ảnh con 822b được định vị trên ranh giới bên phải các ảnh 801 và cho các ảnh con 822c được định vị trên ranh giới phía dưới các ảnh 802. Trong ví dụ được thể hiện, các ảnh con 822b có chiều rộng là giữa năm và sáu CTU 825. Tuy nhiên, các ảnh con 822b mà không được định vị trên ranh giới phía dưới các ảnh 802 vẫn được ràng buộc để duy trì chiều cao ảnh con mà là bội của kích thước CTU 825. Trong ví dụ được thể hiện, các ảnh con 822c có chiều cao là giữa sáu và bảy CTU 825. Tuy nhiên, các ảnh con 822c mà không được định vị trên ranh giới bên phải các ảnh 801 vẫn được ràng buộc để duy trì chiều rộng ảnh con mà là bội của kích thước CTU 825.

Như được lưu ý ở trên, một số hệ thống video có thể giới hạn các ảnh con 822 để bao gồm các chiều cao và các chiều rộng mà các bội của kích thước CTU 825. Điều này có thể ngăn các ảnh con 822 không hoạt động một cách chính xác với nhiều bố cục ảnh, ví dụ với ảnh 800 mà chứa tổng chiều rộng hoặc chiều cao mà không phải là bội của kích thước CTU 825. Bằng việc cho phép các ảnh con ở dưới 822c và các ảnh con bên phải 822b bao gồm các chiều cao và các chiều rộng, một cách tương ứng, mà không phải là các bội của kích thước CTU 825, các ảnh con 822 có thể được sử dụng với ảnh 800 bất kỳ mà không gây ra các lỗi giải mã. Điều này dẫn đến việc làm tăng chức năng bộ mã hóa và bộ giải mã. Ngoài ra, chức năng được tăng cho phép bộ mã hóa tạo mã các ảnh một cách hiệu quả hơn, mà làm

giảm việc sử dụng của các tài nguyên mạng, các tài nguyên bộ nhớ, và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Như được mô tả ở đây, phần bộc lộ này mô tả các thiết kế cho việc phân chia ảnh trên cơ sở ảnh con trong việc tạo mã video. Ảnh con là vùng hình chữ nhật trong ảnh mà có thể được giải mã một cách độc lập sử dụng tiến trình giải mã tương tự như được sử dụng cho ảnh. Phần bộc lộ này liên quan đến việc báo hiệu của các ảnh con trong chuỗi video được tạo mã và/hoặc luồng bit cũng như tiến trình cho việc trích xuất ảnh con. Các phần mô tả của các kỹ thuật là dựa vào VVC bởi JVET của ITU-T và ISO/IEC. Tuy nhiên, các kỹ thuật cũng áp dụng cho các phần mô tả codec video khác. Dưới đây là các phương án ví dụ được mô tả ở đây. Các phương án như vậy có thể được áp dụng riêng lẻ hoặc kết hợp.

Thông tin có liên quan tới các ảnh con mà có thể có mặt trong chuỗi video được tạo mã (CVS) có thể được báo hiệu trong tập hợp thông số lớp chuỗi, chẳng hạn như SPS. Việc báo hiệu như vậy có thể bao gồm thông tin sau. Số lượng của các ảnh con mà có mặt trong mỗi ảnh của CVS có thể được báo hiệu trong SPS. Trong ngữ cảnh của SPS hoặc CVS, các ảnh con cùng vị trí cho tất cả các đơn vị truy cập (access unit - AU) có thể được gọi một cách chọn lọc là chuỗi ảnh con. Vòng cho thông tin quy định thêm mô tả các đặc tính của mỗi ảnh con cũng có thể được bao gồm trong SPS. Thông tin này có thể bao gồm việc nhận dạng ảnh con, vị trí của ảnh con (ví dụ, khoảng cách dịch vị giữa mẫu độ chói góc trên bên trái của ảnh con và mẫu độ chói góc trên bên trái của ảnh), và kích thước của ảnh con. Ngoài ra, SPS có thể báo hiệu xem mỗi ảnh con có phải là ảnh con được ràng buộc chuyển động (chứa chức năng của MCTS) hay không. Thông tin biên dạng, tầng, và cấp độ cho mỗi ảnh con cũng có thể được báo hiệu hoặc có thể dẫn ra ở bộ giải mã. Thông tin như vậy có thể được sử dụng để xác định thông tin biên dạng, tầng, và cấp độ cho luồng bit được tạo ra bằng việc trích xuất các ảnh con từ luồng bit ban đầu. Biên dạng và tầng của mỗi ảnh con có thể được suy ra để giống như biên dạng và tầng của toàn bộ luồng bit. Mức cho mỗi ảnh con có thể được báo hiệu một cách tường minh. Việc báo hiệu như vậy có thể có mặt trong vòng được chứa trong SPS. Các thông số bộ giải mã tham chiếu giả định cấp độ chuỗi (hypothetical reference decoder - HRD) có thể được báo hiệu trong phần thông tin khả năng sử dụng video (video usability information - VUI) của SPS cho mỗi ảnh con (hoặc một cách tương đương, mỗi chuỗi ảnh con).

Khi ảnh không được phân chia thành hai hoặc nhiều ảnh con, các đặc tính của ảnh con (ví dụ, vị trí, kích thước, v.v.), ngoại trừ ID ảnh con, có thể không có mặt / được báo

hiệu trong luồng bit. Khi ảnh con của các ảnh trong CVS được trích xuất, mỗi đơn vị truy cập trong luồng bit mới có thể không chứa các ảnh con. Trong trường hợp này, ảnh trong mỗi AU trong luồng bit mới không được phân chia thành nhiều ảnh con. Do đó không cần phải báo hiệu các đặc tính ảnh con chẳng hạn như vị trí và kích thước trong SPS do thông tin như vậy có thể được suy ra từ các đặc tính ảnh. Tuy nhiên, việc nhận dạng ảnh con có thể vẫn được báo hiệu do ID có thể được tham chiếu đến bởi các đơn vị NAL VCL / các nhóm ô mà được bao gồm trong ảnh con được trích xuất. Điều này cho phép các ID ảnh con duy trì như cũ khi trích xuất ảnh con.

Vị trí của ảnh con trong ảnh (độ dịch vị x và độ dịch vị y) có thể được báo hiệu trong các đơn vị của các mẫu độ chói. Vị trí thể hiện khoảng cách giữa mẫu độ chói góc trên bên trái của ảnh con và mẫu độ chói góc phía trên cùng bên trái của ảnh. Mặt khác, vị trí của ảnh con trong ảnh có thể được báo hiệu trong các đơn vị của kích thước khối độ chói tạo mã nhỏ nhất (MinCbSizeY). Mặt khác, đơn vị của vị trí ảnh con các độ dịch vị có thể được chỉ thị một cách tường minh bởi phần tử cú pháp trong tập hợp thông số. Đơn vị có thể là CtbSizeY, MinCbSizeY, mẫu độ chói, hoặc các giá trị khác.

Kích thước của ảnh con (chiều rộng ảnh con và chiều cao ảnh con) có thể được báo hiệu trong các đơn vị của các mẫu độ chói. Mặt khác, kích thước của ảnh con có thể được báo hiệu trong các đơn vị của kích thước khối độ chói tạo mã nhỏ nhất (MinCbSizeY). Mặt khác, đơn vị của các giá trị kích thước ảnh con có thể được chỉ thị một cách tường minh bởi phần tử cú pháp trong tập hợp thông số. Đơn vị có thể là CtbSizeY, MinCbSizeY, mẫu độ chói, hoặc các giá trị khác. Khi ranh giới bên phải của ảnh con không trùng với ranh giới bên phải của ảnh, chiều rộng của ảnh con có thể được yêu cầu là bội số nguyên của kích thước CTU độ chói (CtbSizeY). Tương tự, khi ranh giới phía dưới của ảnh con không trùng với ranh giới phía dưới của ảnh, chiều cao của ảnh con có thể được yêu cầu là bội số nguyên của kích thước CTU độ chói (CtbSizeY). Nếu chiều rộng của ảnh con không phải là bội số nguyên của kích thước CTU độ chói, ảnh con có thể được yêu cầu được nằm ở vị trí ngoài cùng bên phải trong ảnh. Tương tự, nếu chiều cao của ảnh con không phải là bội số nguyên của kích thước CTU độ chói, ảnh con có thể được yêu cầu được nằm ở phía dưới cùng trong ảnh. Trong một số trường hợp, chiều rộng của ảnh con có thể được báo hiệu trong các đơn vị của kích thước CTU độ chói, nhưng chiều rộng của ảnh con không phải là bội số nguyên của kích thước CTU độ chói. Trong trường hợp này, chiều rộng thực trong các mẫu độ chói có thể được suy ra dựa vào vị trí độ dịch vị của ảnh con. Chiều rộng của ảnh con có thể được

suy ra dựa vào kích thước CTU độ chói và chiều cao của ảnh có thể được suy ra dựa vào các mẫu độ chói. Tương tự, chiều cao của ảnh con có thể được báo hiệu trong các đơn vị của kích thước CTU độ chói, nhưng chiều cao của ảnh con không phải là bội số nguyên của kích thước CTU độ chói. Trong trường hợp như vậy, chiều cao thực trong mẫu độ chói có thể được suy ra dựa vào vị trí độ dịch vị của ảnh con. Chiều cao của ảnh con có thể được suy ra dựa vào kích thước CTU độ chói và chiều cao của ảnh có thể được suy ra dựa vào các mẫu độ chói.

Đối với ảnh con bất kỳ, ID ảnh con có thể là khác với chỉ số ảnh con. Chỉ số ảnh con có thể là chỉ số của ảnh con như được báo hiệu trong vòng của các ảnh con trong SPS. ID ảnh con có thể là chỉ số của ảnh con trong thứ tự quét mảnh của ảnh con trong ảnh. Khi giá trị của ID ảnh con của mỗi ảnh con là giống như chỉ số ảnh con, ID ảnh con có thể được báo hiệu hoặc được suy ra. Khi ID ảnh con của mỗi ảnh con là khác với chỉ số ảnh con, ID ảnh con được báo hiệu một cách tường minh. Số lượng của các bit để báo hiệu của các ID ảnh con có thể được báo hiệu trong cùng tập hợp thông số mà chứa các đặc tính ảnh con (ví dụ, trong SPS). Một số giá trị cho ID ảnh con có thể được dự trữ cho các mục đích nhất định. Ví dụ, khi các phần đầu nhóm ô chứa các ID ảnh con để quy định ảnh con nào chứa nhóm ô, giá trị không có thể được dự trữ và không được sử dụng cho các ảnh con để đảm bảo rằng một vài bit đầu tiên của phần đầu nhóm ô không phải đều là không để tránh sự bao gồm tình cờ của mã chống giả lập. Trong các trường hợp tùy chọn trong đó các ảnh con của ảnh không bao trùm toàn bộ diện tích của ảnh mà không có khoảng hở và không có khoảng chồng lấp, giá trị (ví dụ, giá trị một) có thể được dự trữ cho các nhóm ô mà không phải một phần của ảnh con bất kỳ. Mặt khác, ID ảnh con của diện tích còn lại được báo hiệu một cách tường minh. Số lượng của các bit để báo hiệu ID ảnh con có thể được ràng buộc như dưới đây. Khoảng giá trị nên đủ để nhận dạng duy nhất tất cả các ảnh con trong ảnh, bao gồm các giá trị dự trữ của ID ảnh con. Ví dụ, số lượng tối thiểu của các bit cho ID ảnh con có thể là giá trị của $\text{Ceil}(\log_2(\text{số lượng của các ảnh con trong ảnh} + \text{số lượng của ID ảnh con được dự trữ}))$.

Điều có thể được ràng buộc là sự kết hợp của các ảnh con phải bao trùm toàn bộ ảnh mà không có khoảng hở và không có khoảng chồng lấp. Khi ràng buộc này được áp dụng, cho mỗi ảnh con, cờ có thể có mặt để quy định xem ảnh con có phải là ảnh con được ràng buộc chuyển động hay không, mà chỉ thị ảnh con có thể được trích xuất. Mặt khác, sự kết

hợp của các ảnh con có thể không bao trùm toàn bộ ảnh, nhưng các khoảng chồng lấp có thể không được cho phép.

ID các ảnh con có thể có mặt ngay sau phần đầu đơn vị NAL để hỗ trợ tiến trình trích xuất ảnh con mà không yêu cầu bộ trích xuất phân tích phần còn lại của các bit đơn vị NAL. Đối với các đơn vị NAL VCL, ID ảnh con có thể có mặt trong các bit thứ nhất của các phần đầu nhóm ô. Đối với đơn vị NAL không phải VCL, phần sau đây có thể áp dụng. Đối với SPS, ID ảnh con không cần phải có mặt ngay sau phần đầu đơn vị NAL. Đối với PPS, nếu tất cả các nhóm ô của cùng ảnh được ràng buộc tham chiếu đến cùng PPS, ID ảnh con không cần phải có mặt ngay sau phần đầu đơn vị NAL của nó. Nếu các nhóm ô của cùng ảnh được cho phép để tham chiếu tới các PPS khác nhau, ID ảnh con có thể có mặt trong các bit thứ nhất của PPS (ví dụ, ngay sau phần đầu đơn vị NAL). Trong trường hợp này, các nhóm ô bất kỳ của một ảnh có thể được cho phép chia sẻ cùng PPS. Mặt khác, khi các nhóm ô của cùng ảnh được cho phép tham chiếu tới các PPS khác nhau, và nhóm ô khác của cùng ảnh cũng được cho phép chia sẻ cùng PPS, không có ID ảnh con nào có thể có mặt trong cú pháp PPS. Mặt khác, khi các nhóm ô của cùng ảnh được cho phép để tham chiếu tới các PPS khác nhau, và nhóm ô khác của cùng ảnh cũng được cho phép chia sẻ cùng PPS, danh sách của các ID ảnh con có thể có mặt trong cú pháp PPS. Danh sách chỉ thị các ảnh con mà cho chúng PPS áp dụng. Đối với các đơn vị NAL không phải VCL khác, nếu đơn vị không phải VCL áp dụng cho cấp độ ảnh hoặc cao hơn (ví dụ, ký hiệu phân cách đơn vị truy cập, kết thúc chuỗi, kết thúc luồng bit, v.v.), thì ID ảnh con có thể không có mặt ngay sau phần đầu đơn vị NAL. Nếu không, ID ảnh con có thể có mặt ngay sau phần đầu đơn vị NAL.

Với việc báo hiệu SPS nêu trên, việc phân chia ô trong các ảnh con riêng lẻ có thể được báo hiệu trong PPS. Các nhóm ô trong cùng ảnh có thể được cho phép để liên hệ tới các PPS khác nhau. Trong trường hợp này, việc nhóm ô có thể chỉ là trong mỗi ảnh con. Khái niệm việc nhóm ô là việc phân chia của ảnh con thành các ô.

Mặt khác, tập hợp thông số để mô tả việc phân chia ô trong các ảnh con riêng lẻ được xác định. Tập hợp thông số như vậy có thể được gọi là tập hợp thông số ảnh con (Sub-Picture Parameter Set - SPPS). SPPS là để chỉ SPS. Phần tử cú pháp tham chiếu đến SPS ID là có mặt trong SPPS. SPPS có thể chứa ID ảnh con. Nhằm các mục đích trích xuất ảnh con, phần tử cú pháp tham chiếu đến ID ảnh con là phần tử cú pháp thứ nhất trong SPPS. SPPS chứa cấu trúc ô (ví dụ, số lượng của các cột, số lượng của các hàng, việc đặt cách ô đồng đều, v.v.) SPPS có thể chứa cờ để chỉ thị xem bộ lọc vòng có được kích hoạt ở các đường biên ảnh con

được kết hợp hay không. Mặt khác, các đặc tính ảnh con cho mỗi ảnh con có thể được báo hiệu trong SPPS thay vì trong SPS. Việc phân chia ô trong các ảnh con riêng lẻ có thể vẫn được báo hiệu trong PPS. Các nhóm ô trong cùng ảnh được cho phép để tham chiếu tới các PPS khác nhau. Một khi SPPS được kích hoạt, SPPS duy trì cho chuỗi các AU liên tiếp theo thứ tự giải mã. Tuy nhiên, SPPS có thể được ngắt kích hoạt/được kích hoạt tại AU mà không phải là điểm bắt đầu của CVS. Tại thời điểm bất kỳ trong suốt tiến trình giải mã của luồng bit lớp đơn với nhiều ảnh con tại một số AU, nhiều SPPS có thể là hoạt động. SPPS có thể được chia sẻ bởi các ảnh con khác nhau của AU. Mặt khác, SPPS và PPS có thể được gộp thành một tập hợp thông số. Trong trường hợp như vậy, tất cả các nhóm ô của cùng ảnh có thể không được yêu cầu phải tham chiếu tới cùng PPS. Sự ràng buộc có thể được áp dụng sao cho tất cả các nhóm ô trong cùng ảnh con có thể là để chỉ cùng tập hợp thông số được tạo ra từ việc gộp giữa SPPS và PPS.

Số lượng của các bit được sử dụng để báo hiệu ID ảnh con có thể được báo hiệu trong phần đầu đơn vị NAL. Khi có mặt trong phần đầu đơn vị NAL thông tin như vậy có thể trợ giúp các tiến trình trích xuất ảnh con trong việc phân tích giá trị ID ảnh con tại điểm bắt đầu của phần tải hữu ích của đơn vị NAL (ví dụ, một vài bit thứ nhất ngay sau phần đầu đơn vị NAL). Đối với việc báo hiệu như vậy, một số bit dự trữ (ví dụ, bảy bit dự trữ) trong phần đầu đơn vị NAL có thể được sử dụng để tránh việc tăng chiều dài của phần đầu đơn vị NAL. Số lượng của các bit cho việc báo hiệu như vậy có thể bao trùm giá trị của sub-picture-ID-bit-len. Ví dụ, bốn bit trong số bảy bit dự trữ của phần đầu đơn vị NAL các VVC có thể được sử dụng cho mục đích này.

