



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

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

(11)



1-0048044

(51)^{2020.01} H04N 19/436; H04N 19/17

(13) B

(21) 1-2021-04860

(22) 09/01/2020

(86) PCT/US2020/012921 09/01/2020

(87) WO2020/146623 A1 16/07/2020

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

(45) 25/06/2025 447

(43) 25/11/2021 404A

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

Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong
518129, P. R. China

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

(74) Công ty TNHH một thành viên Sở hữu trí tuệ VCCI (VCCI-IP CO.,LTD)

(54) PHƯƠNG PHÁP GIẢI MÃ, PHƯƠNG PHÁP MÃ HÓA, PHƯƠNG TIỆN BẮT
BIẾN ĐƯỢC ĐƯỢC BỞI MÁY TÍNH, BỘ GIẢI MÃ VÀ BỘ MÃ HÓA

(21) 1-2021-04860

(57) Sáng chế đề cập đến cơ chế mã video. Cơ chế này bao gồm bước thu dòng bit bao gồm các ảnh con được phân chia từ ảnh sao cho sự liên kết của các ảnh con bao trùm tổng diện tích của ảnh mà không chồng lấn. Dòng bit được phân tách để thu nhận một hoặc nhiều ảnh con. Một hoặc nhiều ảnh con được giải mã để tạo chuỗi video. Chuỗi video được chuyển tiếp cho việc hiển thị. Sáng chế còn đề cập đến bộ mã hóa, bộ giải mã, phương pháp mã hóa, phương pháp giải mã, thiết bị mã video, và phương tiện bất biến đọc được bằng máy tính.

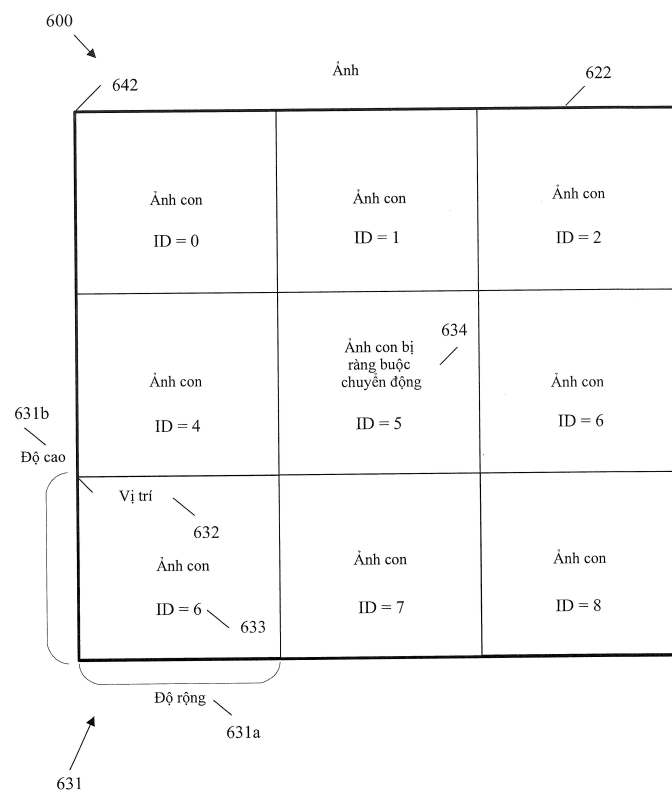


FIG. 6

Lĩnh vực kỹ thuật được đề cập

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

Tình trạng kỹ thuật của sáng chế

Lượng dữ liệu video được cần để mô tả thậm chí video tương đối ngắn có thể là đáng kể, mà có thể dẫn đến các sự khó khăn khi dữ liệu được tạo dòng hoặc theo cách khác được truyền thông qua mạng truyền thông với dung lượng độ rộng dải giới hạn. Vì vậy, dữ liệu video nói chung được nén trước khi được truyền thông qua các mạng viễn thông ngày nay. Kích thước của video cũng có thể là vấn đề khi video được lưu trữ trên thiết bị lưu trữ bởi vì các tài nguyên bộ nhớ có thể bị giới hạn. Các thiết bị nén video thường sử dụng phần mềm và/hoặc phần cứng ở nguồn để mã dữ liệu video trước khi truyền hoặc lưu trữ, nhờ vậy làm giảm số lượng dữ liệu được cần để biểu diễn các hình ảnh video kỹ thuật số. Dữ liệu được nén sau đó được thu ở đích bởi thiết bị giải nén video mà giải nén dữ liệu video. Với các tài nguyên mạng giới hạn và các nhu cầu ngày càng tăng về chất lượng video cao hơn, mong muốn các kỹ thuật nén và giải nén được nâng cao để nâng cao tỷ lệ nén với việc suy giảm ít hoặc không suy giảm chất lượng hình ảnh.

Bản chất kỹ thuật của sáng chế

Theo phương án, sáng chế bao gồm phương pháp được thực hiện ở bộ giải mã, trong đó phương pháp bao gồm các bước: thu, bởi bộ thu của bộ giải mã, dòng bit bao gồm các ảnh con được phân chia từ ảnh sao cho sự liên kết của các ảnh con bao trùm tổng diện tích của ảnh mà không chồng lấn; phân tách, bởi bộ xử lý của bộ giải mã, dòng bit để thu nhận một hoặc nhiều ảnh con; giải mã, bởi bộ xử lý, một hoặc nhiều ảnh con để tạo chuỗi video; và chuyển tiếp, bởi bộ xử lý, chuỗi video cho việc hiển thị. Một số hệ thống mã video cho phép các sơ đồ phân chia ảnh con bao gồm các khe hở và các sự chồng lấn. Điều này tạo tiềm năng cho các

nhóm ô/các lát được kết hợp với nhiều ảnh con. Nếu điều này được phép ở bộ mã hóa, thì các bộ giải mã phải được xây dựng để hỗ trợ sơ đồ mã như vậy thậm chí khi sơ đồ giải mã hiếm khi được sử dụng. Các ví dụ được bộc lộ rằng buộc các ảnh con để bao trùm ảnh mà không có khe hở hoặc chồng lấn. Bằng cách không cho phép các khe hở và các sự chồng lấn ảnh con, độ phức tạp của bộ giải mã có thể được giảm khi bộ giải mã không được yêu cầu để chiếm các khe hở và các sự chồng lấn tiềm năng khi xác định các kích thước và các vị trí ảnh con. Hơn nữa, việc không cho phép các khe hở và các sự chồng lấn ảnh con làm giảm độ phức tạp của các quy trình xử lý tối ưu hóa tốc độ biến dạng (rate distortion optimization - RDO) ở bộ mã hóa bởi vì bộ mã hóa có thể bỏ qua việc xem xét các trường hợp khe hở và chồng lấn khi lựa chọn việc mã hóa cho chuỗi video. Theo đó, việc tránh khỏi các khe hở và các sự chồng lấn có thể làm giảm việc sử dụng các tài nguyên bộ nhớ và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Một cách tùy chọn, theo bất kỳ trong số các khía cạnh nêu trên, cách thực hiện khác của khía cạnh đề xuất, trong đó sự liên kết của các ảnh con còn bao trùm tổng diện tích của ảnh mà không có khe hở.

Một cách tùy chọn, theo bất kỳ trong số các khía cạnh nêu trên, cách thực hiện khác của khía cạnh đề xuất, trong đó mỗi ảnh con bao gồm độ rộng ảnh con mà là bội số nguyên của kích thước đơn vị cây mã hóa (CTU) khi mỗi ảnh con bao gồm ranh giới bên phải mà không trùng khớp với ranh giới bên phải của ảnh.

Một cách tùy chọn, theo bất kỳ trong số các khía cạnh nêu trên, cách thực hiện khác của khía cạnh đề xuất, trong đó mỗi ảnh con bao gồm độ cao ảnh con mà là bội số nguyên của kích thước CTU khi mỗi ảnh con bao gồm ranh giới dưới cùng mà không trùng khớp với ranh giới dưới cùng của ảnh.