Khi giải mã ảnh con, vị trí của mỗi khối cây tạo mã (ví dụ, xCtb và yCtb) có thể được điều chỉnh cho vị trí mẫu độ chói thực trong ảnh thay vì vị trí mẫu độ chói trong ảnh con. Theo cách này, việc trích xuất của ảnh con cùng vị trí từ mỗi ảnh tham chiếu có thể được tránh do khối cây tạo mã được giải mã với việc tham chiếu đến ảnh thay vì ảnh con. Để điều chỉnh vị trí của khối cây tạo mã, các biến số SubpictureXOffset và SubpictureYOffset có thể được suy ra dựa vào vị trí ảnh con (subpic_x_offset và subpic_y_offset). Các giá trị của các biến số có thể được bổ sung vào các giá trị của các tọa độ x và y vị trí mẫu độ chói, một cách tương ứng, của mỗi khối cây tạo mã trong ảnh con.

Tiến trình trích xuất ảnh con có thể được xác định như dưới đây. Đầu vào tiến trình là ảnh con đích cần được trích xuất. Đầu vào có thể dưới dạng ID ảnh con hoặc vị trí ảnh con. Khi đầu vào là vị trí của ảnh con, ID ảnh con được kết hợp có thể được giải bằng việc

phân tích thông tin ảnh con trong SPS. Đối với các đơn vị NAL không phải VCL, phần sau đây được áp dụng. Các thành phần cú pháp trong SPS có liên quan tới kích thước và cấp độ ảnh có thể được cập nhật với thông tin kích thước và cấp độ của ảnh con. Các đơn vị NAL không phải VCL dưới đây được giữ không thay đổi: PPS, ký hiệu phân cách đơn vị truy cập (Access Unit Delimiter - AUD), kết thúc chuỗi (End of Sequence - EOS), kết thúc luồng bit (End of Bitstream - EOB), và bất kỳ các đơn vị NAL không phải VCL NAL khác mà có thể áp dụng cho cấp độ ảnh hoặc cao hơn. Các đơn vị NAL không phải VCL còn lại với ID ảnh con không bằng với ID ảnh con đích có thể được loại bỏ. Các đơn vị NAL VCL với ID ảnh con không bằng với ID ảnh con đích cũng có thể được loại bỏ.

Thông điệp SEI lồng ảnh con cấp độ chuỗi có thể được sử dụng cho việc lồng các thông điệp SEI cấp độ ảnh con hoặc cấp độ AU cho tập hợp của các ảnh con. Điều này có thể bao gồm khoảng đệm, định thời ảnh, và các thông điệp SEI không phải HRD. Cú pháp và các ngữ nghĩa của thông điệp SEI lồng ảnh con này có thể là như dưới đây. Đối với các sự vận hành các hệ thống, chẳng hạn như trong các môi trường định dạng phương tiện đa hướng (omnidirectional media format - OMAF), tập hợp của các chuỗi ảnh con bao trùm công nhìn có thể được yêu cầu và được giải mã bởi máy chơi OMAF. Do đó, thông điệp SEI cấp độ chuỗi được sử dụng để mang thông tin của tập hợp của các chuỗi ảnh con mà bao trùm một cách chọn lọc vùng ảnh hình chữ nhật. Thông tin có thể được sử dụng bởi các hệ thống, và thông tin là chỉ thị của khả năng giải mã được yêu cầu cũng như tốc độ bit của tập hợp của các chuỗi ảnh con. Thông tin chỉ thị mức của luồng bit bao gồm chỉ tập hợp của các chuỗi ảnh con. Thông tin này cũng chỉ thị tốc độ bit của luồng bit chứa chỉ tập hợp của các chuỗi ảnh con. Tùy chọn là, tiến trình trích xuất luồng bit con có thể được quy định cho tập hợp của các chuỗi ảnh con. Lợi ích của việc làm này là luồng bit bao gồm chỉ tập hợp của các chuỗi ảnh con cũng có thể phù hợp. Nhược điểm là ở chỗ trong việc xem xét các khả năng kích thước công nhìn khác nhau có thể có nhiều tập hợp như vậy bên cạnh các số lượng khả thi đã lớn sẵn của các chuỗi ảnh con riêng lẻ.

Trong phương án ví dụ, một hoặc nhiều ví dụ được bộc lộ có thể được thực hiện như dưới đây. Ảnh con có thể được xác định như vùng hình chữ nhật của một hoặc nhiều nhóm ô trong ảnh. Tiến trình tách nhị nhân được cho phép có thể được xác định như dưới đây. Các đầu vào tới tiến trình này là: chế độ chia nhị phân btSplit, chiều dài khối tạo mã cbWidth, chiều cao khối tạo mã cbHeight, vị trí (x0, y0) của mẫu độ chói phía trên cùng bên trái của mẫu tạo mã được xem xét so với mẫu độ chói phía trên cùng bên trái của ảnh, chiều sâu cây

nhiều loại `mttDepth`, chiều sâu cây nhiều loại lớn nhất với độ dịch vị `maxMttDepth`, kích thước cây nhị phân lớn nhất `maxBtSize`, và chỉ số phân chia `partIdx`. Đầu ra của tiến trình này là biến số `allowBtSplit`.

	<code>btSplit == SPLIT_BT_VER</code>	<code>btSplit == SPLIT_BT_HOR</code>
<code>parallelTtSplit</code>	<code>SPLIT_TT_VER</code>	<code>SPLIT_TT_HOR</code>
<code>cbSize</code>	<code>cbWidth</code>	<code>cbHeight</code>

Phần mô tả của `parallelTtSplit` và `cbSize` dựa vào `btSplit`

Các biến số `parallelTtSplit` và `cbSize` được suy ra như được quy định ở trên. Biến số `allowBtSplit` được suy ra như dưới đây. Nếu một hoặc nhiều điều kiện sau đây là đúng, `allowBtSplit` được thiết đặt bằng là SAI (FALSE): `cbSize` là nhỏ hơn hoặc bằng với `MinBtSizeY`, `cbWidth` là lớn hơn `maxBtSize`, `cbHeight` là lớn hơn `maxBtSize`, và `mttDepth` là lớn hơn hoặc bằng với `maxMttDepth`. Nếu không, nếu tất cả các điều kiện sau đây là đúng, `allowBtSplit` được thiết đặt bằng với FALSE: `btSplit` là bằng với `SPLIT_BT_VER`, và `y0 + cbHeight` là lớn hơn `SubPicBottomBorderInPic`. Nếu không, nếu tất cả các điều kiện sau đây là đúng, `allowBtSplit` được thiết đặt bằng với FALSE, `btSplit` là bằng với `SPLIT_BT_HOR`, `x0 + cbWidth` là lớn hơn `SubPicRightBorderInPic`, và `y0 + cbHeight` là nhỏ hơn hoặc bằng với `SubPicBottomBorderInPic`. Nếu không, nếu tất cả các điều kiện sau đây là đúng, `allowBtSplit` được thiết đặt bằng với FALSE: `mttDepth` là lớn hơn không, `partIdx` là bằng một, và `MttSplitMode[ x0 ][ y0 ][ mttDepth - 1 ]` là bằng với `parallelTtSplit`. Nếu không nếu tất cả các điều kiện sau đây là đúng, `allowBtSplit` được thiết đặt bằng với FALSE: `btSplit` là bằng với `SPLIT_BT_VER`, `cbWidth` là nhỏ hơn hoặc bằng với `MaxTbSizeY`, và `cbHeight` là lớn hơn `MaxTbSizeY`. Nếu không nếu tất cả các điều kiện sau đây là đúng, `allowBtSplit` được thiết đặt bằng với FALSE: `btSplit` là bằng với `SPLIT_BT_HOR`, `cbWidth` là lớn hơn `MaxTbSizeY`, và `cbHeight` là nhỏ hơn hoặc bằng với `MaxTbSizeY`. Nếu không, `allowBtSplit` được thiết đặt bằng với ĐÚNG (TRUE).

Tiến trình chia bậc ba được cho phép có thể được xác định như dưới đây. Các đầu vào tiến trình này là: chế độ chia bậc ba `ttSplit`, chiều dài khối tạo mã `cbWidth`, chiều cao khối tạo mã `cbHeight`, vị trí (`x0`, `y0`) của mẫu độ chói phía trên cùng bên trái của mẫu tạo mã được xem xét so với mẫu độ chói phía trên cùng bên trái của ảnh, chiều sâu cây nhiều loại `mttDepth`, chiều sâu cây nhiều loại lớn nhất với độ dịch vị `maxMttDepth`, và kích thước cây nhị phân lớn nhất `maxTtSize`. Đầu ra của tiến trình này là biến số `allowTtSplit`.

	ttSplit == SPLIT_TT_VER	ttSplit == SPLIT_TT_HOR
cbSize	cbWidth	cbHeight

Phần mô tả của cbSize dựa vào ttSplit.

Biến số cbSize được suy ra như được quy định ở trên. Biến số allowTtSplit được suy ra như dưới đây. Nếu một hoặc nhiều điều kiện sau đây là đúng, allowTtSplit được thiết đặt bằng với FALSE: cbSize là nhỏ hơn hoặc bằng với $2 * \text{MinTtSizeY}$, cbWidth là lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$, cbHeight là lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$, mttDepth là lớn hơn hoặc bằng với maxMttDepth, $x0 + \text{cbWidth}$ là lớn hơn SubPicRightBorderInPic, và $y0 + \text{cbHeight}$ là lớn hơn SubPicBottomBorderInPic. Nếu không, allowTtSplit được thiết đặt bằng với TRUE.

Các ngữ nghĩa và cú pháp RBSP tập hợp thông số chuỗi là như sau đây.

seq_parameter_set_rbsp() {	Bộ mô tả
sps_seq_parameter_set_id	ue(v)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
num_subpic_minus1	ue(v)
subpic_id_len_minus1	ue(v)
for (i = 0; i <= num_subpic_minus1; i++) {	
subpic_id[i]	u(v)
if(num_subpic_minus1 > 0) {	
subpic_level_idc[i]	u(8)
subpic_x_offset[i]	ue(v)
subpic_y_offset[i]	ue(v)
subpic_width_in_luma_samples[i]	ue(v)
subpic_height_in_luma_samples[i]	ue(v)
subpic_motion_constrained_flag[i]	u(1)
}	
}	
...	
}	

`pic_width_in_luma_samples` quy định chiều rộng của mỗi ảnh được giải mã trong các đơn vị của các mẫu độ chói. `pic_width_in_luma_samples` sẽ không bằng không và sẽ là bội số nguyên của `MinCbSizeY`. `pic_height_in_luma_samples` quy định chiều cao của mỗi ảnh được giải mã trong các đơn vị của các mẫu độ chói. `pic_height_in_luma_samples` sẽ không bằng không và sẽ là bội số nguyên của `MinCbSizeY`. `num_subpicture_minus1` cộng 1 quy định số lượng của các ảnh con được phân chia trong các ảnh được tạo mã thuộc về chuỗi video được tạo mã. `subpic_id_len_minus1` cộng 1 quy định số lượng của các bit được sử dụng để biểu thị phần tử cú pháp `subpic_id[ i ]` trong SPS, `spps_subpic_id` trong SPPS tham chiếu đến SPS, và `tile_group_subpic_id` trong các phần đầu nhóm ô tham chiếu đến SPS. Giá trị của `subpic_id_len_minus1` sẽ nằm trong khoảng từ $\text{Ceil}(\text{Log2}(\text{num_subpic_minus1} + 2))$ đến tám, không loại trừ. `subpic_id[ i ]` quy định ID ảnh con của ảnh con thứ i của các ảnh tham chiếu đến SPS. Chiều dài của `subpic_id[ i ]` là `subpic_id_len_minus1 + 1` bit. Giá trị của `subpic_id[i]` sẽ lớn hơn không. `subpic_level_idc[ i ]` chỉ thị cấp độ mà với nó CVS được tạo thành từ việc trích xuất của các ảnh con thứ i phù hợp với các yêu cầu tài nguyên được quy định. Các luồng bit không chứa các giá trị của `subpic_level_idc[ i ]` ngoài các giá trị đã được quy định. Các giá trị khác của `subpic_level_idc[ i ]` được dự trữ. Khi không có mặt, giá trị của `subpic_level_idc[ i ]` được suy ra là bằng với giá trị của `general_level_idc`.

`subpic_x_offset[ i ]` quy định độ dịch vị ngang của góc trên cùng bên trái của ảnh con thứ i so với góc trên cùng bên trái của ảnh. Khi không có mặt, giá trị của `subpic_x_offset[ i ]` được suy ra là bằng với 0. Giá trị của ảnh con độ dịch vị x được suy ra như dưới đây: $\text{SubpictureXOffset}[i] = \text{subpic_x_offset}[i]$. `subpic_y_offset[ i ]` quy định độ dịch vị thẳng của góc trên cùng bên trái của ảnh con thứ i so với góc trên cùng bên trái của ảnh. Khi không có mặt, giá trị của `subpic_y_offset[ i ]` được suy ra là bằng không. Giá trị của ảnh con độ dịch vị y được suy ra như dưới đây: $\text{SubpictureYOffset}[i] = \text{subpic_y_offset}[i]$. `subpic_width_in_luma_samples[ i ]` quy định chiều rộng của ảnh con được giải mã thứ i mà đối với nó SPS này là SPS hoạt động. Khi tổng của $\text{SubpictureXOffset}[i]$ và `subpic_width_in_luma_samples[ i ]` là nhỏ hơn `pic_width_in_luma_samples`, giá trị của `subpic_width_in_luma_samples[ i ]` sẽ là bội số nguyên của `CtbSizeY`. Khi không có mặt, giá trị của `subpic_width_in_luma_samples[ i ]` được suy ra là bằng với giá trị của `pic_width_in_luma_samples`. `subpic_height_in_luma_samples[ i ]` quy định chiều cao của ảnh con được giải mã thứ i mà đối với nó SPS này là SPS hoạt động. Khi tổng của

$\text{SubpictureYOffset}[i]$ và $\text{subpic_height_in_luma_samples}[i]$ là nhỏ hơn $\text{pic_height_in_luma_samples}$, giá trị của $\text{subpic_height_in_luma_samples}[i]$ sẽ là bội số nguyên của CtbSizeY . Khi không có mặt, giá trị của $\text{subpic_height_in_luma_samples}[i]$ được suy ra là bằng với giá trị của $\text{pic_height_in_luma_samples}$.

Đây là một yêu cầu của sự tương thích luồng bit rằng sự kết hợp của các ảnh con bao trùm toàn bộ diện tích của ảnh mà không có khoảng chồng lấp và khoảng trống. $\text{subpic_motion_constrained_flag}[i]$ bằng một quy định ảnh con thứ i là ảnh con được ràng buộc chuyển động theo thời gian. $\text{subpic_motion_constrained_flag}[i]$ bằng không quy định ảnh con thứ i có thể là hoặc có thể không phải là ảnh con được ràng buộc chuyển động theo thời gian. Khi không có mặt, giá trị của $\text{subpic_motion_constrained_flag}$ được suy ra là bằng không.

Các biến số $\text{SubpicWidthInCtbsY}$, $\text{SubpicHeightInCtbsY}$, SubpicSizeInCtbsY , $\text{SubpicWidthInMinCbsY}$, $\text{SubpicHeightInMinCbsY}$, $\text{SubpicSizeInMinCbsY}$, $\text{SubpicSizeInSamplesY}$, $\text{SubpicWidthInSamplesC}$, và $\text{SubpicHeightInSamplesC}$ được suy ra như dưới đây:

$$\text{SubpicWidthInLumaSamples}[i] = \text{subpic_width_in_luma_samples}[i]$$

$$\text{SubpicHeightInLumaSamples}[i] = \text{subpic_height_in_luma_samples}[i]$$

$$\text{SubPicRightBorderInPic}[i] = \text{SubpictureXOffset}[i] + \text{PicWidthInLumaSamples}[i]$$

$$\text{SubPicBottomBorderInPic}[i] = \text{SubpictureYOffset}[i] + \text{PicHeightInLumaSamples}[i]$$

$$\text{SubpicWidthInCtbsY}[i] = \text{Ceil}(\text{SubpicWidthInLumaSamples}[i] \div \text{CtbSizeY})$$

$$\text{SubpicHeightInCtbsY}[i] = \text{Ceil}(\text{SubpicHeightInLumaSamples}[i] \div \text{CtbSizeY})$$

$$\text{SubpicSizeInCtbsY}[i] = \text{SubpicWidthInCtbsY}[i] * \text{SubpicHeightInCtbsY}[i]$$

$$\text{SubpicWidthInMinCbsY}[i] = \text{SubpicWidthInLumaSamples}[i] / \text{MinCbSizeY}$$

$$\text{SubpicHeightInMinCbsY}[i] = \text{SubpicHeightInLumaSamples}[i] / \text{MinCbSizeY}$$

$$\text{SubpicSizeInMinCbsY}[i] = \text{SubpicWidthInMinCbsY}[i] * \text{SubpicHeightInMinCbsY}[i]$$

$$\text{SubpicSizeInSamplesY}[i] = \text{SubpicWidthInLumaSamples}[i] *$$

$$\text{SubpicHeightInLumaSamples}[i]$$

$$\text{SubpicWidthInSamplesC}[i] = \text{SubpicWidthInLumaSamples}[i] / \text{SubWidthC}$$

$\text{SubpicHeightInSamplesC}[i] = \text{SubpicHeightInLumaSamples}[i] / \text{SubHeightC}$

Các ngữ nghĩa và cú pháp RBSP tập hợp thông số ảnh con là như sau.

sub_pic_parameter_set_rbsp() {	Bộ mô tả
spps_subpic_id	u(v)
spps_subpic_parameter_set_id	ue(v)
spps_seq_parameter_set_id	ue(v)
single_tile_in_subpic_flag	u(1)
if(!single_tile_in_subpic_flag) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_tile_spacing_flag	u(1)
if(!uniform_tile_spacing_flag) {	
for(i = 0; i <= num_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
if(loop_filter_across_tiles_enabled_flag)	
loop_filter_across_subpic_enabled_flag	u(1)
rbsp_trailing_bits()	
}	

spps_subpic_id nhận dạng ảnh con mà SPPS thuộc về. Chiều dài của spps_subpic_id là subpic_id_len_minus1 + 1 bit. spps_subpic_parameter_set_id nhận dạng SPPS cho việc tham chiếu bởi các thành phần cú pháp khác. Giá trị của spps_subpic_parameter_set_id sẽ nằm trong khoảng từ không đến sáu mươi ba, không loại trừ. spps_seq_parameter_set_id quy định giá trị của sps_seq_parameter_set_id đối với SPS hoạt động. Giá trị của spps_seq_parameter_set_id sẽ nằm trong khoảng từ không đến mười lăm, không loại trừ. single_tile_in_subpic_flag bằng một quy định rằng chỉ có một ô trong mỗi ảnh con tham chiếu đến SPPS. single_tile_in_subpic_flag bằng không quy định rằng có nhiều hơn một ô trong mỗi ảnh con tham chiếu đến SPPS. num_tile_columns_minus1 cộng 1 quy định số lượng cột ô phân chia ảnh con. num_tile_columns_minus1 sẽ nằm trong khoảng từ không đến PicWidthInCtbsY[spps_subpic_id] - 1, không loại trừ. Khi không hiện hữu, thì giá trị

của `num_tile_columns_minus1` được suy ra là bằng không. `num_tile_rows_minus1` cộng 1 quy định số lượng hàng ô phân chia ảnh con. `num_tile_rows_minus1` sẽ nằm trong khoảng từ không đến `PicHeightInCtbsY[ spps_subpic_id ] - 1`, không loại trừ. Khi không hiện hữu, thì giá trị của `num_tile_rows_minus1` được suy ra là bằng không. Biến số `NumTilesInPic` được đặt bằng $(\text{num_tile_columns_minus1} + 1) * (\text{num_tile_rows_minus1} + 1)$.