Một cách tùy chọn, theo bất kỳ trong số các khía cạnh nêu trên, cách thực hiện khác của khía cạnh đề xuất, trong đó dòng bit còn bao gồm tập hợp thông số chuỗi (sequence parameter set - SPS) bao gồm kích thước ảnh con cho các ảnh con và vị trí ảnh con cho mỗi trong số các ảnh con.

Một cách tùy chọn, theo bất kỳ trong số các khía cạnh nêu trên, cách thực hiện khác của khía cạnh đề xuất, trong đó SPS còn bao gồm các ký hiệu nhận dạng (identifier - ID) ảnh con cho mỗi trong số các ảnh con.

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

Theo phương án, sáng chế bao gồm phương pháp được thực hiện ở bộ mã hóa, trong đó phương pháp này bao gồm các bước: phân chia, bởi bộ xử lý của bộ mã hóa, ảnh thành các ảnh con sao cho sự liên kết của các ảnh con bao trùm tổng diện tích của ảnh mà không chồng lấn; mã hóa, bởi bộ xử lý, một hoặc nhiều trong số các ảnh con thành dòng bit; và lưu trữ, trong bộ nhớ của bộ mã hóa, dòng bit cho việc truyền thông về phía bộ giải mã. Một số hệ thống mã video cho phép các sơ đồ phân chia ảnh con bao gồm các khe hở và các sự chồng lấn. Điều này tạo tiềm năng cho các nhóm ô/các lát được kết hợp với nhiều ảnh con. Nếu điều này được phép ở bộ mã hóa, thì các bộ giải mã phải được xây dựng để hỗ trợ sơ đồ mã như vậy thậm chí khi sơ đồ giải mã hiếm khi được sử dụng. Các ví dụ được bộc lộ ràng buộc các ảnh con để bao trùm ảnh mà không có khe hở hoặc chồng lấn. Bằng cách không cho phép các khe hở và các sự chồng lấn ảnh con, độ phức tạp của bộ giải mã có thể được giảm khi bộ giải mã không được yêu cầu để chiếm các khe hở và các sự chồng lấn tiềm năng khi xác định các kích thước và các vị trí ảnh con. Hơn nữa, việc không cho phép các khe hở và các sự chồng lấn ảnh con làm giảm độ phức tạp của các quy trình xử lý tối ưu hóa tốc độ biến dạng (RDO) ở bộ mã hóa bởi vì bộ mã hóa có thể bỏ qua việc xem xét các trường hợp khe hở và chồng lấn khi lựa chọn việc mã hóa cho chuỗi video. Theo đó, việc tránh khỏi các khe hở và các sự chồng lấn có thể làm giảm việc sử dụng các tài nguyên bộ nhớ và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Một cách tùy chọn, theo bất kỳ trong số các khía cạnh nêu trên, cách thực hiện khác của khía cạnh đề xuất, trong đó sự liên kết của các ảnh con còn bao trùm tổng diện tích của ảnh mà không có khe hở.

Một cách tùy chọn, theo bất kỳ trong số các khía cạnh nêu trên, cách thực hiện khác của khía cạnh đề xuất, trong đó mỗi ảnh con bao gồm độ rộng ảnh con mà là bội số nguyên của kích thước CTU khi mỗi ảnh con bao gồm ranh giới bên phải mà không trùng khớp với ranh giới bên phải của ảnh.

Một cách tùy chọn, theo bất kỳ trong số các khía cạnh nêu trên, cách thực hiện khác của khía cạnh đề xuất, trong đó mỗi ảnh con bao gồm độ cao ảnh con mà là bội số nguyên của kích thước CTU khi mỗi ảnh con bao gồm ranh giới dưới cùng mà không trùng khớp với ranh giới dưới cùng của ảnh.

Một cách tùy chọn, theo bất kỳ trong số các khía cạnh nêu trên, cách thực hiện khác của khía cạnh đề xuất, còn bao gồm bước mã hóa, bởi bộ xử lý, SPS thành dòng bit, SPS bao gồm kích thước ảnh con cho các ảnh con và vị trí ảnh con cho mỗi trong số các ảnh con.

Một cách tùy chọn, theo bất kỳ trong số các khía cạnh nêu trên, cách thực hiện khác của khía cạnh đề xuất, trong đó SPS còn bao gồm các ID ảnh con cho mỗi trong số các ảnh con.

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

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

Theo phương án, sáng chế bao gồm phương tiện đọc được bởi máy tính bất biến bao gồm sản phẩm chương trình máy tính cho việc sử dụng bởi thiết bị mã video, sản phẩm chương trình máy tính bao gồm các lệnh thực hiện được bởi máy tính được lưu trữ trên phương tiện đọc được bởi máy tính bất biến sao cho khi được thực hiện bởi bộ xử lý khiến thiết bị mã video thực hiện phương pháp theo khía cạnh bất kỳ trong số các khía cạnh nêu trên.

Theo phương án, sáng chế bao gồm bộ giải mã bao gồm: phương tiện thu để thu dòng bit bao gồm các ảnh con được phân chia từ ảnh sao cho sự liên kết của các ảnh con bao trùm tổng diện tích của ảnh mà không chồng lấn; phương tiện phân tách để phân tách dòng bit để thu nhận một hoặc nhiều ảnh con; phương tiện giải mã để giải mã một hoặc nhiều ảnh con để tạo chuỗi video; và phương tiện chuyển tiếp để chuyển tiếp chuỗi video cho việc hiển thị.

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

Theo phương án, sáng chế bao gồm bộ mã hóa bao gồm: phương tiện phân chia để phân chia ảnh thành các ảnh con sao cho sự liên kết của các ảnh con bao trùm tổng diện tích của ảnh mà không chồng lấn; phương tiện mã hóa để mã hóa một hoặc nhiều trong số các ảnh con thành dòng bit; và phương tiện lưu trữ để lưu trữ dòng bit cho việc truyền thông về phía bộ giải mã.

Một cách tùy chọn, theo bất kỳ trong số các khía cạnh nêu trên, 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.

Nhằm mục đích rõ ràng, phương án bất kỳ trong số các phương án nêu trên có thể được kết hợp với bất kỳ một hoặc nhiều trong số các phương án nêu trên còn lại để tạo phương án mới nằm trong phạm vi của sáng chế.

Các dấu hiệu này và các dấu hiệu khác sẽ được hiểu một cách rõ ràng hơn từ phần mô tả chi tiết sau đây kết hợp với các hình vẽ và các điểm yêu cầu bảo hộ kèm theo.

Mô tả vắn tắt các hình vẽ

Để hiểu đầy đủ hơn về sáng chế, tham chiếu đến đây được thực hiện tới phần mô tả ngắn gọn sau đây, kết hợp với các hình vẽ và phần mô tả chi tiết kèm theo, trong đó các số chỉ dẫn giống nhau biểu diễn các phần giống nhau.

Fig.1 là lưu đồ của phương pháp ví dụ về việc mã tín hiệu video.

Fig.2 là hình vẽ giản lược của hệ thống mã và giải mã (bộ mã hóa/giải mã (codec)) ví dụ dùng cho việc mã video.

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

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

Fig.5 là hình vẽ giản lược minh họa dòng bit và dòng bit con ví dụ được trích từ dòng bit.

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

Fig.7 là hình vẽ giản lược minh họa cơ chế ví dụ để liên kết các lát với sự phối trí ảnh con.

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

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

Fig.10 là lưu đồ của phương pháp ví dụ về việc mã hóa dòng bit của các ảnh con với các sự ràng buộc vị trí.

Fig.11 là lưu đồ của phương pháp ví dụ về việc giải mã dòng bit của các ảnh con với các sự ràng buộc vị trí.

Fig.12 là hình vẽ giản lược của hệ thống ví dụ về việc truyền tín hiệu dòng bit của các ảnh con với các sự ràng buộc vị trí.

Mô tả chi tiết sáng chế

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

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

Nhiều kỹ thuật nén video có thể được ứng dụng để làm giảm kích thước của các tệp video với sự mất mát dữ liệu nhỏ nhất. Ví dụ, các kỹ thuật nén video có thể bao gồm bước thực hiện việc dự báo theo không gian (ví dụ, ảnh bên trong) và/hoặc việc dự báo theo thời gian (ví dụ, ảnh liên kết) để làm giảm hoặc loại bỏ phần dư dữ liệu trong các chuỗi video. Đối với việc mã video dựa vào khối, lát video (ví dụ, ảnh video hoặc 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 mã hóa (CTB), các đơn vị cây mã hóa (CTU), các đơn vị mã hóa (CU), và/hoặc các nút mã hóa. Các khối video trong lát (I) được mã hóa bên trong của ảnh được mã hóa nhờ sử dụng việc dự báo theo không gian đối với các mẫu tham chiếu trong các khối lân cận trong cùng ảnh. Các khối video trong lát (B) dự báo hai chiều hoặc (P) dự báo một chiều được mã hóa liên kết của ảnh có thể được mã hóa bằng cách ứng dụng việc dự báo theo không gian đối với các mẫu tham chiếu trong các khối lân cận trong cùng ảnh hoặc việc dự báo theo thời gian đối với các mẫu tham chiếu trong cách 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 báo theo không gian hoặc theo thời gian dẫn đến khối dự báo biểu diễn khối hình ảnh. Dữ liệu dư biểu diễn các độ chênh lệch điểm ảnh giữa khối hình ảnh gốc và khối dự báo. Theo đó, khối được mã hóa liên kết được mã hóa theo vector chuyển động mà chỉ tới khối của các mẫu tham chiếu tạo nên khối dự báo và dữ liệu dư chỉ báo độ chênh lệch giữa khối được mã hóa và khối dự

báo. Khối được mã hóa bên trong được mã hóa theo chế độ mã hóa bên trong và dữ liệu dư. Để nén thêm, dữ liệu dư có thể được biến đổi từ miền điểm ảnh đến miền biến đổi. Những điều này dẫn đến các hệ số biến đổi dư, mà có thể được lượng tử hóa. Các hệ số biến đổi được lượng tử hóa ban đầu có thể được sắp xếp trong mảng hai chiều. Các hệ số biến đổi được lượng tử hóa có thể được quét để tạo ra vectơ một chiều của các hệ số biến đổi. Việc mã entropi có thể được áp dụng để đạt được thậm chí việc nén nhiều hơn. Các kỹ thuật nén video như vậy được thảo luận chi tiết hơn dưới đây.

Để đảm bảo video được mã hóa có thể được giải mã một cách chính xác, video được mã hóa và được giải mã theo các chuẩn mã hóa video tương ứng. Các chuẩn mã hóa video bao gồm lĩnh vực tiêu chuẩn hóa liên minh viễn thông quốc tế (International Telecommunication Union - ITU) (ITU-T) H.261, nhóm các chuyên gia hình ảnh động (Motion Picture Experts Group - MPEG)-1 Tổ chức Tiêu chuẩn hóa quốc tế/Ủy ban Kỹ thuật điện quốc tế (International Organization for Standardization/International Electrotechnical Commission - ISO/IEC)-1 phần 2, ITU-T H.262 hoặc ISO/IEC MPEG-2 phần 2, ITU-T H.263, ISO/IEC MPEG-4 phần 2, mã hóa video nâng cao (Advanced Video Coding - AVC), cũng được biết đến là ITU-T H.264 hoặc ISO/IEC MPEG-4 phần 10, và mã hóa video hiệu quả cao (High Efficiency Video Coding - HEVC), cũng được biết đến là ITU-T H.265 hoặc MPEG-H phần 2. AVC bao gồm các sự mở rộng chẳng hạn như mã hóa video có thể mở rộng (Scalable Video Coding - SVC), mã hóa video nhiều màn hình (Multiview Video Coding - MVC) và mã hóa video nhiều màn hình cộng độ sâu (Multiview Video Coding plus Depth - MVC+D), và ba chiều (three dimensional - 3D) AVC (3D-AVC). HEVC bao gồm các sự mở rộng chẳng hạn như HEVC có thể mở rộng (Scalable HEVC - SHVC), HEVC nhiều màn hình (Multiview HEVC - MV-HEVC), và 3D HEVC (3D-HEVC). Nhóm các chuyên gia video liên kết (Joint Video Experts Team - JVET) của ITU-T và ISO/IEC đã bắt đầu phát triển chuẩn mã hóa video được gọi là mã hóa video đa năng (VVC). VVC được bao gồm trong bản dự thảo làm việc (WD), mà bao gồm JVET-L1001-v9.

Để mã hóa hình ảnh video, hình ảnh đầu tiên được phân chia, và các phần chia được mã hóa thành dòng bit. Các sơ đồ phân chia ảnh khác nhau khả dụng. Ví

dự, hình ảnh có thể được phân chia thành các lát thông thường, các lát phụ thuộc, các ô, và/hoặc theo việc xử lý song song mặt sóng (Wavefront Parallel Processing - WPP). Nhằm đơn giản, HEVC giới hạn các bộ mã hóa sao cho chỉ các lát thông thường, các lát phụ thuộc, các ô, WPP, và các sự kết hợp của chúng có thể được sử dụng khi phân chia lát thành các nhóm của các CTB để mã hóa video. Việc phân chia như vậy có thể được áp dụng để hỗ trợ việc so khớp kích thước bộ phận chuyển đổi lớn nhất (MTU), việc xử lý song song, và sự trì hoãn đầu cuối được giảm. MTU biểu thị lượng dữ liệu lớn nhất mà có thể được truyền trong gói đơn. Nếu tải trọng gói vượt quá MTU, thì tải trọng đó được phân tách thành hai gói qua quy trình xử lý được gọi là phân đoạn.

Lát thông thường, cũng được gọi một cách đơn giản là lát, là phần được phân chia của hình ảnh mà có thể được tái cấu trúc một cách độc lập với các lát thông thường khác nằm trong cùng ảnh, mặc dù một số sự phụ thuộc lẫn nhau do các thao tác lọc vòng lặp. Mỗi lát thông thường được đóng gói trong bộ phận lớp trừu tượng mạng (NAL) của nó dùng cho việc truyền. Hơn nữa, việc dự báo trong ảnh (dự báo mẫu bên trong, dự báo thông tin chuyển động, dự báo chế độ mã hóa) và sự phụ thuộc mã hóa entropi qua các ranh giới lát có thể được ngắt để hỗ trợ việc tái cấu trúc độc lập. Việc tái cấu trúc độc lập như vậy hỗ trợ phép song song. Ví dụ, phép song song dựa vào lát thông thường ứng dụng việc truyền thông bộ xử lý liên kết hoặc lõi liên kết nhỏ nhất. Tuy nhiên, với mỗi lát thông thường độc lập, mỗi lát được kết hợp với đoạn đầu lát riêng biệt. Việc sử dụng các lát thông thường có thể chịu phí tổn mã hóa đáng kể do chi phí bit của đoạn đầu lát đối với mỗi lát và do việc thiếu dự báo qua các ranh giới lát. Hơn nữa, các lát thông thường có thể được ứng dụng để hỗ trợ việc so khớp đối với các yêu cầu kích thước MTU. Cụ thể là, với lát thông thường được đóng gói trong đơn vị NAL tách biệt và có thể được mã hóa một cách độc lập, mỗi lát thông thường sẽ nhỏ hơn MTU trong các sơ đồ MTU để tránh khỏi việc chia lát thành các gói nhỏ hơn. Như vậy, mục đích của phép song song và mục đích của việc so khớp kích thước MTU có thể đưa ra các yêu cầu trái ngược với sự phối trí lát trong ảnh.

Các lát phụ thuộc giống với các lát thông thường, nhưng đã làm ngăn các đoạn đầu lát và cho phép phân chia các ranh giới khối cây hình ảnh mà không chia

việc dự báo bên trong ảnh. Theo đó, các lát phụ thuộc cho phép lát thông thường được phân đoạn thành nhiều đơn vị NAL, mà đưa ra sự trì hoãn đầu cuối được giảm bằng cách cho phép phần của lát thông thường được gửi ra ngoài trước khi việc mã hóa của toàn bộ lát thông thường hoàn tất.