Khi `single_tile_in_subpic_flag` là bằng không, `NumTilesInPic` sẽ lớn hơn không. `uniform_tile_spacing_flag` bằng một quy định rằng các đường biên cột ô và tương tự như vậy các đường biên hàng ô được phân bố đồng nhất trên ảnh con. `uniform_tile_spacing_flag` bằng không quy định rằng các đường biên cột ô và tương tự như vậy các đường biên hàng ô không được phân bố đồng nhất trên ảnh con nhưng được báo hiệu tường minh sử dụng các phần tử cú pháp `tile_column_width_minus1[i]` và `tile_row_height_minus1[i]`. Khi không có mặt, giá trị của `uniform_tile_spacing_flag` được suy ra là bằng một. `tile_column_width_minus1[i] + 1` quy định chiều rộng của cột ô thứ *i* với các đơn vị là các CTB. `tile_row_height_minus1[i] + 1` quy định chiều cao của hàng ô thứ *i* với các đơn vị là các CTB.

Các biến số sau đây được suy ra bằng cách gọi ra tiến trình chuyển đổi quét mảnh và ô CTB: danh sách `ColWidth[i]` đối với *i* nằm trong khoảng từ không đến `num_tile_columns_minus1`, không loại trừ, quy định chiều rộng của cột ô thứ *i* trong các đơn vị của CTB; danh sách `RowHeight[j]` đối với *j* nằm trong khoảng từ không đến `num_tile_rows_minus1`, không loại trừ, quy định chiều cao của hàng cột thứ *j* trong các đơn vị của CTB; danh sách `ColBd[i]` đối với *i* nằm trong khoảng từ không đến `num_tile_columns_minus1 + 1`, không loại trừ, quy định vị trí của đường biên cột ô thứ *i* trong các đơn vị của CTB; danh sách `RowBd[j]` đối với *j* nằm trong khoảng từ không đến `num_tile_rows_minus1 + 1`, không loại trừ, quy định vị trí của đường biên hàng cột thứ *j* trong các đơn vị của CTB; danh sách `CtbAddrRsToTs[ ctbAddrRs ]` đối với `ctbAddrRs` nằm trong khoảng từ không đến `PicSizeInCtbsY - 1`, không loại trừ, quy định việc chuyển đổi từ địa chỉ CTB trong việc quét mảng CTB của ảnh sang địa chỉ CTB trong việc quét ô; danh sách `CtbAddrTsToRs[ ctbAddrTs ]` đối với `ctbAddrTs` nằm trong khoảng từ không đến `PicSizeInCtbsY - 1`, không loại trừ, quy định việc chuyển đổi từ địa chỉ CTB trong việc quét mảng CTB của ảnh; danh sách `TileId[ ctbAddrTs ]` đối với `ctbAddrTs` nằm trong khoảng từ không đến `PicSizeInCtbsY - 1`, không loại trừ, quy định việc chuyển đổi từ địa chỉ CTB trong việc quét ô tới ID ô; danh sách

NumCtusInTile[tileIdx] đối với tileIdx nằm trong khoảng từ không đến PicSizeInCtbsY – 1, không loại trừ, quy định việc chuyển đổi từ chỉ số ô sang số lượng của các CTU trong ô; danh sách FirstCtbAddrTs[tileIdx] đối với tileIdx nằm trong khoảng từ không đến NumTilesInPic – 1, không loại trừ, quy định việc chuyển đổi từ ID ô sang địa chỉ CTB trong việc quét ô của CTB thứ nhất trong ô; danh sách ColumnWidthInLumaSamples[i] đối với i nằm trong khoảng từ không đến num_tile_columns_minus1, không loại trừ, quy định chiều rộng của cột ô thứ i trong các đơn vị của các mẫu độ chói; và danh sách RowHeightInLumaSamples[j] đối với j nằm trong khoảng từ không đến num_tile_rows_minus1, không loại trừ, quy định chiều cao của hàng cột thứ j trong các đơn vị của các mẫu độ chói. Các giá trị của ColumnWidthInLumaSamples[i] đối với i nằm trong khoảng từ không đến num_tile_columns_minus1, không loại trừ, và RowHeightInLumaSamples[j] đối với j nằm trong khoảng từ không đến num_tile_rows_minus1, không loại trừ, sẽ đều lớn hơn không.

loop_filter_across_tiles_enabled_flag bằng một quy định rằng các hoạt động lọc trong vòng lặp có thể được thực hiện giữa các đường biên ô trong các ảnh con tham chiếu đến SPPS. loop_filter_across_tiles_enabled_flag bằng không quy định rằng các hoạt động lọc trong vòng lặp không được thực hiện giữa các đường biên ô trong các ảnh con tham chiếu đến SPPS. Các hoạt động lọc trong vòng lặp bao gồm các hoạt động của bộ lọc khử khối, bộ lọc dịch vị thích ứng mẫu, và bộ lọc vòng thích ứng. Khi không có mặt, giá trị của loop_filter_across_tiles_enabled_flag được suy ra là bằng một. loop_filter_across_subpic_enabled_flag bằng một quy định rằng các hoạt động lọc trong vòng lặp có thể được thực hiện giữa các đường biên ảnh con trong các ảnh con tham chiếu đến SPPS. loop_filter_across_subpic_enabled_flag bằng không quy định rằng các hoạt động lọc trong vòng lặp không được thực hiện giữa các đường biên ảnh con trong các ảnh con tham chiếu đến SPPS. Các hoạt động lọc trong vòng lặp bao gồm các hoạt động của bộ lọc khử khối, bộ lọc dịch vị thích ứng mẫu, và bộ lọc vòng thích ứng. Khi không có mặt, giá trị của loop_filter_across_subpic_enabled_flag được suy ra là bằng với giá trị của loop_filter_across_tiles_enabled_flag.

Cú pháp phần đầu nhóm ô và các ngữ nghĩa chung là như sau đây.

tile_group_header() {	Bộ mô tả
tile_group_subpic_id	u(v)
tile_group_subpic_parameter_set_id	u(v)
...	
}	

Giá trị của phần tử cú pháp phần đầu nhóm ô tile_group_pic_parameter_set_id và tile_group_pic_order_cnt_lsb sẽ là giống nhau trong tất cả các phần đầu nhóm ô của ảnh được tạo mã. Giá trị của phần tử cú pháp phần đầu nhóm ô tile_group_subpic_id sẽ là giống nhau trong tất cả các phần đầu nhóm ô của ảnh con được tạo mã. tile_group_subpic_id nhận dạng ảnh con mà nhóm ô thuộc về. Chiều dài của tile_group_subpic_id là subpic_id_len_minus1 + 1 bit. tile_group_subpic_parameter_set_id quy định giá trị của sps_subpic_parameter_set_id đối với SPSS đang được sử dụng. Giá trị của tile_group_spps_parameter_set_id sẽ nằm trong khoảng từ không đến sáu mươi ba, không loại trừ.

Các biến số sau đây được suy ra và ghi đè các biến số tương ứng được suy ra từ SPS hoạt động:

PicWidthInLumaSamples = SubpicWidthInLumaSamples[tile_group_subpic_id]

PicHeightInLumaSamples = PicHeightInLumaSamples[tile_group_subpic_id]

SubPicRightBorderInPic = SubPicRightBorderInPic[tile_group_subpic_id]

SubPicBottomBorderInPic = SubPicBottomBorderInPic[tile_group_subpic_id]

PicWidthInCtbsY = SubPicWidthInCtbsY[tile_group_subpic_id]

PicHeightInCtbsY = SubPicHeightInCtbsY[tile_group_subpic_id]

PicSizeInCtbsY = SubPicSizeInCtbsY[tile_group_subpic_id]

PicWidthInMinCbsY = SubPicWidthInMinCbsY[tile_group_subpic_id]

PicHeightInMinCbsY = SubPicHeightInMinCbsY[tile_group_subpic_id]

PicSizeInMinCbsY = SubPicSizeInMinCbsY[tile_group_subpic_id]

PicSizeInSamplesY = SubPicSizeInSamplesY[tile_group_subpic_id]

PicWidthInSamplesC = SubPicWidthInSamplesC[tile_group_subpic_id]

$\text{PicHeightInSamplesC} = \text{SubPicHeightInSamplesC}[\text{tile_group_subpic_id}]$

Cú pháp đơn vị cây tạo mã là như sau đây.

<code>coding_tree_unit() {</code>	Bộ mô tả
<code> xCtb = (CtbAddrInRs % PicWidthInCtbsY) << CtbLog2SizeY + SubpictureXOffset</code>	
<code> yCtb = (CtbAddrInRs / PicWidthInCtbsY) << CtbLog2SizeY + SubpictureYOffset</code>	
<code> ...</code>	
<code>}</code>	

<code>dual_tree_implicit_qt_split(x0, y0, log2CbSize, cqtDepth) {</code>	Bộ mô tả
<code> if(log2CbSize > 6) {</code>	
<code> x1 = x0 + (1 << (log2CbSize - 1))</code>	
<code> y1 = y0 + (1 << (log2CbSize - 1))</code>	
<code> dual_tree_implicit_qt_split(x0, y0, log2CbSize - 1, cqtDepth + 1)</code>	
<code> if(x1 < SubPicRightBorderInPic)</code>	
<code> dual_tree_implicit_qt_split(x1, y0, log2CbSize - 1, cqtDepth + 1)</code>	
<code> if(y1 < SubPicBottomBorderInPic)</code>	
<code> dual_tree_implicit_qt_split(x0, y1, log2CbSize - 1, cqtDepth + 1)</code>	
<code> if(x1 < SubPicRightBorderInPic && y1 < SubPicBottomBorderInPic)</code>	
<code> dual_tree_implicit_qt_split(x1, y1, log2CbSize - 1, cqtDepth + 1)</code>	
<code> } else {</code>	
<code> coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_LUMA)</code>	
<code> coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_CHROMA)</code>	
<code> }</code>	
<code>}</code>	

Cú pháp cây tứ phân tạo mã và các ngữ nghĩa là như sau đây.

<code>coding_quadtree(x0, y0, log2CbSize, cqtDepth, treeType) {</code>	Bộ mô tả
<code> minQtSize = (treeType == DUAL_TREE_CHROMA) ? MinQtSizeC : MinQtSizeY</code>	
<code> maxBtSize = (treeType == DUAL_TREE_CHROMA) ? MaxBtSizeC : MaxBtSizeY</code>	
<code> if((((x0 + (1 << log2CbSize) <= PicWidthInLumaSamples) ? 1 : 0) + ((y0 + (1 << log2CbSize) <= PicHeightInLumaSamples) ? 1 : 0) + (((1 << log2CbSize) <= maxBtSize) ? 1 : 0)) >= 2 && (1 << log2CbSize) > minQtSize)</code>	
<code> qt_split_cu_flag[x0][y0]</code>	<code>ae(v)</code>
<code> if(cu_qp_delta_enabled_flag && cqtDepth <= diff_cu_qp_delta_depth) {</code>	
<code> IsCuQpDeltaCoded = 0</code>	
<code> CuQpDeltaVal = 0</code>	
<code> CuQgTopLeftX = x0</code>	
<code> CuQgTopLeftY = y0</code>	
<code> }</code>	
<code> if(qt_split_cu_flag[x0][y0]) {</code>	
<code> x1 = x0 + (1 << (log2CbSize - 1))</code>	
<code> y1 = y0 + (1 << (log2CbSize - 1))</code>	
<code> coding_quadtree(x0, y0, log2CbSize - 1, cqtDepth + 1, treeType)</code>	
<code> if(x1 < SubPicRightBorderInPic)</code>	
<code> coding_quadtree(x1, y0, log2CbSize - 1, cqtDepth + 1, treeType)</code>	
<code> if(y1 < SubPicBottomBorderInPic)</code>	
<code> coding_quadtree(x0, y1, log2CbSize - 1, cqtDepth + 1, treeType)</code>	
<code> if(x1 < SubPicRightBorderInPic && y1 < SubPicBottomBorderInPic)</code>	
<code> coding_quadtree(x1, y1, log2CbSize - 1, cqtDepth + 1, treeType)</code>	
<code> } else</code>	
<code> multi_type_tree(x0, y0, 1 << log2CbSize, 1 << log2CbSize, cqtDepth, 0, 0, 0, treeType)</code>	
<code>}</code>	

`qt_split_cu_flag[ x0 ][ y0 ]` quy định xem đơn vị tạo mã có được chia thành các đơn vị tạo mã với kích thước nửa ngang và dọc hay không. Các chỉ số mảng `x0, y0` quy định vị trí `( x0, y0 )` của mẫu độ chói phía trên cùng bên trái của mẫu tạo mã được xem xét so với

mẫu độ chói phía trên cùng bên trái của ảnh. Khi `qt_split_cu_flag[ x0 ][ y0 ]` không có mặt, phần sau đây áp dụng: Nếu một hoặc nhiều điều kiện sau đây là đúng, giá trị của `qt_split_cu_flag[ x0 ][ y0 ]` được suy ra là bằng một. $x0 + (1 \ll \log_2 \text{CbSize})$ là lớn hơn `SubPicRightBorderInPic` và $(1 \ll \log_2 \text{CbSize})$ là lớn hơn `MaxBtSizeC` nếu `treeType` là bằng với `DUAL_TREE_CHROMA` hoặc nếu không thì lớn hơn `MaxBtSizeY`. $y0 + (1 \ll \log_2 \text{CbSize})$ là lớn hơn `SubPicBottomBorderInPic` và $(1 \ll \log_2 \text{CbSize})$ là lớn hơn `MaxBtSizeC` nếu `treeType` là bằng với `DUAL_TREE_CHROMA` hoặc nếu không thì lớn hơn `MaxBtSizeY`.

Nếu không, nếu tất cả các điều kiện sau đây là đúng, giá trị của `qt_split_cu_flag[ x0 ][ y0 ]` được suy ra là bằng với 1: $x0 + (1 \ll \log_2 \text{CbSize})$ là lớn hơn `SubPicRightBorderInPic`, $y0 + (1 \ll \log_2 \text{CbSize})$ là lớn hơn `SubPicBottomBorderInPic`, và $(1 \ll \log_2 \text{CbSize})$ là lớn hơn `MinQtSizeC` nếu `treeType` là bằng với `DUAL_TREE_CHROMA` hoặc nếu không thì lớn hơn `MinQtSizeY`. Nếu không, giá trị của `qt_split_cu_flag[ x0 ][ y0 ]` được suy ra là bằng với zero.

Các ngữ nghĩa và cú pháp cây nhiều loại là như sau đây.