Ô là phần được phân chia của hình ảnh được tạo bởi các ranh giới theo chiều ngang và theo chiều thẳng đứng mà tạo các cột và các hàng của các ô. Các ô có thể được mã hóa theo thứ tự quét mảnh (phải sang trái và trên xuống dưới). Thứ tự quét của các CTB là cục bộ nằm trong ô. Theo đó, các CTB ở ô thứ nhất được mã hóa theo thứ tự quét mảnh, trước khi đến các CTB ở ô tiếp theo. Giống với các lát thông thường, các ô chia các sự phụ thuộc dự báo trong ảnh cũng như các sự phụ thuộc giải mã entropi. Tuy nhiên, các ô có thể không được bao gồm trong các đơn vị NAL riêng, và do đó các ô có thể không được sử dụng cho việc so khớp kích thước MTU. Mỗi ô có thể được xử lý bởi một bộ xử lý/lỗi, và việc truyền thông bộ xử lý liên kết/lỗi liên kết được ứng dụng cho việc dự báo trong ảnh giữa các bộ phận xử lý giải mã các ô lân cận có thể bị giới hạn ở việc truyền đoạn đầu lát được chia sẻ (khi các ô liên kề trong cùng lát), và thực hiện việc chia sẻ liên quan đến việc lọc vòng lặp của các mẫu và siêu dữ liệu được tái cấu trúc. Khi nhiều hơn một ô được bao gồm trong lát, dịch vị byte điểm mục nhập cho mỗi ô khác ngoài dịch vị điểm mục nhập thứ nhất trong lát có thể được truyền tín hiệu ở đoạn đầu lát. Đối với mỗi lát và ô, ít nhất một trong số các điều kiện sau đây sẽ được hoàn tất: 1) tất cả các khối cây được mã hóa trong lát thuộc về cùng ô; và 2) tất cả các khối cây được mã hóa trong ô thuộc về cùng lát.

Trong WPP, hình ảnh được phân chia thành các hàng đơn của các CTB. Các cơ chế dự báo và giải mã entropi có thể sử dụng dữ liệu từ các CTB ở các hàng khác. Việc xử lý song song được làm cho khả thi qua việc giải mã song song của các hàng CTB. Ví dụ, hàng hiện thời có thể được giải mã song song với hàng trước đó. Tuy nhiên, việc giải mã của hàng hiện thời được trì hoãn từ quy trình giải mã của các hàng trước đó bởi hai CTB. Sự trì hoãn này đảm bảo rằng dữ liệu liên quan đến CTB ở trên và CTB ở trên và tới bên phải của CTB hiện thời trong hàng hiện thời là khả dụng trước khi CTB hiện thời được mã hóa. Cách tiếp cận này xuất hiện như mặt sóng khi được biểu diễn bằng đồ thị. Sự bắt đầu kinh ngạc này cho phép

phép song song với lên tới nhiều bộ xử lý/nhiều lõi khi hình ảnh chứa các hàng CTB. Bởi vì việc dự báo trong ảnh giữa các hàng khối cây lân cận nằm trong ảnh được phép, nên việc truyền thông bộ xử lý liên kết/lõi liên kết cho phép việc dự báo trong ảnh có thể là đáng kể. Việc phân chia WPP xem xét các kích thước đơn vị NAL. Do đó, WPP không hỗ trợ việc so khớp kích thước MTU. Tuy nhiên, các lát thông thường có thể được sử dụng kết hợp với WPP, với phí tổn mã hóa nhất định, để thực hiện việc so khớp kích thước MTU như mong muốn.

Các ô cũng có thể bao gồm các tập hợp ô bị ràng buộc chuyển động. Tập hợp ô bị ràng buộc chuyển động (MCTS) là tập hợp ô được thiết kế sao cho các vectơ chuyển động được kết hợp được giới hạn để chỉ tới các vị trí mẫu đầy đủ bên trong MCTS và tới các vị trí mẫu phân số mà yêu cầu chỉ các vị trí mẫu đầy đủ bên trong MCTS dùng cho việc nội suy. Hơn nữa, việc sử dụng các ứng viên vectơ chuyển động cho việc dự báo vectơ chuyển động theo thời gian được dẫn ra từ các khối bên ngoài MCTS không được phép. Cách này, mỗi MCTS có thể được giải mã một cách độc lập mà không có sự tồn tại của các ô không được bao gồm trong MCTS. Các bản tin thông tin nâng cao bổ sung (supplemental enhancement information - SEI) các MCTS theo thời gian có thể được sử dụng để chỉ báo sự tồn tại của các MCTS ở dòng bit và truyền tín hiệu các MCTS. Bản tin SEI các MCTS cung cấp thông tin bổ sung mà có thể được sử dụng trong việc trích dòng bit con MCTS (được quy định là phần ngữ nghĩa của bản tin SEI) để tạo ra dòng bit phù hợp cho tập hợp MCTS. Thông tin bao gồm một số tập hợp thông tin trích, đều định rõ một số tập hợp MCTS và chứa các byte tải trọng chuỗi byte thô (RBSP) của các tập hợp thông số video (video parameter set - VPS) thay thế, các tập hợp thông số chuỗi (SPS), và các tập hợp thông số hình ảnh (PPS) để được sử dụng trong suốt thời gian quy trình trích dòng bit con MCTS. Khi trích dòng bit con theo quy trình trích dòng bit con MCTS, các tập hợp thông số (các VPS, các SPS, và các PPS) có thể được ghi lại hoặc được thay thế, và các đoạn đầu lát có thể được cập nhật bởi vì một hoặc tất cả các thành phần cú pháp liên quan đến địa chỉ lát (bao gồm `first_slice_segment_in_pic_flag` và `slice_segment_address`) có thể ứng dụng các trị số khác nhau ở dòng bit con được trích.

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 đề cập đến PPS tách biệt và do đó có thể có phần chia ô tách biệt. Các ảnh con có thể được coi là các ảnh trong quy trình giải mã. Các ảnh con tham chiếu để giải mã ảnh con hiện thời được tạo ra bằng cách trích diện tích được sắp xếp với ảnh con hiện thời từ các ảnh tham chiếu trong bộ đệm ảnh được giải mã. Diện tích được trích được coi là ảnh con được giải mã. Việc dự báo liên kết có thể diễn ra giữa các ảnh con có cùng kích thước và cùng vị trí nằm trong ảnh. Nhóm ô, cũng được biết đến là lát, là chuỗi của các ô liên quan trong ảnh hoặc ảnh con. Nhiều mục có thể được dẫn ra để xác định vị trí của ảnh con trong ảnh. Ví dụ, mỗi ảnh con hiện thời có thể được bố trí ở vị trí không được chiếm giữ tiếp theo theo thứ tự quét mảnh CTU nằm trong ảnh mà đủ lớn để chứa ảnh con hiện thời nằm trong các ranh giới ảnh.

Hơn nữa, việc phân chia ảnh có thể dựa vào các ô mức ảnh và các ô mức chuỗi. Các ô mức chuỗi có thể bao gồm tính năng của MCTS, và có thể được thực hiện như các ảnh con. Ví dụ, ô mức ảnh có thể được định rõ là vùng hình chữ nhật của các khối cây lập mã nằm trong cột ô cụ thể và hàng ô cụ thể trong ảnh. Ô mức chuỗi có thể được định rõ là tập hợp của các vùng hình chữ nhật của các khối cây lập mã được bao gồm trong các khung khác nhau trong đó mỗi vùng hình chữ nhật còn bao gồm một hoặc nhiều ô mức ảnh và tập hợp của các vùng hình chữ nhật của các khối cây lập mã có thể giải mã được một cách độc lập với tập hợp khác bất kỳ của các vùng hình chữ nhật giống nhau. 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 truyền tín hiệu trong đơn vị NAL không có lớp mã hóa video (video coding layer - VCL) với ký hiệu nhận dạng (ID) được kết hợp ở đoạn đầu đơn vị NAL.