<code>multi_type_tree(x0, y0, cbWidth, cbHeight, cqtDepth, mttDepth, depthOffset, partIdx, treeType) {</code>	Bộ mô tả
<code>if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && (x0 + cbWidth <= SubPicRightBorderInPic) && (y0 + cbHeight <= SubPicBottomBorderInPic))</code>	
<code> mtt_split_cu_flag</code>	<code>ae(v)</code>
<code> if(cu_qp_delta_enabled_flag && (cqtDepth + mttDepth) <= diff_cu_qp_delta_depth) {</code>	
<code> IsCuQpDeltaCoded = 0</code>	
<code> CuQpDeltaVal = 0</code>	
<code> CuQgTopLeftX = x0</code>	
<code> CuQgTopLeftY = y0</code>	
<code> }</code>	
<code> if(mtt_split_cu_flag) {</code>	
<code> if((allowSplitBtHor allowSplitTtHor) && (allowSplitBtVer allowSplitTtVer))</code>	
<code> mtt_split_cu_vertical_flag</code>	<code>ae(v)</code>

if((allowSplitBtVer && allowSplitTtVer && mtt_split_cu_vertical_flag) (allowSplitBtHor && allowSplitTtHor && !mtt_split_cu_vertical_flag))	
mtt_split_cu_binary_flag	ae(v)
if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) {	
depthOffset += (x0 + cbWidth > SubPicRightBorderInPic) ? 1 : 0	
x1 = x0 + (cbWidth / 2)	
multi_type_tree(x0, y0, cbWidth / 2, cbHeight, cqtDepth, mttDepth + 1, depthOffset, 0, treeType)	
if(x1 < SubPicRightBorderInPic)	
multi_type_tree(x1, y0, cbWidth / 2, cbHeightY, cqtDepth, mttDepth + 1, depthOffset, 1, treeType)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR) {	
depthOffset += (y0 + cbHeight > SubPicBottomBorderInPic) ? 1 : 0	
y1 = y0 + (cbHeight / 2)	
multi_type_tree(x0, y0, cbWidth, cbHeight / 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType)	
if(y1 < SubPicBottomBorderInPic)	
multi_type_tree(x0, y1, cbWidth, cbHeight / 2, cqtDepth, mttDepth + 1, depthOffset, 1, treeType)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER) {	
x1 = x0 + (cbWidth / 4)	
x2 = x0 + (3 * cbWidth / 4)	
multi_type_tree(x0, y0, cbWidth / 4, cbHeight, cqtDepth, mttDepth + 1, depthOffset, 0, treeType)	
multi_type_tree(x1, y0, cbWidth / 2, cbHeight, cqtDepth, mttDepth + 1, depthOffset, 1, treeType)	
multi_type_tree(x2, y0, cbWidth / 4, cbHeight, cqtDepth, mttDepth + 1, depthOffset, 2, treeType)	
} else { /* SPLIT_TT_HOR */	
y1 = y0 + (cbHeight / 4)	
y2 = y0 + (3 * cbHeight / 4)	

multi_type_tree(x0, y0, cbWidth, cbHeight / 4, cqtDepth, mttDepth + 1, depthOffset, 0, treeType)	
multi_type_tree(x0, y1, cbWidth, cbHeight / 2, cqtDepth, mttDepth + 1, depthOffset, 1, treeType)	
multi_type_tree(x0, y2, cbWidth, cbHeight / 4, cqtDepth, mttDepth + 1, depthOffset, 2, treeType)	
}	
} else	
coding_unit(x0, y0, cbWidth, cbHeight, treeType)	
}	

mtt_split_cu_flag bằng không quy định rằng đơn vị tạo mã không bị chia. mtt_split_cu_flag bằng một quy định rằng đơn vị tạo mã được chia thành hai đơn vị tạo mã sử dụng phép chia nhị phân hoặc thành ba đơn vị tạo mã sử dụng phép chia bậc ba như được chỉ thị bởi phần tử cú pháp mtt_split_cu_binary_flag. Phép chia nhị phân hoặc bậc ba có thể là thẳng hoặc ngang như được chỉ thị bởi phần tử cú pháp mtt_split_cu_vertical_flag. Khi mtt_split_cu_flag không có mặt, giá trị của mtt_split_cu_flag được suy ra như dưới đây. Nếu một hoặc nhiều điều kiện sau đây là đúng, giá trị của mtt_split_cu_flag được suy ra là bằng với 1: $x0 + cbWidth$ là lớn hơn SubPicRightBorderInPic, và $y0 + cbHeight$ là lớn hơn SubPicBottomBorderInPic. Nếu không, giá trị của mtt_split_cu_flag được suy ra là bằng với không.

Tiến trình suy ra để dự đoán vector chuyển động của độ chói theo thời gian là như sau. Các đầu ra của tiến trình này là: việc dự báo vector chuyển động mvLXCol trong độ chính xác mẫu phân 1/16, và cờ độ khả dụng availableFlagLXCol. Biến số currCb quy định khối tạo mã độ chói hiện tại tại vị trí độ chói (xCb, yCb). Các biến số mvLXCol và availableFlagLXCol được suy ra như dưới đây. Nếu tile_group_temporal_mvp_enabled_flag là bằng không, hoặc nếu ảnh tham chiếu là ảnh hiện tại, cả hai thành phần của mvLXCol được thiết đặt bằng không và availableFlagLXCol được thiết đặt bằng không. Nếu không (tile_group_temporal_mvp_enabled_flag là bằng một và ảnh tham chiếu không phải là hình ảnh hiện tại), các bước được xếp thứ tự sau đây áp dụng. Vector chuyển động cùng vị trí phía dưới bên phải được suy ra như dưới đây:

$$xColBr = xCb + cbWidth \quad (8-355)$$

$$yColBr = yCb + cbHeight \quad (8-356)$$

Nếu $yCb \gg CtbLog2SizeY$ là bằng với $yColBr \gg CtbLog2SizeY$, $yColBr$ là nhỏ hơn $SubPicBottomBorderInPic$ và $xColBr$ là nhỏ hơn $SubPicRightBorderInPic$, phần sau đây áp dụng. Biến số $colCb$ quy định khối tạo mã độ chói bao phủ vị trí được cải biến được cho bởi $((xColBr \gg 3) \ll 3, (yColBr \gg 3) \ll 3)$ bên trong ảnh nằm cùng vị trí được quy định bởi $ColPic$. Vị trí độ chói $(xColCb, yColCb)$ được đặt bằng mẫu phía trên cùng bên trái của khối tạo mã độ chói cùng vị trí mà được quy định bởi $colCb$ so với mẫu độ chói phía trên cùng bên trái của ảnh nằm cùng vị trí mà được quy định bởi $ColPic$. Tiến trình suy ra đối với các vectơ chuyển động nằm cùng vị trí được gọi ra với $currCb$, $colCb$, $(xColCb, yColCb)$, $refIdxLX$ và $sbFlag$ được thiết đặt bằng không làm các đầu vào, và đầu ra được ấn định cho $mvLXCol$ và $availableFlagLXCol$. Nếu không, cả hai thành phần của $mvLXCol$ được thiết đặt bằng không và $availableFlagLXCol$ được thiết đặt bằng không.

Tiến trình suy ra đối với các ứng viên gộp tam giác theo thời gian là như sau đây. Các biến số $mvLXColC0$, $mvLXColC1$, $availableFlagLXColC0$ và $availableFlagLXColC1$ được suy ra như dưới đây. Nếu $tile_group_temporal_mvp_enabled_flag$ là bằng không, cả hai thành phần của $mvLXColC0$ và $mvLXColC1$ được thiết đặt bằng không và $availableFlagLXColC0$ và $availableFlagLXColC1$ được thiết đặt bằng không. Nếu không ($tile_group_temporal_mvp_enabled_flag$ là bằng với 1), các bước được xếp thứ tự sau đây áp dụng. Vectơ chuyển động cùng vị trí phía dưới bên phải $mvLXColC0$ được suy ra như dưới đây:

$$xColBr = xCb + cbWidth \quad (8-392)$$

$$yColBr = yCb + cbHeight \quad (8-393)$$

Nếu $yCb \gg CtbLog2SizeY$ là bằng với $yColBr \gg CtbLog2SizeY$, $yColBr$ là nhỏ hơn $SubPicBottomBorderInPic$ và $xColBr$ là nhỏ hơn $SubPicRightBorderInPic$, phần sau đây áp dụng. Biến số $colCb$ quy định khối tạo mã độ chói bao phủ vị trí được cải biến được cho bởi $((xColBr \gg 3) \ll 3, (yColBr \gg 3) \ll 3)$ bên trong ảnh nằm cùng vị trí được quy định bởi $ColPic$. Vị trí độ chói $(xColCb, yColCb)$ được đặt bằng mẫu phía trên cùng bên trái của khối tạo mã độ chói cùng vị trí mà được quy định bởi $colCb$ so với mẫu độ chói phía trên cùng bên trái của ảnh nằm cùng vị trí mà được quy định bởi $ColPic$. Tiến trình suy ra đối với các vectơ chuyển động nằm cùng vị trí được gọi ra với $currCb$, $colCb$, $(xColCb, yColCb)$, $refIdxLXC0$ và $sbFlag$ được thiết đặt bằng không làm các đầu vào, và đầu ra được ấn định cho $mvLXColC0$ và $availableFlagLXColC0$. Nếu không, cả hai thành

phần của mvLXColC0 được thiết đặt bằng không và availableFlagLXColC0 được thiết đặt bằng không.

Tiến trình suy ra đối với các ứng viên gộp vector chuyển động điểm điều khiển afin được kết cấu là như sau đây. Vector chuyển động điểm điều khiển (nằm cùng vị trí phía dưới bên phải) thứ tư cpMvLXCorner[3], chỉ số tham chiếu refIdxLXCorner[3], cờ sử dụng danh sách dự báo predFlagLXCorner[3] và cờ độ khả dụng availableFlagCorner[3] với X là 0 và 1 được suy ra như dưới đây. Các chỉ số tham chiếu đối với ứng viên gộp theo thời gian, refIdxLXCorner[3], với X là không hoặc một, được thiết đặt bằng với không. Các biến số mvLXCol và availableFlagLXCol, với X là không hoặc một, được suy ra như dưới đây. Nếu tile_group_temporal_mvp_enabled_flag là bằng không, cả hai thành phần của mvLXCol được thiết đặt bằng không và availableFlagLXCol được thiết đặt bằng không. Nếu không (tile_group_temporal_mvp_enabled_flag là bằng một), phần sau đây áp dụng:

$$xColBr = xCb + cbWidth \quad (8-566)$$

$$yColBr = yCb + cbHeight \quad (8-567)$$

Nếu $yCb \gg CtbLog2SizeY$ là bằng với $yColBr \gg CtbLog2SizeY$, $yColBr$ là nhỏ hơn SubPicBottomBorderInPic và $xColBr$ là nhỏ hơn SubPicRightBorderInPic, phần sau đây áp dụng. Biến số colCb quy định khối tạo mã độ chói bao phủ vị trí được cải biến được cho bởi $((xColBr \gg 3) \ll 3, (yColBr \gg 3) \ll 3)$ bên trong ảnh nằm cùng vị trí được quy định bởi ColPic. Vị trí độ chói $(xColCb, yColCb)$ được đặt bằng mẫu phía trên cùng bên trái của khối tạo mã độ chói cùng vị trí mà được quy định bởi colCb so với mẫu độ chói phía trên cùng bên trái của ảnh nằm cùng vị trí mà được quy định bởi ColPic. Tiến trình suy ra đối với các vector chuyển động nằm cùng vị trí được gọi ra với currCb, colCb, $(xColCb, yColCb)$, refIdxLX và sbFlag được thiết đặt bằng không làm các đầu vào, và đầu ra được ấn định cho mvLXCol và availableFlagLXCol. Nếu không, cả hai thành phần của mvLXCol được thiết đặt bằng với 0 và availableFlagLXCol được thiết đặt bằng không. Thay thế tất cả các sự xuất hiện của pic_width_in_luma_samples với PicWidthInLumaSamples. Thay thế tất cả các sự xuất hiện của pic_height_in_luma_samples với PicHeightInLumaSamples.

Trong phương án ví dụ thứ hai, các ngữ nghĩa và cú pháp RBSP tập hợp thông số chuỗi là như sau đây.

seq_parameter_set_rbsp() {	Bộ mô tả
sps_seq_parameter_set_id	ue(v)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
num_subpic_minus1	ue(v)
subpic_id_len_minus1	ue(v)
for(i = 0; i <= num_subpic_minus1; i++)	
subpic_id[i]	u(v)
if(num_subpic_minus1 > 0) {	
subpic_level_idc[i]	u(8)
subpic_x_offset[i]	ue(v)
subpic_y_offset[i]	ue(v)
subpic_width_in_luma_samples[i]	ue(v)
subpic_height_in_luma_samples[i]	ue(v)
}	
}	
...	
}	

subpic_id_len_minus1 cộng một quy định số lượng của các bit được sử dụng để biểu thị phần tử cú pháp subpic_id[i] trong SPS, sps_subpic_id trong SPPS tham chiếu đến SPS, và tile_group_subpic_id trong các phần đầu nhóm ô tham chiếu đến SPS. Giá trị của subpic_id_len_minus1 sẽ nằm trong khoảng từ $\text{Ceil}(\text{Log2}(\text{num_subpic_minus1} + 3))$ đến tám, không loại trừ. Là một yêu cầu của sự tương thích luồng bit rằng không có sự chồng lấp nào trong số ảnh con[i] đối với i từ 0 đến num_subpic_minus1, không loại trừ. Mỗi ảnh con có thể là ảnh con được ràng buộc chuyển động theo thời gian.

Các ngữ nghĩa phần đầu nhóm ô chung là như sau đây. tile_group_subpic_id nhận dạng ảnh con mà nhóm ô thuộc về. Chiều dài của tile_group_subpic_id là subpic_id_len_minus1 + 1 bit. tile_group_subpic_id bằng một chỉ thị nhóm ô không thuộc về ảnh con bất kỳ.

Trong phương án ví dụ thứ ba, các ngữ nghĩa và cú pháp phần đầu đơn vị NAL là như sau đây.

<code>nal_unit_header() {</code>	Bộ mô tả
<code>forbidden_zero_bit</code>	<code>f(1)</code>
<code>nal_unit_type</code>	<code>u(5)</code>
<code>nuh_temporal_id_plus1</code>	<code>u(3)</code>
<code>nuh_subpicture_id_len</code>	<code>u(4)</code>
<code>nuh_reserved_zero_4bits</code>	<code>u(3)</code>
<code>}</code>	

`nuh_subpicture_id_len` quy định số lượng của các bit được sử dụng để biểu thị phần tử cú pháp quy định ID ảnh con. Khi giá trị của `nuh_subpicture_id_len` là lớn hơn không, cá bit thứ `nuh_subpicture_id_len` thứ nhất là sau `nuh_reserved_zero_4bits` quy định ID của ảnh con mà phần tải hữu ích của đơn vị NAL thuộc về. Khi `nuh_subpicture_id_len` là lớn hơn không, giá trị của `nuh_subpicture_id_len` bằng với giá trị của `subpic_id_len_minus1` trong SPS hoạt động. Giá trị của `nuh_subpicture_id_len` đối với các đơn vị NAL không phải VCL được ràng buộc như dưới đây. Nếu `nal_unit_type` là bằng với `SPS_NUT` hoặc `PPS_NUT`, `nuh_subpicture_id_len` bằng không, `nuh_reserved_zero_3bits` bằng với '000'. Các bộ giải mã lờ (ví dụ, loại từ luồng bit và bỏ) các đơn vị NAL với các giá trị của `nuh_reserved_zero_3bits` không bằng với '000'.

Trong ví dụ phương án ví dụ thứ tư, cú pháp lồng ảnh con là như sau đây.

<code>sub-picture_nesting(payloadSize) {</code>	Bộ mô tả
<code>all_sub_pictures_flag</code>	<code>u(1)</code>
<code>if(!all_sub_pictures_flag) {</code>	
<code>nesting_num_sub_pictures_minus1</code>	<code>ue(v)</code>
<code>for(i = 0; i <= nesting_num_sub_pictures_minus1; i++)</code>	
<code>nesting_sub_picture_id[i]</code>	<code>u(v)</code>
<code>}</code>	
<code>while(!byte_aligned())</code>	
<code>sub_picture_nesting_zero_bit /* equal to 0 */</code>	<code>u(1)</code>
<code>do</code>	
<code>sei_message()</code>	
<code>while(more_rbsp_data())</code>	
<code>}</code>	

`all_sub_pictures_flag` bằng một quy định rằng các thông điệp SEI được lồng áp dụng cho tất cả các ảnh con. `all_sub_pictures_flag` bằng một quy định rằng các ảnh con mà cho chúng các thông điệp SEI được lồng áp dụng được báo hiệu một cách tường minh bởi các phần tử cú pháp sau đây. `nesting_num_sub_pictures_minus1` cộng 1 quy định số lượng của các ảnh con mà cho chúng các thông điệp SEI được lồng áp dụng. `nesting_sub_picture_id[ i ]` chỉ thị ID ảnh con của ảnh con thứ i mà cho chúng các thông điệp SEI được lồng áp dụng. Phần tử cú pháp `nesting_sub_picture_id[ i ]` được biểu diễn bởi $\text{Ceil}(\text{Log2}(\text{nesting_num_sub_pictures_minus1} + 1))$ bit. `sub_picture_nesting_zero_bit` bằng không.

Fig.9 là hình vẽ sơ lược của thiết bị tạo mã video 900 ví dụ. Thiết bị tạo mã video 900 là thích hợp cho việc sử dụng các ví dụ/các phương án được bộc lộ như được mô tả ở đây. Thiết bị tạo mã video 900 bao gồm các cổng xuôi dòng 920, các cổng ngược dòng 950, và/hoặc các đơn vị thu phát (Tx/Rx) 910, bao gồm các bộ phát và/hoặc các bộ thu cho việc truyền thông xuôi dòng và/hoặc ngược dòng dữ liệu qua mạng. Thiết bị tạo mã video 900 cũng bao gồm bộ xử lý 930 bao gồm đơn vị logic và/hoặc đơn vị xử lý trung tâm (central processing unit - CPU) để xử lý dữ liệu và bộ nhớ 932 để lưu trữ dữ liệu. Thiết bị tạo mã video 900 cũng có thể bao gồm thành phần điện, thành phần quang sang điện (Optical-to-Electrical - OE), thành phần điện sang quang (Electrical-to-Optical - EO), và/hoặc các thành phần truyền thông không dây được ghép nối đến các cổng ngược dòng 950 và/hoặc các cổng xuôi dòng 920 để truyền thông dữ liệu qua mạng điện, mạng quang, hoặc các mạng truyền thông không dây. Thiết bị tạo mã video 900 cũng có thể bao gồm các thiết bị đầu vào và/hoặc đầu ra (Input/Output - I/O) 960 để truyền thông dữ liệu qua lại với người dùng. Các thiết bị I/O 960 có thể bao gồm các thiết bị đầu ra chẳng hạn như bộ hiển thị để hiển thị dữ liệu video, các loa để xuất dữ liệu audio, v.v.. Các thiết bị I/O 960 cũng có thể bao gồm các thiết bị đầu vào, chẳng hạn như bàn phím, chuột, bi xoay, v.v., và/hoặc các giao diện tương ứng để tương tác với các thiết bị đầu ra như vậy.

Bộ xử lý 930 được thực hiện bằng phần cứng và phần mềm. Bộ xử lý 930 có thể được thực hiện dưới dạng một hoặc nhiều chip, lõi CPU (ví dụ, dưới dạng bộ xử lý đa lõi), mảng cổng lập trình được dạng trường (Field-Programmable Gate Array - FPGA), mạch tích hợp chuyên dụng (Application Specific Integrated Circuit - ASIC), và bộ xử lý tín hiệu số (Digital Signal Processor - DSP). Bộ xử lý 930 được kết nối với cổng phía dưới 920, Tx/Rx 910, cổng phía trên 950, và bộ nhớ 932. Bộ xử lý 930 bao gồm môđun mã hóa 914. Môđun tạo

mã 914 thực hiện các phương án được bộc lộ được mô tả trên đây, chẳng hạn các phương pháp 100, 1000, 1100, và/hoặc cơ chế 700, mà có thể sử dụng luồng bit 500, ảnh 600 và/hoặc hình 800. Môđun tạo mã 914 cũng có thể thực hiện phương pháp/cơ chế bất kỳ khác được mô tả ở đây. Ngoài ra, môđun tạo mã 914 có thể thực hiện hệ thống codec 200, bộ mã hóa 300, và/hoặc bộ giải mã 400. Ví dụ, môđun tạo mã 914 có thể được sử dụng để báo hiệu và/hoặc nhận các kích thước và các vị trí ảnh con trong SPS. Trong một ví dụ khác, môđun tạo mã 914 có thể ràng buộc các chiều rộng ảnh con và các chiều cao ảnh con là các bội số của kích thước CTU trừ khi các ảnh con này được định vị ở ranh giới bên phải của ảnh hoặc ranh giới phía dưới của ảnh, một cách tương ứng. Trong một ví dụ khác, môđun tạo mã 914 có thể ràng buộc các ảnh con để bao trùm ảnh mà không có khoảng trống hoặc khoảng chồng lấp. Trong một ví dụ khác, môđun tạo mã 914 có thể được sử dụng để báo hiệu và/hoặc nhận dữ liệu chỉ thị một số ảnh con là các ảnh con ràng buộc chuyển động theo thời gian và các ảnh con khác thì không. Trong một ví dụ khác, môđun tạo mã 914 có thể báo hiệu tập hợp đầy đủ của các ID ảnh con trong SPS và bao gồm ID ảnh con trong mỗi phần đầu lát để chỉ thị ảnh con mà chứa các lát tương ứng. Trong một ví dụ khác, môđun tạo mã 914 có thể báo hiệu các mức cho mỗi ảnh con. Như vậy, môđun tạo mã 914 làm cho thiết bị tạo mã video 900 đem lại chức năng bổ sung, tránh việc xử lý nhất định để làm giảm phí tổn xử lý và/hoặc tăng hiệu quả tạo mã khi phân chia và tạo mã dữ liệu video. Theo đó, môđun tạo mã 914 cải thiện chức năng của thiết bị tạo mã video 900 cũng như khắc phục các vấn đề riêng của lĩnh vực tạo mã video. Ngoài ra, môđun tạo mã 914 thực hiện việc biến đổi thiết bị tạo mã video 900 sang trạng thái khác. Theo cách khác, môđun tạo mã 914 có thể được thực hiện dưới dạng các lệnh được lưu trữ trong bộ nhớ 932 và được thực thi bởi bộ xử lý 930 (ví dụ, dưới dạng sản phẩm chương trình máy tính được lưu trữ trên phương tiện phi chuyển tiếp).

Bộ nhớ 932 bao gồm một hoặc nhiều loại bộ nhớ chẳng hạn như các đĩa, các ổ băng, các ổ cứng trạng thái rắn, bộ nhớ chỉ đọc (read only medium - ROM), bộ nhớ truy cập ngẫu nhiên (random access memory - RAM), bộ nhớ chớp, bộ nhớ ghi địa chỉ được nội dung bậc ba (ternary content-addressable memory - TCAM), bộ nhớ truy cập ngẫu nhiên tĩnh (static random-access memory - SRAM), v.v.. Bộ nhớ 932 có thể được sử dụng làm thiết bị lưu trữ dữ liệu tràn, để lưu trữ các chương trình khi các chương trình này được chọn lựa để thực thi, và để lưu trữ các lệnh và dữ liệu mà được đọc trong quá trình thực thi chương trình.

Fig.10 là lưu đồ của phương pháp ví dụ 1000 để mã hóa các ID ảnh con trong luồng bit, chẳng hạn như luồng bit 500, để hỗ trợ việc giải mã của các ảnh con, chẳng hạn như các

ảnh con 522, 523, 622, 722, và/hoặc 822. Phương pháp 1000 có thể được sử dụng bởi bộ mã hóa, chẳng hạn hệ thống codec 200, bộ mã hóa 300, và/hoặc thiết bị tạo mã video 900 khi thực hiện phương pháp 100.

Phương pháp 1000 có thể bắt đầu khi bộ mã hóa thu chuỗi video bao gồm nhiều hình ảnh và xác định là mã hóa chuỗi video đó vào luồng bit, ví dụ, dựa trên đầu vào người dùng. Chuỗi video này được phân chia thành các ảnh/các hình ảnh/các khung để phân chia tiếp trước khi mã hóa. Tại bước 1001, ảnh được phân chia thành nhiều ảnh con. Mỗi trong số nhiều ảnh con được phân chia thêm thành một hoặc nhiều lát. Trong một số ví dụ, việc phân chia được ràng buộc sao cho sự kết hợp của nhiều ảnh con bao trùm toàn bộ diện tích của ảnh mà không có khoảng hở và không có khoảng chồng lấp.

Tại bước 1003, SPS được mã hóa thành luồng bit. SPS bao gồm các ID ảnh con cho mỗi ảnh con được phân chia từ ảnh. SPS có thể còn bao gồm kích thước ảnh con cho mỗi ảnh con trong số nhiều ảnh con. Kích thước ảnh con có thể bao gồm chiều cao ảnh con trong các mẫu độ chói và chiều rộng ảnh con trong các mẫu độ chói. SPS có thể còn bao gồm vị trí ảnh con cho mỗi ảnh con trong số nhiều ảnh con. Vị trí ảnh con có thể bao gồm khoảng cách dịch vị giữa mẫu phía trên cùng bên trái của ảnh con tương ứng và mẫu phía trên cùng bên trái của ảnh. SPS có thể còn bao gồm việc chỉ thị rằng một hoặc nhiều ảnh con là các ảnh con ràng buộc chuyển động theo thời gian.

Tại bước 1005, các phần đầu lát được mã hóa thành luồng bit. Các phần đầu lát bao gồm các ID ảnh con chỉ thị các lát được liên kết với các phần đầu lát được bao gồm trong ảnh con được chỉ thị. Tại bước 1007, một hoặc nhiều ảnh con trong số nhiều ảnh con cũng được mã hóa thành luồng bit. Tại bước 1009, luồng bit được lưu trữ cho việc truyền thông về phía bộ giải mã. Luồng bit có thể sau đó được phát về phía bộ giải mã như mong muốn. Trong một số ví dụ, luồng bit con có thể được trích xuất từ luồng bit được mã hóa. Trong trường hợp như vậy, luồng bit được phát là luồng bit con. Trong các ví dụ khác, luồng bit được mã hóa có thể được phát cho việc trích xuất luồng bit con ở bộ giải mã. Trong các ví dụ khác nữa, luồng bit được mã hóa có thể được giải mã và được hiển thị mà không có việc trích xuất luồng bit con. Trong ví dụ bất kỳ trong số các ví dụ trên, danh sách hoàn chỉnh của tất cả các ID ảnh con có thể được gửi trong SPS cùng với thông tin kích thước tương ứng. Mỗi phần đầu lát chứa ID ảnh con chỉ thị ảnh con mà bao gồm lát tương ứng. Theo cách này, các ảnh con và các lát tương ứng có thể được trích xuất và được định vị mà không tham chiếu tới các ảnh con khác. Do đó, các ID ảnh con hỗ trợ chức năng được nâng cao và/hoặc

hiệu suất mã hóa tăng, mà làm giảm việc sử dụng của các tài nguyên mạng, các tài nguyên bộ nhớ, và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Fig.11 là lưu đồ của phương pháp ví dụ 1100 của việc giải mã luồng bit, chẳng hạn như luồng bit 500 và/hoặc luồng bit con 501, của các ảnh con, chẳng hạn như các ảnh con 522, 523, 622, 722, và/hoặc 822, dựa vào các ID ảnh con. Phương pháp 1100 có thể được sử dụng bởi bộ giải mã, chẳng hạn hệ thống codec 200, bộ giải mã 400, và/hoặc thiết bị tạo mã video 900 khi thực hiện phương pháp 100. Ví dụ, phương pháp 1100 có thể được áp dụng để giải mã luồng bit được tạo ra là kết quả của phương pháp 1000.

Phương pháp 1100 có thể khi bộ giải mã bắt đầu thu luồng bit chứa các ảnh con. Luồng bit có thể bao gồm chuỗi video hoàn chỉnh hoặc luồng bit có thể là luồng bit con chứa tập hợp được giảm của các ảnh con cho việc trích xuất riêng biệt. Ở bước 1101, luồng bit được thu. Luồng bit bao gồm SPS, một hoặc nhiều ảnh con được phân chia từ ảnh, và một hoặc nhiều phần đầu lát được liên kết với các lát được chứa trong một hoặc nhiều ảnh con.

Tại bước 1103, SPS được phân tích để nhận được các ID ảnh con cho một hoặc nhiều ảnh con. Ngoài ra, các phần đầu lát được phân tích để nhận được ID ảnh con hiện tại chỉ thị các lát được liên kết với các phần đầu lát được chứa trong ảnh con hiện tại được nhận dạng bởi ID ảnh con hiện tại. SPS có thể còn bao gồm kích thước ảnh con cho mỗi ảnh con được phân chia từ ảnh. Kích thước ảnh con có thể bao gồm chiều cao ảnh con trong các mẫu độ chói và chiều rộng ảnh con trong các mẫu độ chói. SPS có thể còn bao gồm vị trí ảnh con cho mỗi ảnh con được phân chia từ ảnh. Vị trí ảnh con có thể bao gồm khoảng cách dịch vị giữa mẫu phía trên cùng bên trái của ảnh con tương ứng và mẫu phía trên cùng bên trái của ảnh. SPS có thể còn bao gồm việc chỉ thị rằng một hoặc nhiều ảnh con là các ảnh con ràng buộc chuyển động theo thời gian. Do đó, SPS cũng có thể được phân tích để nhận được kích thước ảnh con, vị trí ảnh con, và/hoặc các chỉ thị bất kỳ của các ảnh con ràng buộc chuyển động theo thời gian.

Tại bước 1105, các lát có thể được so khớp với ảnh con hiện tại từ một hoặc nhiều ảnh con dựa vào ID ảnh con hiện tại trong (các) phần đầu lát. Ngoài ra, các lát có thể được định vị dựa vào kích thước ảnh con và vị trí ảnh con đối với ảnh con hiện tại. Ảnh con hiện tại có thể sau đó được giải mã dựa vào các ID ảnh con để tạo chuỗi video, ví dụ bằng việc sử dụng việc hiệu chỉnh giữa các ID ảnh con giữa SPS và phần đầu lát cùng với vị trí của các lát được xác định dựa vào vị trí ảnh con và kích thước. Chuỗi video có thể sau đó được

chuyển tiếp cho việc hiển thị ở bước 1107. Theo đó, danh sách hoàn chỉnh của tất cả các ID ảnh con có thể được gửi trong SPS cùng với thông tin kích thước tương ứng. Mỗi phần đầu lát chứa ID ảnh con chỉ thị ảnh con mà bao gồm lát tương ứng. Theo cách này, các ảnh con và các lát tương ứng có thể được trích xuất và được định vị mà không tham chiếu tới các ảnh con khác. Do đó, các ID ảnh con hỗ trợ chức năng được nâng cao và/hoặc hiệu suất mã hóa tăng, mà làm giảm việc sử dụng của các tài nguyên mạng, các tài nguyên bộ nhớ, và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Fig.12 là hình vẽ sơ lược của hệ thống ví dụ 1200 để báo hiệu các ID ảnh con, chẳng hạn như các ID ảnh con cho các ảnh con 522, 523, 622, 722, và/hoặc 822, thông qua luồng bit, chẳng hạn như luồng bit 500 và/hoặc luồng bit con 501. Hệ thống 1200 có thể được thực hiện bởi bộ mã hóa và bộ giải mã, chẳng hạn hệ thống codec 200, bộ mã hóa 300, bộ giải mã 400, và/hoặc thiết bị tạo mã video 900. Ngoài ra, hệ thống 1200 có thể được sử dụng khi thực hiện phương pháp 100, 1000, và/hoặc 1100.

Hệ thống 1200 bao gồm bộ mã hóa video 1202. Bộ mã hóa video 1202 bao gồm môđun phân chia 1201 để phân chia ảnh thành nhiều ảnh con; phân chia mỗi ảnh con trong số nhiều ảnh con thành một hoặc nhiều lát; và phân chia mỗi ảnh con trong số nhiều ảnh con thành một hoặc nhiều lát. Bộ mã hóa video 1202 còn bao gồm môđun mã hóa 1203 cho việc mã hóa thành luồng bit SPS bao gồm các ID ảnh con cho mỗi ảnh con được phân chia từ ảnh; mã hóa thành luồng bit các phần đầu lát bao gồm các ID ảnh con chỉ thị các lát được bao gồm trong các ảnh con được chỉ thị; và mã hóa một hoặc nhiều ảnh con trong số nhiều ảnh con vào luồng bit. Bộ mã hóa video 1202 còn bao gồm môđun lưu trữ 1205 để lưu trữ luồng bit cho việc truyền thông về phía bộ giải mã. Bộ mã hóa video 1202 còn bao gồm môđun phát 1207 để phát luồng bit bao gồm (các) ảnh con và các ID ảnh con về phía bộ giải mã. Bộ mã hóa video 1202 có thể còn được tạo cấu hình để thực hiện bước bất kỳ trong số các bước của phương pháp 1000.

Hệ thống 1200 cũng bao gồm bộ giải mã video 1210. Bộ giải mã video 1210 bao gồm môđun thu 1211 để thu luồng bit bao gồm tập hợp thông số chuỗi (SPS), một hoặc nhiều ảnh con được phân chia từ ảnh, và một hoặc nhiều phần đầu lát được liên kết với một hoặc nhiều lát. Bộ giải mã video 1210 còn bao gồm môđun phân tích 1213 để phân tích SPS để nhận được các bộ nhận dạng (các ID) ảnh con cho mỗi trong số một hoặc nhiều ảnh con; và phân tích các phần đầu lát để nhận được ID ảnh con hiện tại chỉ thị các lát được bao gồm trong ảnh con hiện tại từ một hoặc nhiều ảnh con. Bộ giải mã video 1210 còn bao gồm môđun giải

mã 1215 để giải mã ảnh con hiện tại dựa vào các ID ảnh con để tạo chuỗi video. Bộ giải mã video 1110 còn bao gồm môđun chuyển tiếp 1217 để chuyển tiếp chuỗi video cho việc hiển thị. Bộ giải mã video 1210 có thể còn được tạo cấu hình để thực hiện bước bất kỳ trong số các bước của phương pháp 1100.

Thành phần thứ nhất là được ghép nối trực tiếp đến thành phần thứ hai khi không có thành phần xen giữa nào, ngoại trừ đường dây, vết, hoặc phương tiện khác giữa thành phần thứ nhất và thành phần thứ hai. Thành phần thứ nhất được ghép nối gián tiếp đến thành phần thứ hai khi có các thành phần xen giữa mà không phải là đường dây, vết, hoặc phương tiện khác giữa thành phần thứ nhất và thành phần thứ hai. Thuật ngữ “được ghép nối” và các biến thể của nó là bao gồm cả việc được ghép nối trực tiếp và việc được ghép nối gián tiếp. Việc sử dụng thuật ngữ “khoảng” có nghĩa là khoảng bao gồm $\pm 10\%$ của con số theo sau, trừ khi được nói khác đi.

Cũng cần hiểu rằng các bước của các phương pháp ví dụ được nêu ở đây là không nhất thiết phải được thực hiện theo thứ tự được mô tả, và thứ tự của các bước của các phương pháp đó cần được hiểu là chỉ để làm ví dụ. Tương tự như vậy, các bước bổ sung có thể được bao gồm trong các phương pháp đó, và các bước nhất định có thể được lược bỏ hoặc được kết hợp, theo các phương pháp nhất quán với các phương án khác nhau của sáng chế.

Tuy một số phương án đã được cung cấp trong phần bộc lộ này, nhưng cần hiểu rằng các hệ thống và các phương pháp được bộc lộ có thể được thực hiện dưới nhiều dạng cụ thể khác mà không nằm ngoài nguyên lý hoặc phạm vi của sáng chế. Các ví dụ được nêu cần được hiểu là nhằm mục đích minh họa chứ không nhằm mục đích giới hạn, và sáng chế không bị giới hạn ở các chi tiết được cung cấp ở đây. Ví dụ, các phần tử hoặc các thành phần khác nhau có thể được kết hợp hoặc được tích hợp trong hệ thống khác, hoặc các dấu hiệu nhất định có thể được bỏ qua, hoặc không được thực hiện.