Sơ đồ phân chia dựa vào ảnh con trước đó có thể được kết hợp với các vấn đề nhất định. Ví dụ, khi các ảnh con được phép tạo ô nằm trong các ảnh con (việc phân chia của các ảnh con thành các ô) có thể được sử dụng để hỗ trợ việc xử lý song song. Việc phân chia ô của các ảnh con nhằm các mục đích xử lý song song có thể thay đổi từ ảnh đến ảnh (ví dụ, nhằm các mục đích cân bằng tải xử lý song song), và do đó có thể được quản lý ở mức 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 ứng dụng để hỗ trợ vùng quan tâm (ROI) và truy cập ảnh dựa vào ảnh con. Trong trường hợp như vậy, việc truyền tín hiệu của các ảnh con hoặc MCTS trong PPS là không hữu hiệu.

Theo ví dụ khác, khi ảnh con bất kỳ trong ảnh được mã hóa là ảnh con bị ràng buộc chuyển động theo thời gian, tất cả các ảnh con trong ảnh có thể được mã hóa là các ảnh con bị ràng buộc chuyển động theo thời gian. Việc phân chia ảnh như vậy có thể giới hạn. Ví dụ, việc mã ảnh con là ảnh con bị ràng buộc chuyển động theo thời gian có thể làm giảm hiệu quả mã hóa để đổi lại tính năng bổ sung. Tuy nhiên, trong các ứng dụng dựa vào vùng quan tâm, thường chỉ một hoặc một vài trong số các ảnh con sử dụng tính năng dựa vào ảnh con bị ràng buộc chuyển động theo thời gian. Do đó, các ảnh con còn lại chịu hiệu quả mã hóa bị giảm mà không đưa ra lợi ích thực tế bất kỳ.

Theo ví dụ khác, các thành phần cú pháp để xác định kích thước của ảnh con có thể được quy định trong các bộ phận của các kích thước CTU độ chói. Theo đó, cả độ rộng và độ cao ảnh con sẽ là bội số nguyên của `CtbSizeY`. Cơ chế của việc xác định độ rộng và độ 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 được tới các ảnh với độ rộng ảnh và/hoặc độ cao ảnh mà là bội số nguyên của `CtbSizeY`. Điều này làm cho việc phân chia ảnh con không khả dụng đối với các ảnh mà chứa các chiều mà không là các bội số nguyên của `CtbSizeY`. Nếu việc phân chia ảnh con đã được áp dụng tới độ rộng và/hoặc độ cao ảnh khi chiều ảnh không là bội số nguyên của `CtbSizeY`, thì việc dẫn ra của độ rộng ảnh con và/hoặc độ cao ảnh con trong các mẫu độ chói cho ảnh con ngoài cùng bên phải và ảnh con dưới cùng nhất sẽ không chính xác. Việc dẫn ra không chính xác như vậy sẽ khiến các kết quả lỗi trong một số công cụ mã hóa.

Theo ví dụ khác, vị trí của ảnh con trong ảnh có thể không được truyền tín hiệu. Thay vì vậy vị trí được dẫn ra nhờ sử dụng quy tắc sau đây. Ảnh con hiện thời được bố trí ở vị trí không được chiếm giữ như vậy tiếp theo theo thứ tự quét mảnh CTU nằm trong ảnh mà đủ lớn để chứa ảnh con nằm trong các ranh giới ảnh.

Việc dẫn 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 truyền, thì sau đó các vị trí của các ảnh con khác được dẫn ra một cách không chính xác và các mẫu được giải mã được đặt ở các vị trí lỗi. Vấn đề tương tự áp dụng khi các ảnh con đến theo thứ tự sai.

Theo ví dụ khác, việc giải mã ảnh con có thể yêu cầu việc trích các ảnh con được đồng bộ trí trong các ảnh tham chiếu. Điều này có thể áp đặt độ phức tạp bổ sung và gánh nặng kết quả về việc sử dụng tài nguyên bộ xử lý và bộ nhớ.

Theo ví dụ khác, khi ảnh con được ấn định là ảnh con bị ràng buộc chuyển động theo thời gian, các bộ lọc vòng lặp mà đi qua ranh giới ảnh con được ngắt. Điều này xảy ra không quan tâm đến việc xem các bộ lọc vòng lặp mà đi qua các ranh giới ô được phép hay không. Sự ràng buộc như vậy có thể quá hạn chế và có thể dẫn đến các rìa trực quan cho các ảnh video ứng dụng bội số của các ảnh con.

Theo ví dụ khác, mối tương quan giữa SPS, STGPS, PPS và các đoạn đầu nhóm ô như sau. STGPS đề cập đến SPS, PPS đề cập đến STGPS, và các đoạn đầu nhóm ô/các đoạn đầu lát đề cập đến PPS. Tuy nhiên, STGPS và PPS sẽ trực giao hơn là PPS đề cập đến STGPS. Sự sắp xếp trước đó cũng có thể không cho phép tất cả các nhóm ô của cùng ảnh đề cập đến cùng PPS.

Theo ví dụ khác, mỗi STGPS có thể chứa các ID cho bốn cạnh của ảnh con. Các ID như vậy được sử dụng để nhận dạng các ảnh con mà dùng chung cùng ranh giới sao cho mối tương quan không gian tương đối của chúng có thể được định rõ. Tuy nhiên, thông tin như vậy có thể không đủ để dẫn ra thông tin vị trí và kích thước cho tập hợp nhóm ô mức chuỗi trong một số trường hợp. Trong các trường hợp khác, việc truyền tín hiệu thông tin vị trí và kích thước có thể là thừa.

Theo ví dụ khác, STGPS ID có thể được truyền tín hiệu ở đoạn đầu đơn vị NAL của bộ phận VCL NAL nhờ sử dụng tám bit. Điều này có thể hỗ trợ với việc trích ảnh con. Việc truyền tín hiệu như vậy có thể làm tăng một cách không cần thiết độ dài của đoạn đầu đơn vị NAL. Vấn đề khác là nếu các tập hợp nhóm ô mức chuỗi không bị ràng buộc để ngăn ngừa các sự chồng lấn, thì một nhóm ô có thể được kết hợp với nhiều tập hợp nhóm ô mức chuỗi.

Được bộc lộ ở đây là các cơ chế khác nhau để xử lý một hoặc nhiều trong số các vấn đề nêu trên. Theo ví dụ thứ nhất, thông tin phối trí cho các ảnh con được bao gồm trong SPS thay vì PPS. Thông tin phối trí ảnh con bao gồm vị trí ảnh con và kích thước ảnh con. Vị trí ảnh con là dịch vị giữa mẫu trên cùng bên trái của ảnh con và mẫu trên cùng bên trái của ảnh. Kích thước ảnh con là độ cao và độ rộng của ảnh con khi 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 là các ô có thể thay đổi từ ảnh đến ảnh. Tuy nhiên, các ảnh con có thể được sử dụng để hỗ trợ các ứng dụng ROI và truy cập dựa vào ảnh con. Các chức năng này không thay đổi trên cơ sở mỗi ảnh. Hơn nữa, chuỗi video có thể bao gồm SPS đơn (hoặc một trên mỗi phân đoạn video), và có thể bao gồm nhiều nhất một PPS trên mỗi ảnh. Việc đặt thông tin phối trí cho các ảnh con trong SPS đảm bảo rằng sự phối trí chỉ được truyền tín hiệu một lần cho chuỗi/phân đoạn khác ngoài được truyền tín hiệu dư cho mỗi PPS. Theo đó, việc truyền tín hiệu sự phối trí ảnh con trong SPS làm tăng hiệu quả mã hóa và do đó làm giảm việc sử dụng 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 dẫn ra bởi bộ giải mã. Việc truyền tín hiệu thông tin ảnh con làm giảm khả năng lỗi trong trường hợp của các gói bị mất và hỗ trợ tính năng bổ sung về việc trích các ảnh con. Theo đó, việc truyền tín hiệu sự phối trí ảnh con trong SPS nâng cao tính năng của bộ mã hóa và/hoặc bộ giải mã.