Ngoài ra, các kỹ thuật, các hệ thống, các hệ thống con, và các phương pháp mà được mô tả và được thể hiện trong các phương án khác nhau dưới dạng rời rạc hoặc riêng biệt là có thể được kết hợp hoặc được tích hợp với các hệ thống, các thành phần, các kỹ thuật, hoặc các phương pháp khác mà không nằm ngoài phạm vi của sáng chế. Các ví dụ khác về các phương án thay đổi và thay thế là có thể được nhận thấy bởi người có hiểu biết trung bình trong lĩnh vực và có thể được tạo ra mà không nằm ngoài nguyên lý và phạm vi được bộc lộ ở đây.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã được thực hiện ở bộ giải mã, phương pháp này bao gồm các bước:

thu (1101), bởi bộ thu của bộ giải mã, luồng bit bao gồm tập hợp thông số chuỗi (sequence parameter set, SPS), một hoặc nhiều ảnh con được phân chia từ ảnh, và một hoặc nhiều phần đầu lát (slice) được liên kết với một hoặc nhiều lát; trong đó SPS bao gồm tập hợp đầy đủ của tất cả các bộ nhận dạng ảnh con, các ID, đối với một hoặc nhiều ảnh con, trong đó mỗi ảnh con tương ứng với ID ảnh con; và trong đó phần đầu lát bao gồm ID ảnh con hiện tại, ID ảnh con hiện tại chỉ thị ảnh con mà chứa lát được liên kết với phần đầu lát; trong đó SPS còn bao gồm kích thước ảnh con và vị trí ảnh con cho mỗi ảnh con được phân chia từ ảnh;

giải mã (1103), bởi bộ xử lý của bộ giải mã, SPS để nhận được ID ảnh con, kích thước ảnh con và vị trí ảnh con cho mỗi ảnh con trong số một hoặc nhiều ảnh con đó; và

giải mã (1103), bởi bộ xử lý, phần đầu lát để nhận được ID ảnh con hiện tại;

tạo ra (1105), bởi bộ xử lý, ảnh con hiện tại dựa trên ID ảnh con hiện tại để tạo chuỗi video.

2. Phương pháp theo điểm 1, trong đó bước giải mã ảnh con hiện tại bao gồm:

xác định (1105) lát nào thuộc về ảnh con nào trong một hoặc nhiều ảnh con đó dựa trên ID ảnh con hiện tại; và

định vị (1105) lát dựa trên kích thước ảnh con và vị trí ảnh con đối với ảnh con hiện tại.

3. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 2, trong đó vị trí ảnh con bao gồm khoảng cách dịch vị giữa mẫu phía trên cùng bên trái của ảnh con hiện tại và mẫu phía trên cùng bên trái của ảnh.

4. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó kích thước ảnh con bao gồm chiều cao ảnh con trong các mẫu độ chói và chiều rộng ảnh con trong các mẫu độ chói.

5. Phương pháp mã hóa được thực hiện ở bộ mã hóa, phương pháp này bao gồm các bước:

phân chia (1001), bởi bộ xử lý của bộ mã hóa, ảnh thành nhiều ảnh con;

phân chia (1001), bởi bộ xử lý, mỗi ảnh con trong số các ảnh con đó thành một hoặc nhiều lát;

mã hóa (1003) thành luồng bit, bởi bộ xử lý, tập hợp thông số chuỗi (SPS) bao gồm tập hợp đầy đủ của tất cả các bộ nhận dạng (ID) ảnh con đối với các ảnh con được phân chia từ ảnh, trong đó mỗi ảnh con tương ứng với ID ảnh con, và SPS còn bao gồm kích thước ảnh con và vị trí ảnh con cho mỗi ảnh con được phân chia từ ảnh;

mã hóa (1005) thành luồng bit, bởi bộ xử lý, các phần đầu lát bao gồm các ID ảnh con để chỉ thị các lát có trong các ảnh con được chỉ thị;

mã hóa (1007), bởi bộ xử lý, một hoặc nhiều ảnh con trong số các ảnh con đó thành luồng bit.

6. Phương pháp theo điểm 5, phương pháp này còn bao gồm bước:

lưu trữ (1009), trong bộ nhớ của bộ mã hóa, luồng bit cho việc truyền thông về phía bộ giải mã.

7. Phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 6, trong đó vị trí ảnh con bao gồm khoảng cách dịch vị giữa mẫu phía trên cùng bên trái của ảnh con tương ứng và mẫu phía trên cùng bên trái của ảnh.

8. Phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 7, trong đó kích thước ảnh con bao gồm chiều cao ảnh con trong các mẫu độ chói và chiều rộng ảnh con trong các mẫu độ chói.

9. Phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 8, trong đó sự kết hợp của các ảnh con đó bao trùm toàn bộ diện tích của ảnh mà không có khoảng hở và không có khoảng chồng lấp.

10. Vật ghi đọc được bằng máy tính phi chuyển tiếp bao gồm sản phẩm chương trình máy tính để sử dụng bởi thiết bị tạo mã video, sản phẩm chương trình máy tính bao gồm các lệnh thực thi được bằng máy tính được lưu trữ trên vật ghi đọc được bằng máy tính phi chuyển tiếp sao cho khi được thực hiện bởi bộ xử lý sẽ khiến cho thiết bị tạo mã video thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 4.

11. Vật ghi đọc được bằng máy tính phi chuyển tiếp bao gồm sản phẩm chương trình máy tính để sử dụng bởi thiết bị tạo mã video, sản phẩm chương trình máy tính bao gồm các lệnh thực thi được bằng máy tính được lưu trữ trên vật ghi đọc được bằng máy tính phi

chuyển tiếp sao cho khi được thực hiện bởi bộ xử lý sẽ khiến cho thiết bị tạo mã video thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 9.

12. Bộ giải mã, bộ giải mã này bao gồm:

đơn vị thu (1211), được tạo cấu hình để thu luồng bit bao gồm tập hợp thông số chuỗi (SPS), một hoặc nhiều ảnh con được phân chia từ ảnh, và một hoặc nhiều phần đầu lát được liên kết với một hoặc nhiều lát; trong đó SPS bao gồm tập hợp đầy đủ của tất cả các bộ nhận dạng ảnh con, các ID, đối với một hoặc nhiều ảnh con, trong đó mỗi ảnh con tương ứng với ID ảnh con; và phần đầu lát bao gồm ID ảnh con hiện tại, trong đó ID ảnh con hiện tại chỉ thị ảnh con mà chứa lát được liên kết với phần đầu lát; trong đó SPS còn bao gồm kích thước ảnh con và vị trí ảnh con cho mỗi ảnh con được phân chia từ ảnh;

đơn vị giải mã (1215), được tạo cấu hình để:

giải mã SPS để nhận được ID ảnh con, kích thước ảnh con và vị trí ảnh con cho mỗi ảnh con trong số một hoặc nhiều ảnh con đó; và

giải mã các phần đầu lát để nhận được ID ảnh con hiện tại;

đơn vị tạo, được tạo cấu hình để tạo ảnh con hiện tại dựa trên ID ảnh con hiện tại để tạo chuỗi video.

13. Bộ giải mã theo điểm 12, trong đó bộ giải mã còn được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 4.

14. Bộ mã hóa, bộ mã hóa này bao gồm:

đơn vị phân chia (1201), được tạo cấu hình để:

phân chia ảnh thành nhiều ảnh con; và

phân chia mỗi ảnh con trong số các ảnh con đó thành một hoặc nhiều lát;

đơn vị mã hóa (1203), được tạo cấu hình để:

mã hóa thành luồng bit tập hợp thông số chuỗi (SPS), bao gồm tập hợp đầy đủ của tất cả các bộ nhận dạng (ID) ảnh con đối với các ảnh con được phân chia từ ảnh, trong đó mỗi ảnh con tương ứng với ID ảnh con, và SPS còn bao gồm kích thước ảnh con và vị trí ảnh con cho mỗi ảnh con được phân chia từ ảnh;

mã hóa thành dòng bit các phần đầu lát bao gồm các ID ảnh con chỉ thị các lát có trong các ảnh con được chỉ thị; và

mã hóa một hoặc nhiều ảnh con trong số các ảnh con đó thành luồng bit.

15. Bộ mã hóa theo điểm 14, bộ mã hóa này còn bao gồm:

đơn vị lưu trữ (1205), được tạo cấu hình để lưu trữ luồng bit cho việc truyền thông về phía bộ giải mã.

16. Bộ mã hóa theo điểm bất kỳ trong số các điểm từ 14 đến 15, trong đó bộ mã hóa còn được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 9.

17. Phương tiện lưu trữ phi chuyên tiếp mà bao gồm luồng bit được mã hóa, luồng bit này bao gồm một hoặc nhiều ảnh con được phân chia từ ảnh, và bao gồm một hoặc nhiều phần đầu lát được liên kết với một hoặc nhiều lát và tập hợp thông số chuỗi (SPS), trong đó luồng bit này khác biệt ở chỗ:

SPS bao gồm tập hợp đầy đủ của tất cả các bộ nhận dạng (ID) ảnh con đối với các ảnh con được phân chia từ ảnh, trong đó mỗi ảnh con tương ứng với ID ảnh con, và SPS còn bao gồm kích thước ảnh con và vị trí ảnh con cho mỗi ảnh con được phân chia từ ảnh.