Theo ví dụ thứ hai, các độ rộng ảnh con và các độ cao ảnh con bị ràng buộc ở các bội số của kích thước CTU. Tuy nhiên, các sự ràng buộc này được loại bỏ khi ảnh con lần lượt được bố trí ở ranh giới bên phải của ảnh hoặc ranh giới dưới cùng của ảnh. Như được lưu ý ở trên, một số hệ thống video có thể giới hạn các ảnh con để bao gồm các độ cao và các độ rộng mà là các bội số của kích thước CTU. Điều này ngăn ngừa các ảnh con khỏi thao tác một cách chính xác với nhiều sự phối trí ảnh. Bằng cách cho phép các ảnh con dưới cùng và bên phải lần lượt bao gồm các độ cao và các độ rộng, mà không là các bội số 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 tăng tính năng bộ mã hóa và bộ giải mã. Hơn nữa, tính năng được tăng cho phép bộ mã hóa mã hóa các ảnh một cách hữu hiệu hơn, mà làm giảm việc

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

Theo ví dụ thứ ba, các ảnh con bị ràng buộc để bao trùm ảnh mà không có khe hở hoặc chồng lấn. Như được lưu ý ở trên, một số hệ thống mã video cho phép các ảnh con bao gồm các khe hở và các sự chồng lấn. Điều này tạo tiềm năng cho các nhóm ô/các lát được kết hợp với nhiều ảnh con. Nếu điều này được phép ở bộ mã hóa, thì các bộ giải mã phải được xây dựng để hỗ trợ sơ đồ mã như vậy thậm chí khi sơ đồ giải mã hiếm khi được sử dụng. Bằng cách không cho phép các khe hở và các sự chồng lấn ảnh con, độ phức tạp của bộ giải mã có thể được giảm khi bộ giải mã không được yêu cầu để chiếm các khe hở và các sự chồng lấn tiềm năng khi xác định các kích thước và các vị trí ảnh con. Hơn nữa, việc không cho phép các khe hở và các sự chồng lấn ảnh con làm giảm độ phức tạp của các quy trình xử lý tối ưu hóa tốc độ biến dạng (RDO) ở bộ mã hóa bởi vì bộ mã hóa có thể bỏ qua việc xem xét các trường hợp khe hở và chồng lấn khi lựa chọn việc mã hóa cho chuỗi video. Theo đó, việc tránh khỏi các khe hở và các sự chồng lấn có thể làm giảm việc sử dụng các tài nguyên bộ nhớ và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Theo ví dụ thứ tư, có thể được truyền tín hiệu trong SPS để chỉ báo khi nào ảnh con là ảnh con bị ràng buộc chuyển động theo thời gian. Như được lưu ý ở trên, một số hệ thống có thể thiết đặt chung tất cả các ảnh con là các ảnh con bị ràng buộc chuyển động theo thời gian hoặc hoàn toàn không cho phép việc sử dụng các ảnh con bị ràng buộc chuyển động theo thời gian. Các ảnh con bị ràng buộc chuyển động theo thời gian như vậy cung cấp tính năng trích độc lập với chi phí của hiệu quả mã hóa được giảm. Tuy nhiên, trong các ứng dụng dựa vào vùng quan tâm, vùng quan tâm sẽ được mã hóa cho việc trích độc lập trong khi các vùng bên ngoài của vùng quan tâm không cần tính năng như vậy. Do đó, các ảnh con còn lại chịu hiệu quả mã hóa bị giảm mà không đưa ra lợi ích thực tế bất kỳ. Theo đó, có thể cho phép hỗn hợp của các ảnh con bị ràng buộc chuyển động theo thời gian mà cung cấp tính năng trích độc lập và các ảnh con không bị ràng buộc chuyển động đối với hiệu quả mã hóa được tăng khi việc trích độc lập không được mong muốn. Do đó, có thể cho phép tính năng được tăng và/hoặc hiệu quả mã hóa được tăng, mà

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

Theo ví dụ thứ năm, tập hợp của các ID ảnh con hoàn thiện được truyền tín hiệu trong SPS, và các đoạn đầu lát bao gồm ID ảnh con chỉ báo ảnh con mà chứa các lát tương ứng. Như được lưu ý ở trên, một số hệ thống truyền tín hiệu các vị trí ảnh con liên quan đến các ảnh con khác. Điều này gây ra vấn đề nếu các ảnh con bị mất hoặc được trích một cách riêng biệt. Bằng cách ấn định mỗi ảnh con bởi ID, các ảnh con có thể được bố trí và được định kích thước mà không tham chiếu tới các ảnh con khác. Điều này đến lượt hỗ trợ việc điều chỉnh lỗi cũng như các ứng dụng mà chỉ trích một vài trong số các ảnh con và tránh khỏi việc truyền các ảnh con khác. Danh mục của tất cả các ID ảnh con hoàn thiện có thể được gửi trong SPS cùng với thông tin định kích thước thích hợp. Mỗi đoạn đầu lát có thể chứa ID ảnh con chỉ báo ảnh con mà bao gồm lát tương ứng. Theo cách này, các ảnh con và các lát tương ứng có thể được trích và được bố trí mà không tham chiếu tới các ảnh con khác. Do đó, các ID ảnh con hỗ trợ tính năng được tăng và/hoặc hiệu quả mã hóa được tăng, mà làm giảm việc sử dụng các tài nguyên mạng, các tài nguyên bộ nhớ, và/hoặc các tài nguyên xử lý ở bộ mã hóa và bộ giải mã.

Theo ví dụ thứ sáu, các mức được truyền tín hiệu cho mỗi ảnh con. Trong một số hệ thống mã video các mức được truyền tín hiệu cho các ảnh. Mức chỉ báo các tài nguyên phần cứng được cần để giải mã ảnh. Như được lưu ý ở trên, các ảnh con khác nhau có thể có tính năng khác nhau trong một số trường hợp và do đó có thể được xử lý khác nhau trong suốt thời gian quy trình mã hóa. Như vậy, mức dựa vào ảnh có thể không hữu dụng để giải mã một số ảnh con. Do đó, sáng chế bao gồm các mức cho mỗi ảnh con. Theo cách này, mỗi ảnh con có thể được mã hóa một cách độc lập với các ảnh con khác mà không đòi hỏi quá nhiều một cách không cần thiết bộ giải mã bằng cách thiết đặt các yêu cầu giải mã quá cao cho các ảnh con được mã hóa theo các cơ chế ít phức tạp hơn. Thông tin mức ảnh con được truyền tín hiệu hỗ trợ tính năng được tăng và/hoặc hiệu quả mã hóa được tăng, mà làm giảm việc sử dụng 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 thao tác 100 ví dụ về việc mã tín hiệu video. Cụ thể là, tín hiệu video được mã hóa ở bộ mã hóa. Quy trình mã hóa nén tín hiệu video bằng cách ứng dụng các cơ chế khác nhau để làm giảm kích thước tệp video. Kích thước tệp nhỏ hơn cho phép tệp video được nén được truyền tới người dùng, trong khi làm giảm phí tồn độ rộng dải được kết hợp. Bộ giải mã sau đó giải mã tệp video được nén để tái cấu trúc tín hiệu video gốc cho việc hiển thị tới người dùng cuối. Quy trình giải mã nói chung phản chiếu quy trình mã hóa để cho phép bộ giải mã tái cấu trúc một cách phù hợp tín hiệu video.

Ở bước 101, tín hiệu video được đưa vào bộ mã hóa. Ví dụ, tín hiệu video có thể là tệp video không được nén được lưu trữ trong bộ nhớ. Theo ví dụ khác, tệp video có thể được chụp bởi thiết bị chụp video, chẳng hạn như camera video, và được mã hóa để hỗ trợ việc tạo dòng trực tiếp của video. Tệp video có thể bao gồm cả thành phần audio và thành phần video. Thành phần video chứa hàng loạt các khung hình ảnh mà, khi được xem trong chuỗi, đưa ra ấn tượng trực quan của sự chuyển động. Các khung chứa các điểm ảnh mà được thể hiện dưới dạng ánh sáng, ở đây được gọi là các thành phần độ chói (hoặc các mẫu độ chói), và màu sắc, mà được gọi là các thành phần sắc độ (hoặc các mẫu màu). Theo một số ví dụ, các khung cũng có thể chứa các trị số độ sâu để hỗ trợ việc xem ba chiều.