1/12

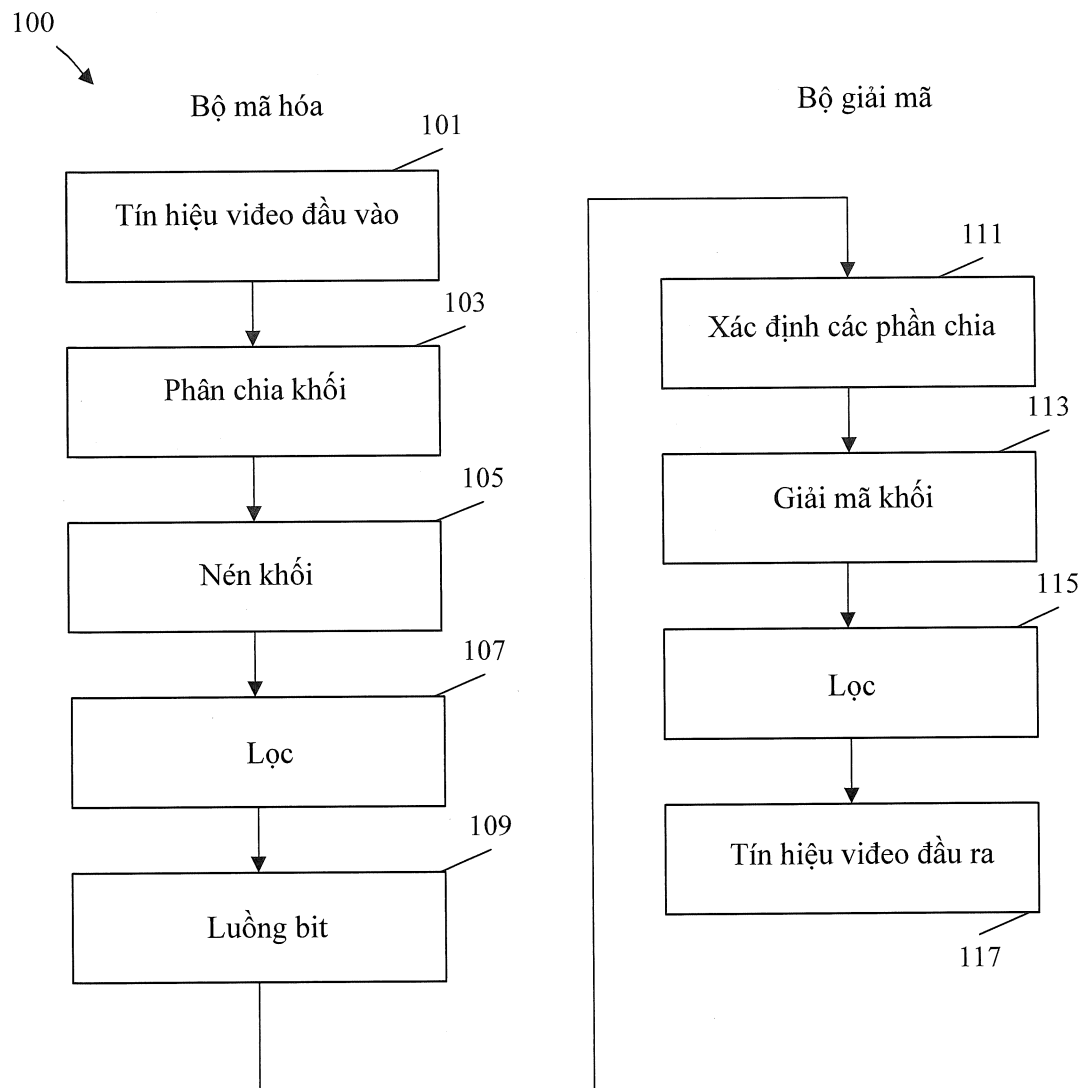


FIG. 1

2/12

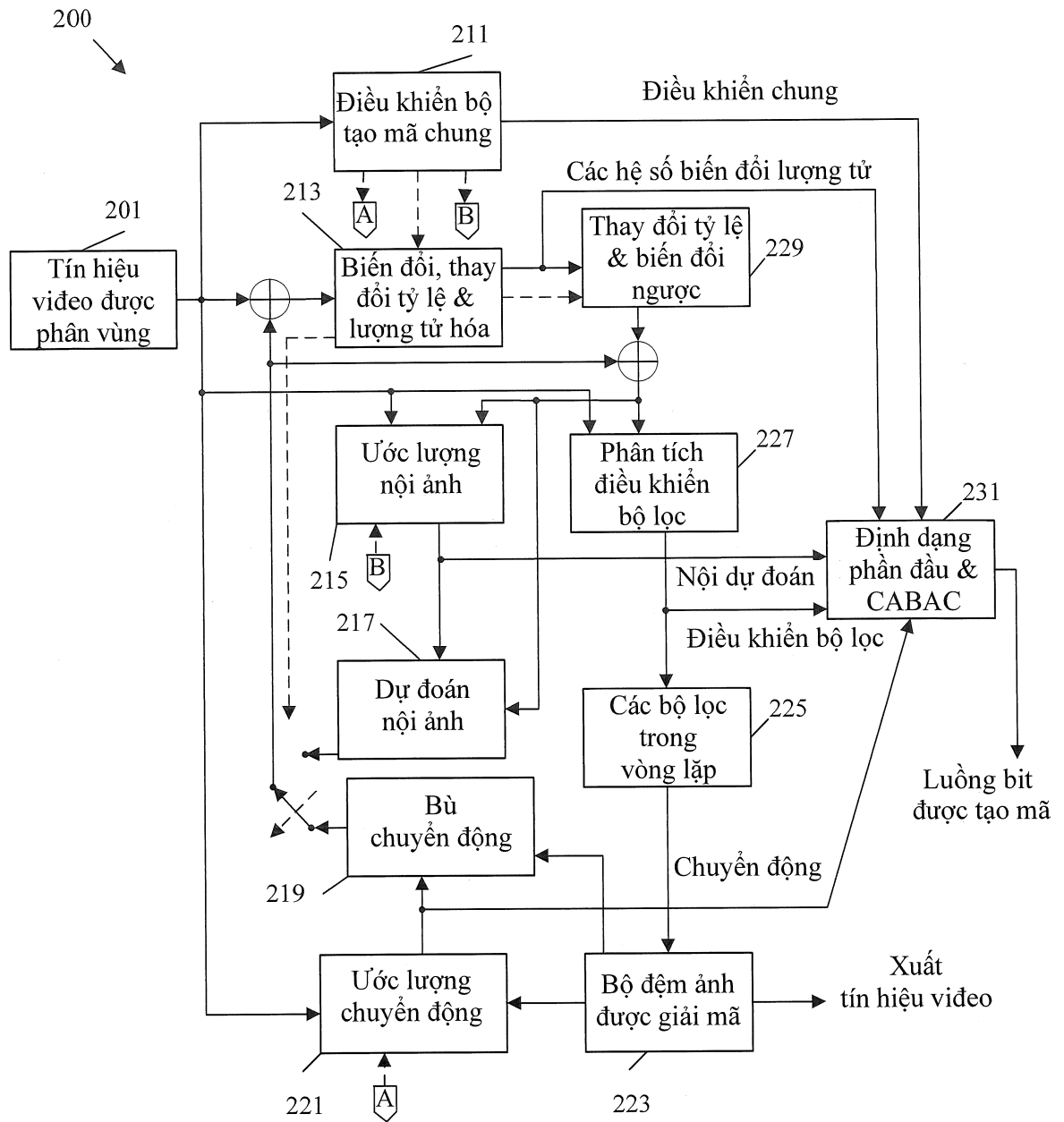


FIG. 2

3/12

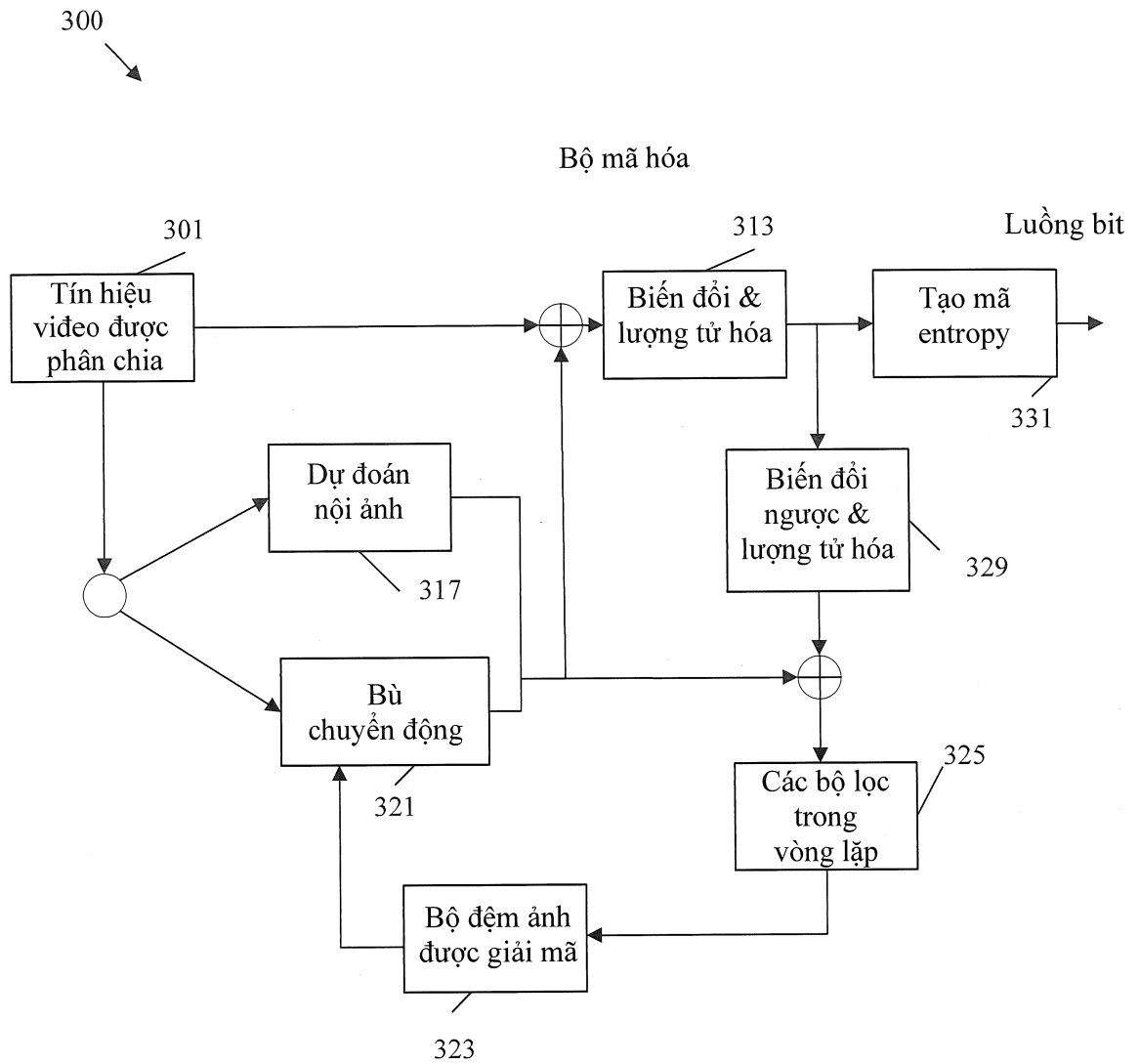


FIG. 3

4/12

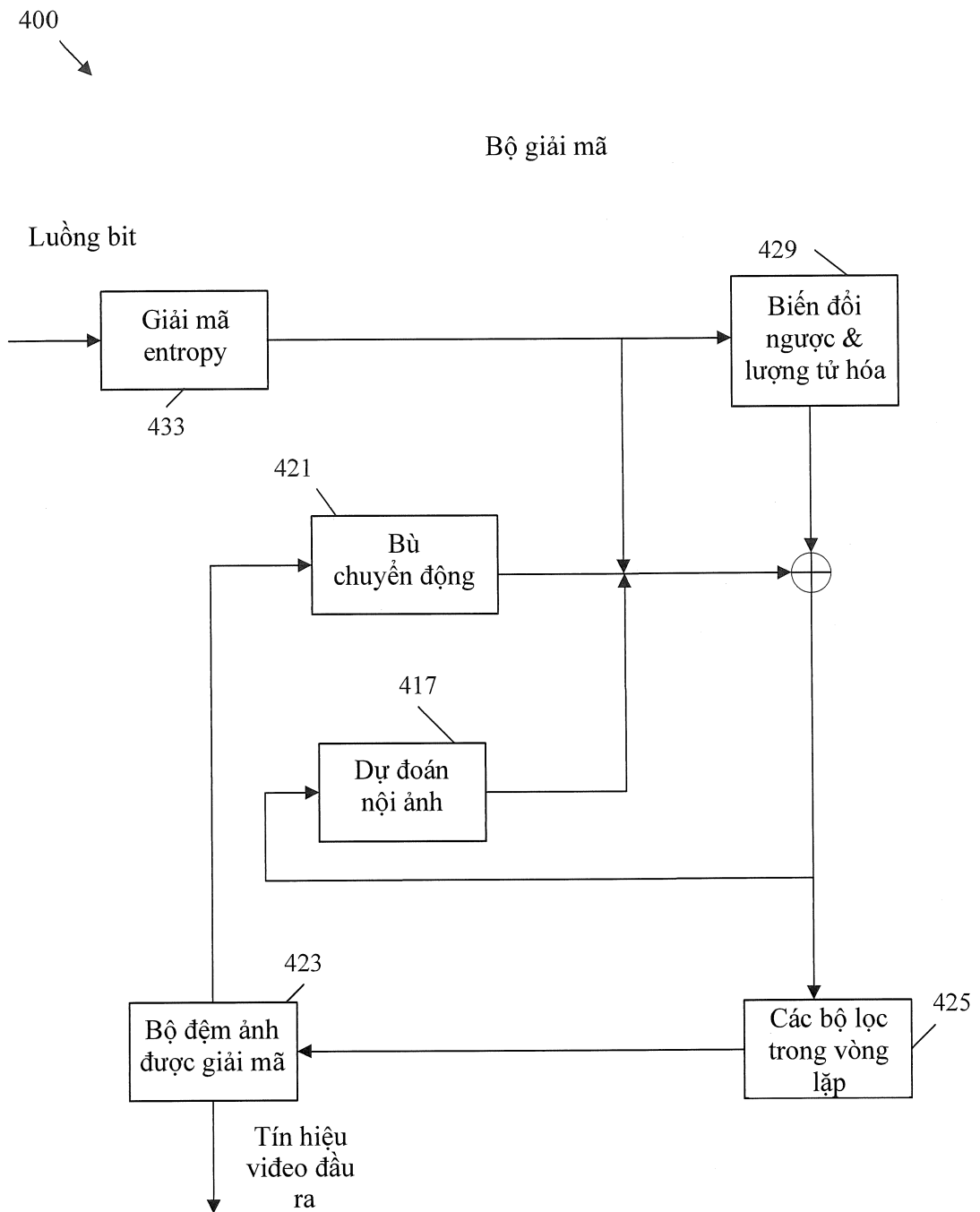


FIG. 4

5/12

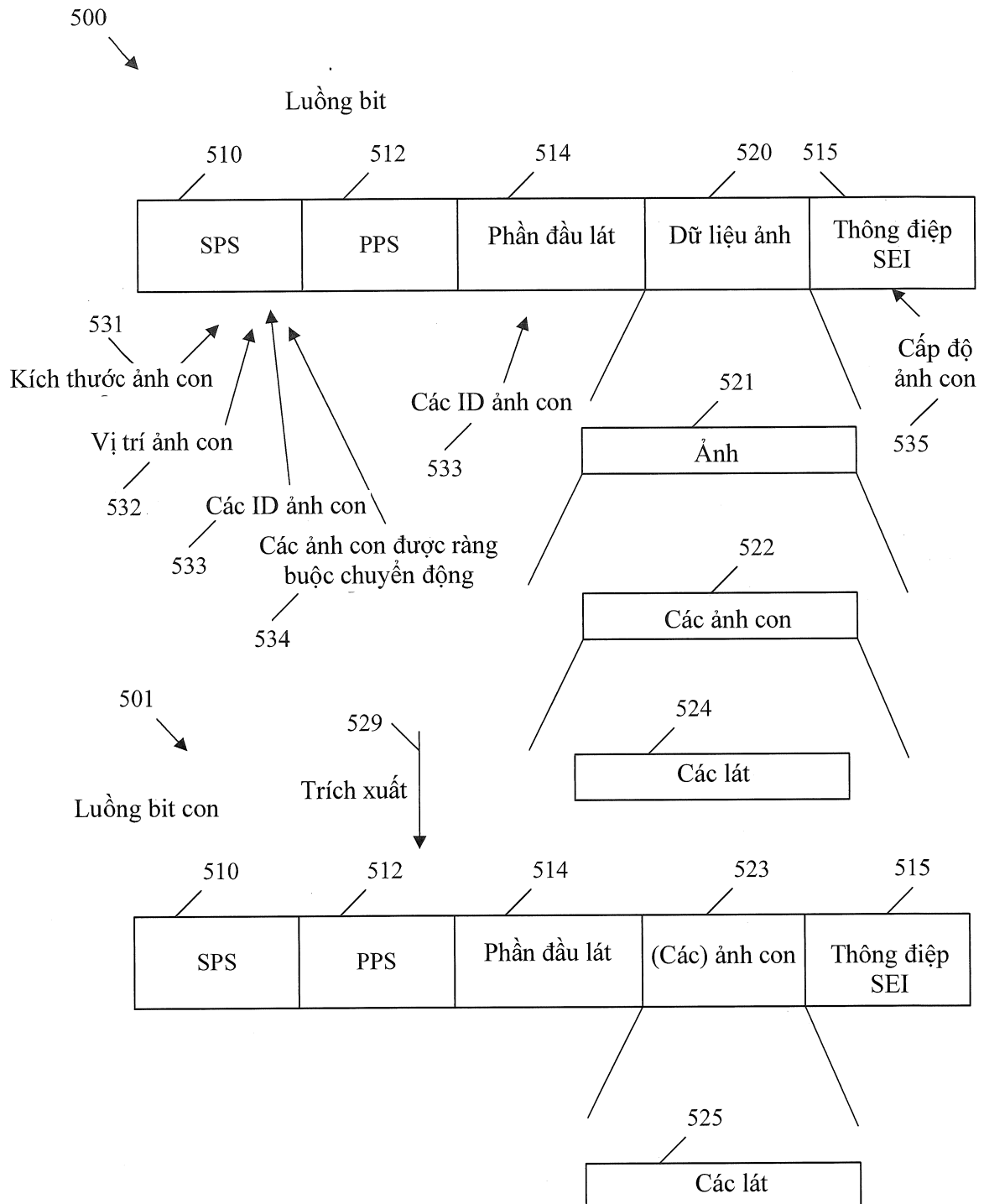


FIG. 5

6/12

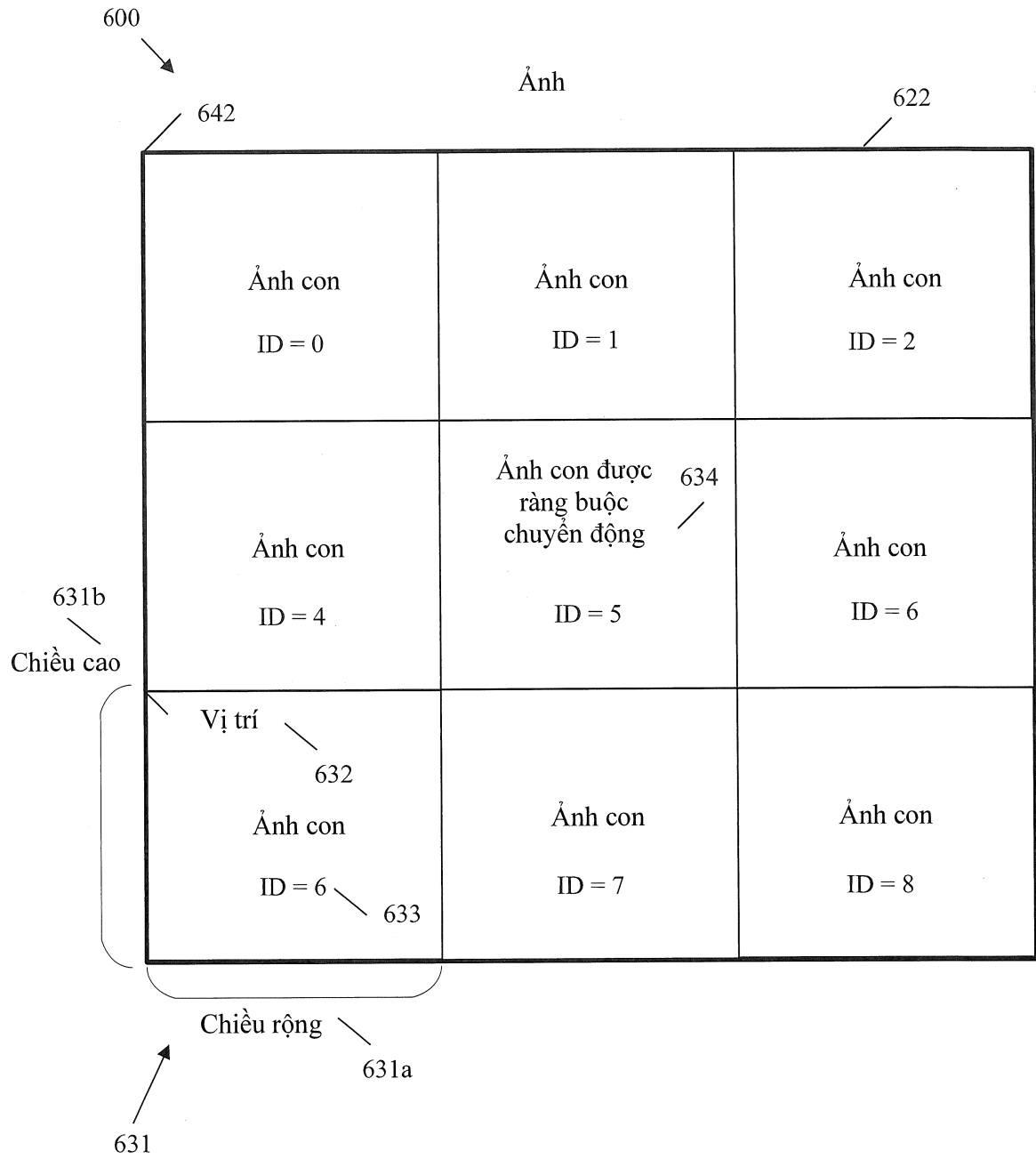


FIG. 6

7/12