Ở bước 103, video được phân chia thành các khối. Việc phân chia bao gồm việc chia nhỏ các điểm ảnh trong mỗi khung thành các khối vuông và/hoặc hình chữ nhật dùng cho việc nén. Ví dụ, trong mã hóa video hiệu quả cao (HEVC) (cũng được biết đến là H.265 và MPEG-H phần 2) khung đầu tiên có thể được chia thành các đơn vị cây mã hóa (CTU), mà là các khối có kích thước được định rõ trước (ví dụ, sáu mươi tư điểm ảnh nhân sáu mươi tư điểm ảnh). Các CTU chứa cả các mẫu độ chói và sắc độ. Các cây mã hóa có thể được ứng dụng để chia các CTU thành các khối và sau đó chia nhỏ đệ quy các khối cho đến khi các cấu hình đạt được mà còn hỗ trợ việc mã hóa. Ví dụ, các thành phần độ chói của khung có thể được chia nhỏ cho đến khi các khối riêng chứa các trị số chiếu sáng tương đối đồng nhất. Hơn nữa, các thành phần sắc độ của khung có thể được chia nhỏ cho đến khi các khối riêng chứa các trị số màu sắc tương đối đồng nhất. Theo đó, các cơ chế phân chia thay đổi tùy thuộc vào nội dung của các khung video.

Ở bước 105, các cơ chế nén khác nhau được ứng dụng để nén các khối hình ảnh được phân chia ở bước 103. Ví dụ, việc dự báo liên kết và/hoặc việc dự báo bên trong có thể được ứng dụng. Việc dự báo liên kết 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 mô tả đối tượng trong khung tham chiếu không cần được mô tả lặp lại trong các khung liên kế. Cụ thể là, đối tượng, chẳng hạn như bảng, vẫn có thể ở vị trí không đổi qua nhiều khung. Do đó bảng được mô tả một lần và các khung liên kế có thể đề cập lại khung tham chiếu. Các cơ chế so khớp mẫu có thể được ứng dụng để so khớp các đối tượng qua nhiều khung. Hơn nữa, các đối tượng di chuyển có thể được biểu diễn qua nhiều khung, ví dụ do sự di chuyển đối tượng hoặc sự di chuyển camera. Theo ví dụ cụ thể, video có thể thể hiện ô tô mà di chuyển qua màn hình qua nhiều khung. Các vector chuyển động có thể được ứng dụng để mô tả sự di chuyển như vậy. Vector chuyển động là vector hai chiều mà cung cấp dịch vị từ các tọa độ của đối tượng trong khung tới các tọa độ của đối tượng trong khung tham chiếu. Như vậy, việc dự báo liên kết có thể mã hóa khối hình ảnh trong khung hiện thời là tập hợp của các vector chuyển động chỉ báo dịch vị từ khối tương ứng trong khung tham chiếu.

Việc dự báo bên trong mã hóa các khối trong khung chung. Việc dự báo bên trong tận dụng thực tế rằng các thành phần độ chói và sắc độ có xu hướng nhóm lại trong khung. Ví dụ, mảng màu xanh lá cây trong phần của cây có xu hướng được bố trí liên kế với các mảng màu xanh lá cây giống nhau. Việc dự báo bên trong ứng dụng nhiều chế độ dự báo định hướng (ví dụ, ba mươi ba trong HEVC), chế độ hai chiều, và chế độ dòng điện một chiều (direct current - DC). Các chế độ định hướng chỉ báo rằng khối hiện thời tương tự/giống với các mẫu của khối lân cận theo hướng tương ứng. Chế độ hai chiều chỉ báo rằng hàng loạt các khối cùng hàng/cột (ví dụ, mặt phẳng) có thể được nội suy dựa vào các khối lân cận ở các mép của hàng. Chế độ hai chiều, có hiệu lực, chỉ báo việc chuyển đổi tron tru của ánh sáng/màu sắc qua hàng/cột bằng cách ứng dụng độ dốc tương đối không đổi trong việc thay đổi các trị số. Chế độ DC được ứng dụng để làm mịn ranh giới và chỉ báo rằng khối tương tự/giống với trị số trung bình được kết hợp với các mẫu của tất cả các khối lân cận được kết hợp với các hướng góc của các chế độ dự báo định hướng. Theo

đó, các khối dự báo bên trong có thể biểu diễn các khối hình ảnh là các trị số chế độ dự báo quan hệ khác nhau thay vì các trị số thực tế. Hơn nữa, các khối dự báo liên kết có thể biểu diễn các khối hình ảnh là các trị số vectơ chuyển động thay vì các trị số thực tế. Trong cả hai trường hợp, các khối dự báo có thể không biểu diễn một cách chính xác các khối hình ảnh trong một số trường hợp. Các độ chênh lệch bất kỳ được lưu trữ trong các khối dự. Các phép biến đổi có thể được áp dụng tới các khối dự để nén thêm tệp.

Ở bước 107, các kỹ thuật lọc khác nhau có thể được áp dụng. Trong HEVC, các bộ lọc được áp dụng theo sơ đồ lọc trong vòng lặp. Việc dự báo dựa vào khối được thảo luận ở trên có thể dẫn đến việc tạo các hình ảnh khối ở bộ giải mã. Hơn nữa, sơ đồ dự báo dựa vào khối có thể mã hóa khối và sau đó tái cấu trúc khối được mã hóa cho việc sử dụng sau này như khối tham chiếu. Sơ đồ lọc trong vòng lặp áp dụng lặp lại các bộ lọc khử nhiễu, các bộ lọc giải khối, các bộ lọc vòng lặp thích nghi, và các bộ lọc dịch vị thích nghi mẫu (sample adaptive offset - SAO) tới các khối/các khung. Các bộ lọc này giảm bớt các rìa chặn như vậy sao cho tệp được mã hóa có thể được tái cấu trúc một cách chính xác. Hơn nữa, các bộ lọc này giảm bớt các rìa trong các khối tham chiếu được tái cấu trúc sao cho các rìa ít có khả năng tạo các rìa bổ sung ở các khối tiếp theo mà được mã hóa dựa vào các khối tham chiếu được tái cấu trúc.

Khi tín hiệu video đã được phân chia, được nén, và được lọc, dữ liệu kết quả được mã hóa ở dòng bit ở bước 109. Dòng bit bao gồm dữ liệu được thảo luận ở trên cũng như dữ liệu truyền tín hiệu bất kỳ được mong muốn để hỗ trợ việc tái cấu trúc tín hiệu video thích hợp ở bộ giải mã. Ví dụ, dữ liệu như vậy có thể bao gồm dữ liệu phân chia, dữ liệu dự báo, các khối dự, và các cờ khác nhau cung cấp các lệnh mã hóa tới bộ giải mã. Dòng bit có thể được lưu trữ trong bộ nhớ cho việc truyền về phía bộ giải mã khi yêu cầu. Dòng bit cũng có thể phát rộng và/hoặc truyền thông đa hướng về phía các bộ giải mã. Việc tạo dòng bit là quy trình lặp lại. Theo đó, các bước 101, 103, 105, 107, và 109 có thể xảy ra liên tục và/hoặc đồng thời qua nhiều khung và nhiều khối. Thứ tự được thể hiện trên Fig.1 được thể hiện nhằm rõ ràng và dễ dàng thảo luận, và không nhằm giới hạn quy trình mã hóa video ở thứ tự cụ thể.