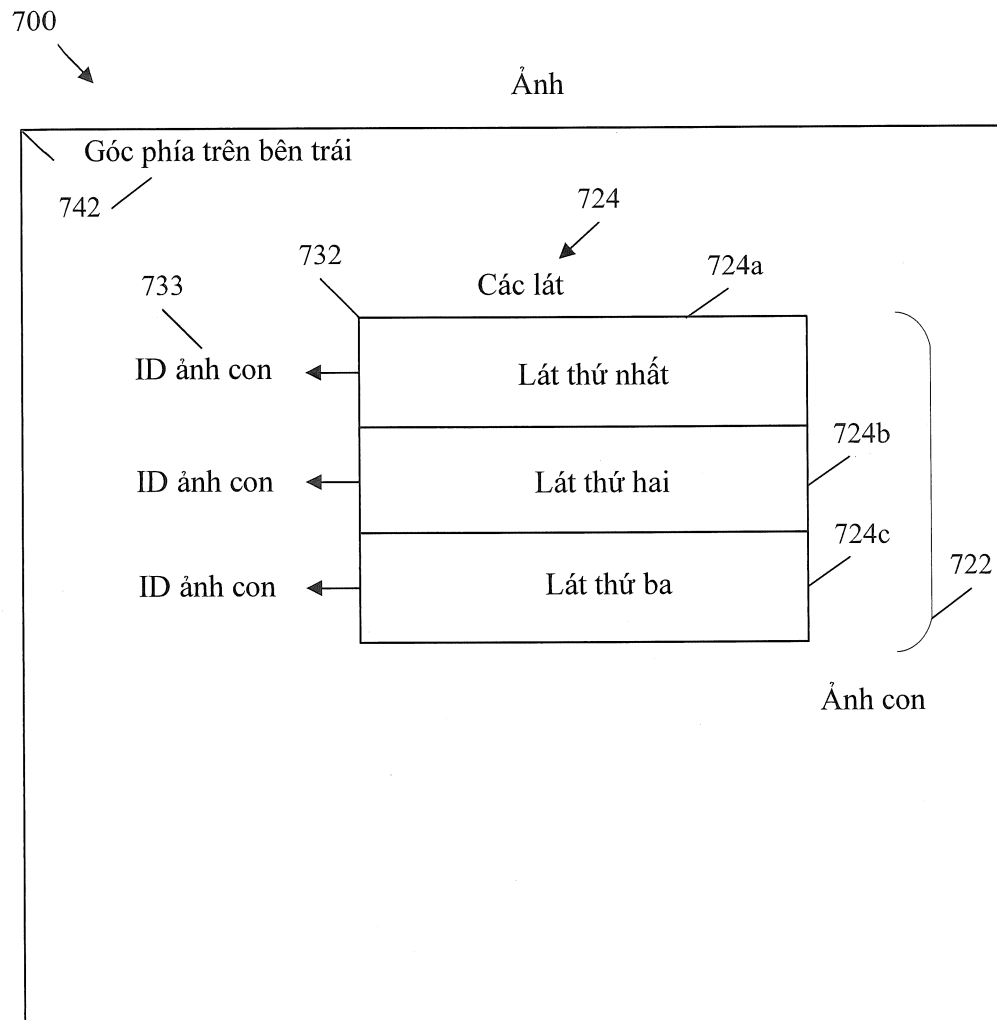


FIG. 7

8/12

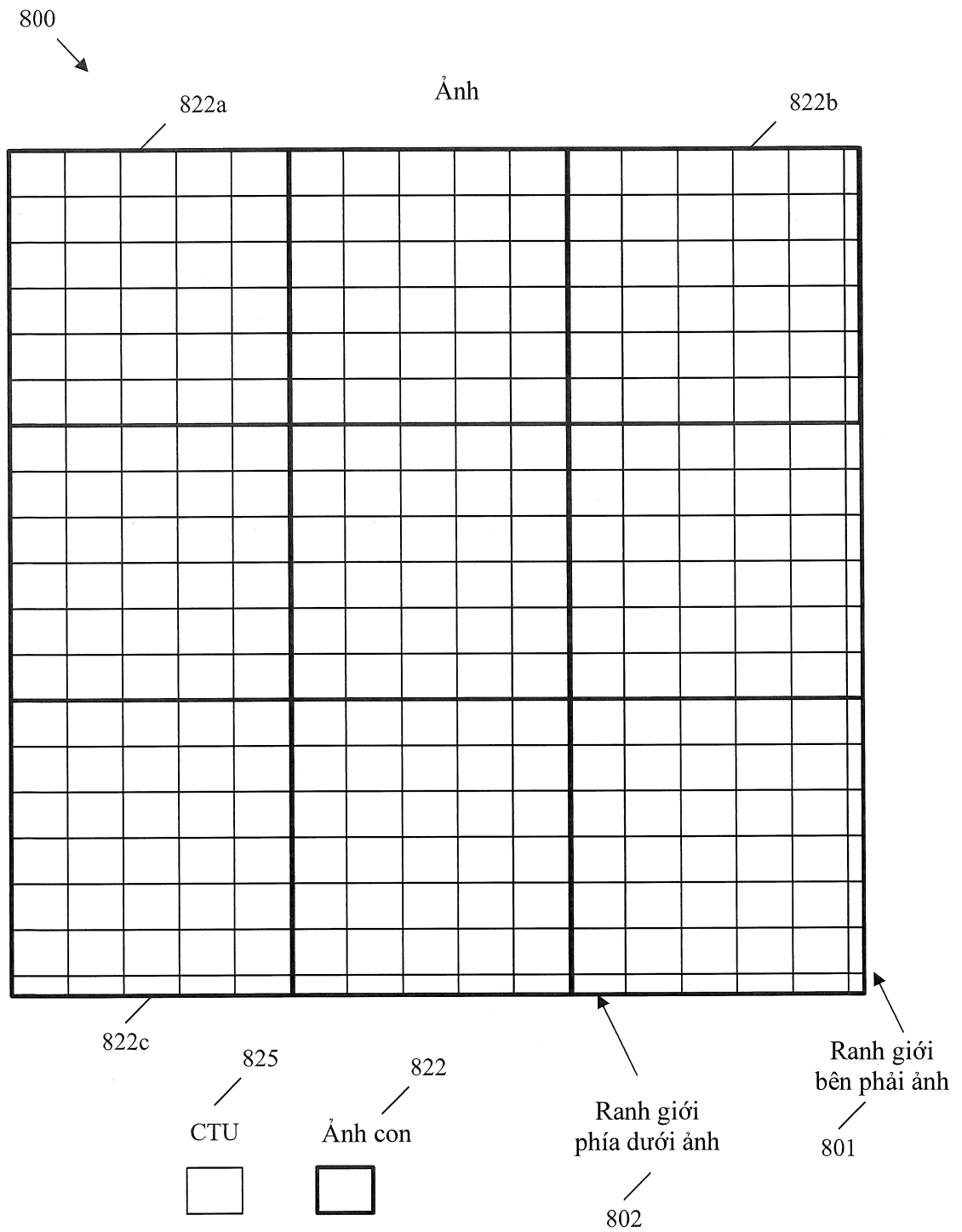


FIG. 8

9/12

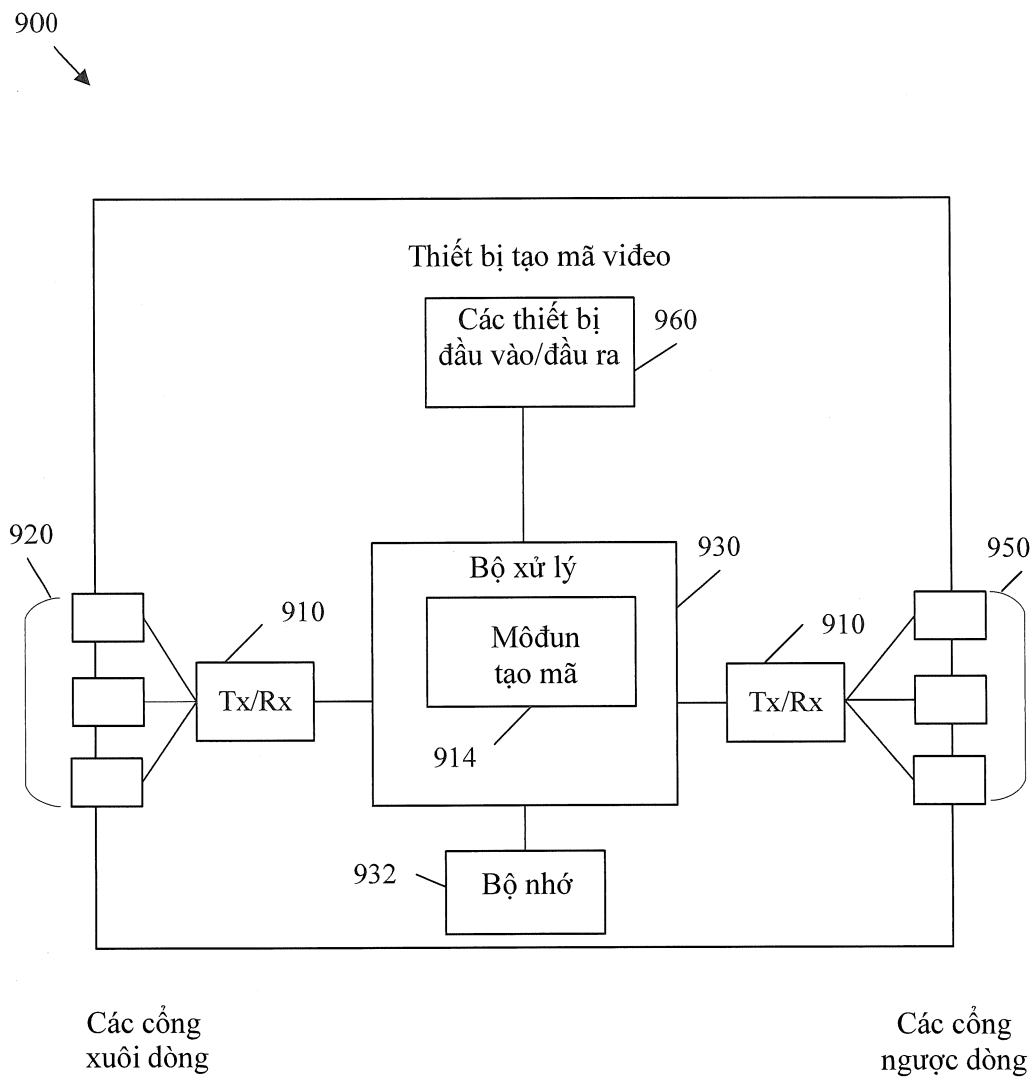


FIG. 9

10/12

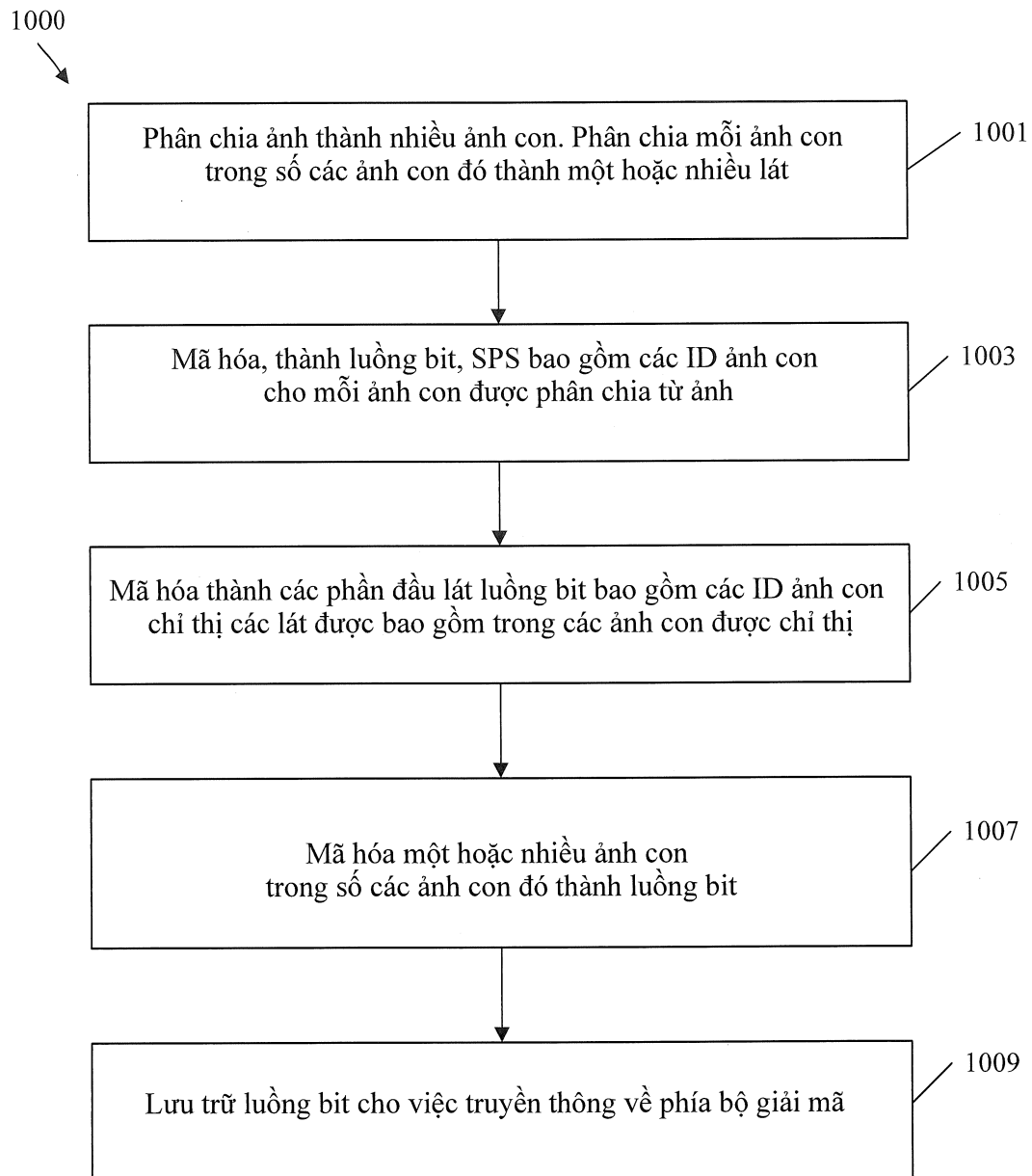


FIG. 10

11/12

1100

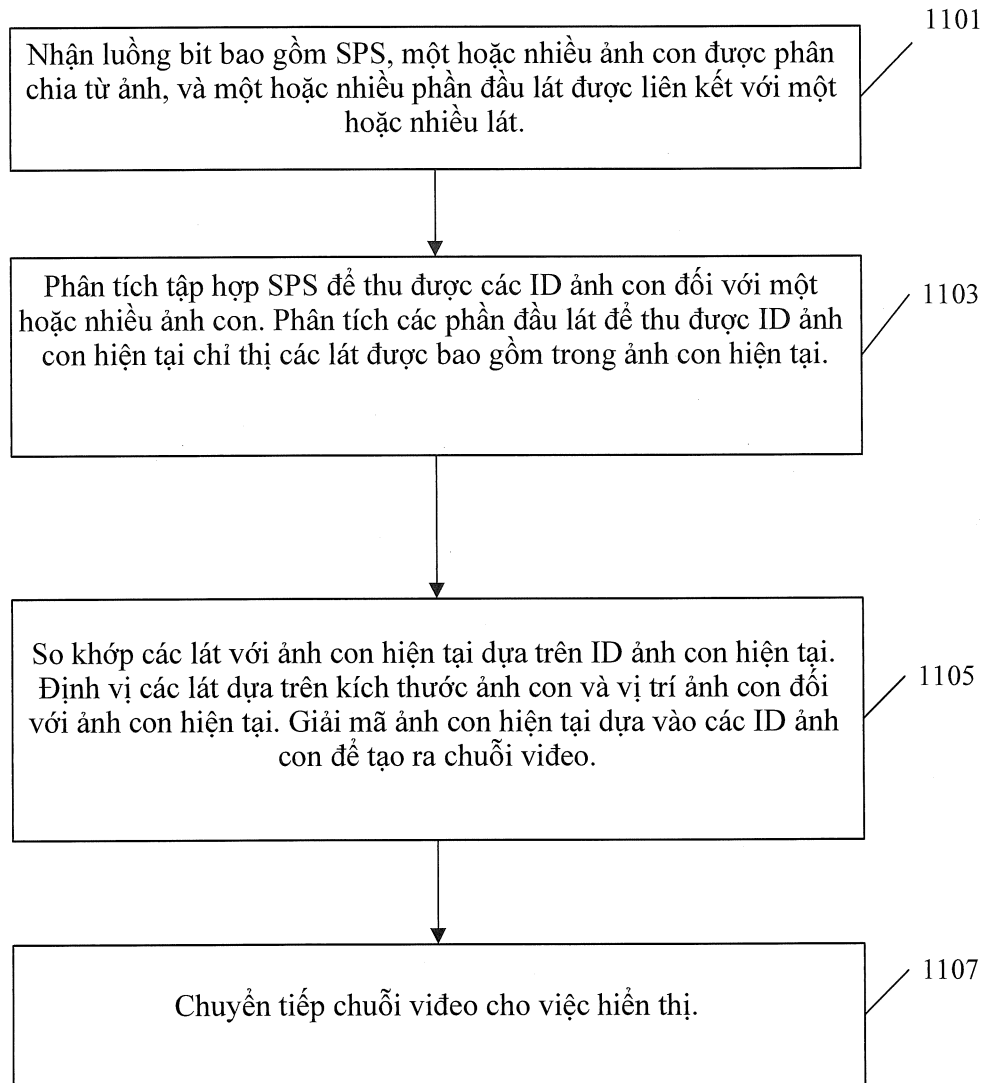


FIG. 11

12/12

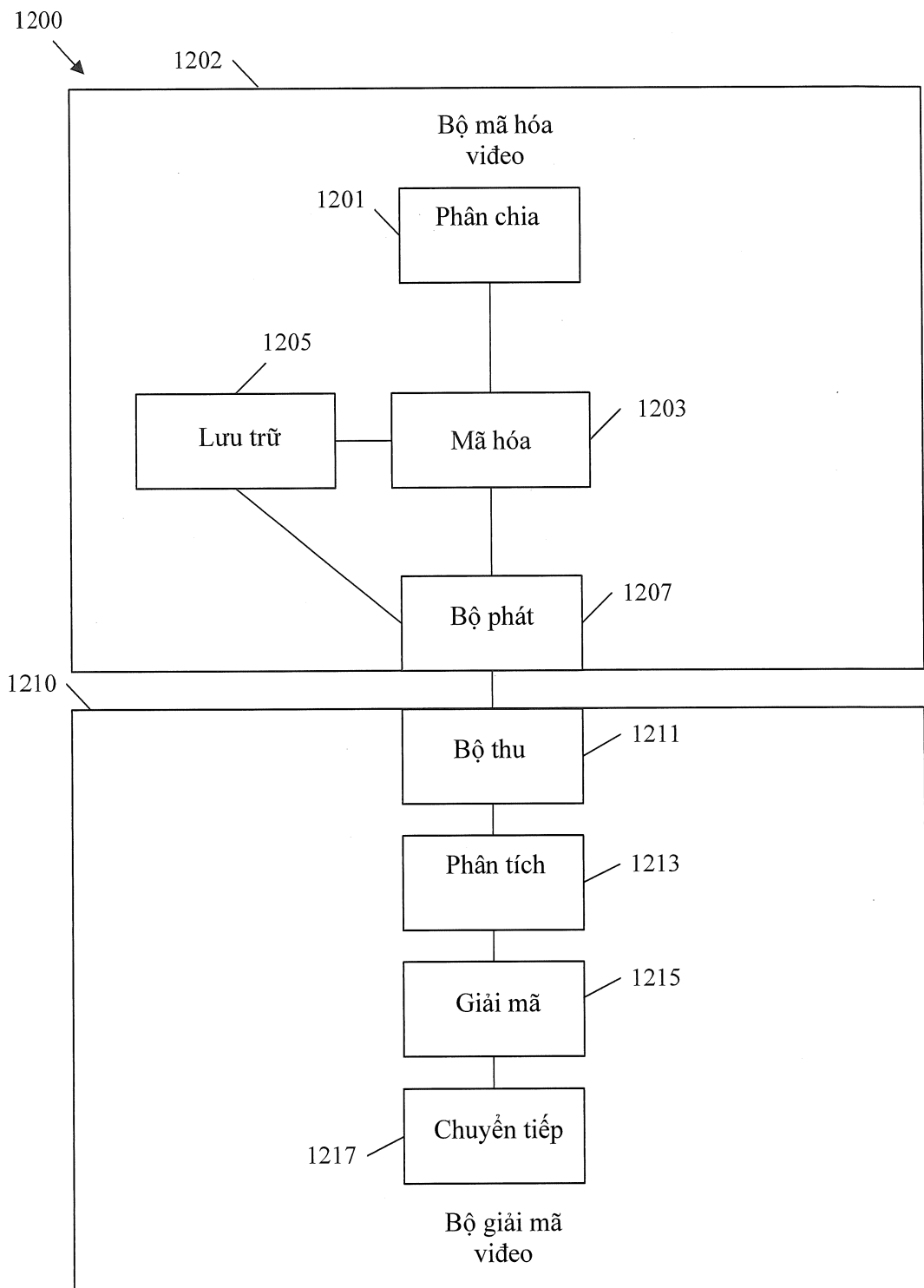


FIG. 12