Bộ giải mã thu dòng bit và bắt đầu quy trình giải mã ở bước 111. Cụ thể là, bộ giải mã ứng dụng sơ đồ giải mã entropi để chuyển đổi dòng bit thành cú pháp và dữ liệu video tương ứng. Bộ giải mã ứng dụng dữ liệu cú pháp từ dòng bit để xác định các phần chia cho các khung ở bước 111. Việc phân chia sẽ so khớp các kết quả của việc phân chia khối ở bước 103. Việc mã hóa/giải mã entropi như được ứng dụng ở bước 111 bây giờ được mô tả. Bộ mã hóa thực hiện nhiều sự lựa chọn trong suốt thời gian quy trình nén, chẳng hạn như lựa chọn các sơ đồ phân chia khối từ nhiều sự lựa chọn khả thi dựa vào việc phân chia theo không gian của các trị số trong (các) hình ảnh được đưa vào. Việc truyền tín hiệu các sự lựa chọn chính xác có thể ứng dụng số lượng lớn các ký tự nhị phân. Như được sử dụng ở đây, ký tự nhị phân là trị số nhị phân mà được coi là biến (ví dụ, trị số bit mà có thể thay đổi tùy thuộc vào ngữ cảnh). Việc mã entropi cho phép bộ mã hóa loại bỏ các tùy chọn bất kỳ mà rõ ràng không khả thi đối với trường hợp cụ thể, để lại tập hợp của các tùy chọn có thể cho phép được. Mỗi tùy chọn có thể cho phép được sau đó được gán từ mã. Độ dài của các từ mã dựa vào số lượng các tùy chọn có thể cho phép được (ví dụ, một ký tự nhị phân cho hai tùy chọn, hai ký tự nhị phân cho ba đến bốn tùy chọn, v.v.) Bộ mã hóa sau đó mã hóa từ mã cho tùy chọn được lựa chọn. Sơ đồ này làm giảm kích thước của các từ mã khi các từ mã lớn như được mong muốn để chỉ báo duy nhất sự lựa chọn từ tập hợp con nhỏ của các tùy chọn có thể cho phép được như trái ngược với chỉ báo duy nhất sự lựa chọn từ tập hợp lớn tiềm năng của tất cả các tùy chọn khả thi. Bộ giải mã sau đó giải mã sự lựa chọn bằng cách xác định tập hợp của các tùy chọn có thể cho phép được theo cách thức giống với bộ mã hóa. Bằng cách xác định tập hợp của các tùy chọn có thể cho phép được, bộ giải mã có thể đọc từ mã và xác định sự lựa chọn được thực hiện bởi bộ mã hóa.

Ở bước 113, bộ giải mã thực hiện việc giải mã khối. Cụ thể là, bộ giải mã ứng dụng các phép biến đổi ngược để tạo ra các khối dư. Sau đó bộ giải mã ứng dụng các khối dư và các khối dự báo tương ứng để tái cấu trúc các khối hình ảnh theo phân chia. Các khối dự báo có thể bao gồm cả các khối dự báo bên trong và các khối dự báo liên kết như được tạo ra ở bộ mã hóa ở bước 105. Các khối hình ảnh được tái cấu trúc sau đó được bố trí vào các khung của tín hiệu video được tái cấu trúc theo dữ liệu phân chia được xác định ở bước 111. Cú pháp cho bước 113

cũng có thể được truyền tín hiệu ở dòng bit qua việc mã entropi như được thảo luận ở trên.

Ở bước 115, việc lọc được thực hiện trên các khung của tín hiệu video được tái cấu trúc theo cách thức giống với bước 107 ở bộ mã hóa. Ví dụ, các bộ lọc khử nhiễu, các bộ lọc giải khối, các bộ lọc vòng lặp thích nghi, và các bộ lọc SAO có thể được áp dụng tới các khung để loại bỏ các rìa chặn. Khi các khung được lọc, tín hiệu video có thể được đưa ra tới màn hình ở bước 117 để xem bởi người dùng cuối.

Fig.2 là hình vẽ giản lược của hệ thống mã và giải mã (codec) 200 ví dụ để mã hóa video. Cụ thể là, hệ thống bộ mã hóa/giải mã 200 cung cấp tính năng để hỗ trợ việc thực hiện của phương pháp thao tác 100. Hệ thống bộ mã hóa/giải mã 200 được khái quát để mô tả các bộ phận được ứng dụng trong cả bộ mã hóa và bộ giải mã. Hệ thống bộ mã hóa/giải mã 200 thu và phân chia tín hiệu video như được thảo luận đối với các bước 101 và 103 theo phương pháp thao tác 100, mà dẫn đến tín hiệu video được phân chia 201. Hệ thống bộ mã hóa/giải mã 200 sau đó nén tín hiệu video được phân chia 201 thành dòng bit được mã hóa khi hoạt động như bộ mã hóa như được thảo luận đối với các bước 105, 107, và 109 theo phương pháp 100. Khi hoạt động như bộ giải mã hệ thống bộ mã hóa/giải mã 200 tạo ra tín hiệu video đầu ra từ dòng bit như được thảo luận đối với các bước 111, 113, 115, và 117 theo phương pháp thao tác 100. Hệ thống bộ mã hóa/giải mã 200 bao gồm bộ phận điều khiển bộ mã chung 211, bộ phận lượng tử hóa và chia tỷ lệ biến đổi 213, bộ phận đánh giá ảnh bên trong 215, bộ phận dự báo ảnh bên trong 217, bộ phận bù chuyển động 219, bộ phận đánh giá chuyển động 221, bộ phận biến đổi ngược và chia tỷ lệ 229, bộ phận phân tích điều khiển bộ lọc 227, bộ phận các bộ lọc trong vòng lặp 225, bộ phận bộ đệm ảnh được giải mã 223, và bộ phận mã hóa thuật toán nhị phân thích nghi ngữ cảnh và định dạng đoạn đầu (header formatting and context adaptive binary arithmetic coding - CABAC) 231. Các bộ phận như vậy được ghép nối như được thể hiện. Trên Fig.2, các đường đen chỉ báo sự di chuyển của dữ liệu được mã hóa/được giải mã trong khi các đường nét đứt chỉ báo sự di chuyển của dữ liệu điều khiển mà điều khiển thao tác của các bộ phận khác. Các bộ phận của hệ thống bộ mã hóa/giải mã 200 tất cả có thể có mặt ở bộ mã hóa. Bộ giải mã có

thể bao gồm tập hợp con của các bộ phận của hệ thống bộ mã hóa/giải mã 200. Ví dụ, bộ giải mã có thể bao gồm bộ phận dự báo ảnh bên trong 217, bộ phận bù chuyển động 219, bộ phận biến đổi ngược và chia tỷ lệ 229, bộ phận các bộ lọc trong vòng lặp 225, và bộ phận bộ đệm ảnh được giải mã 223. Các bộ phận này bây giờ được mô tả.

Tín hiệu video được phân chia 201 là chuỗi video được chụp mà đã được phân chia thành các khối của các điểm ảnh bởi cây mã hóa. Cây mã hóa ứng dụng các chế độ phân tách khác nhau để chia nhỏ khối của các điểm ảnh thành các khối nhỏ hơn của các điểm ảnh. Các khối này sau đó có thể được chia nhỏ thêm thành các khối nhỏ hơn. Các khối có thể được gọi là các nút trên cây mã hóa. Các nút bố mẹ lớn hơn được phân tách thành các nút con nhỏ hơn. Số lượng các lần nút được chia nhỏ được gọi là độ sâu của nút/cây mã hóa. Các khối được chia có thể được bao gồm trong các đơn vị mã hóa (CU) trong một số trường hợp. Ví dụ, CU có thể là phần con của CTU mà chứa khối độ chói, (các) khối (Cr) sắc độ chênh lệch màu đỏ, và (các) khối (Cb) sắc độ chênh lệch màu xanh lam cùng với các lệnh cú pháp tương ứng cho CU. Các chế độ phân tách có thể bao gồm cây nhị phân (binary tree - BT), cây tam phân (triple tree - TT), và cây tứ phân (quad tree - QT) được ứng dụng để lần lượt phân chia nút thành hai, ba, hoặc bốn nút con, có các hình dạng thay đổi tùy thuộc vào các chế độ phân tách được ứng dụng. Tín hiệu video được phân chia 201 được chuyển tiếp tới bộ phận điều khiển bộ mã chung 211, bộ phận lượng tử hóa và chia tỷ lệ biến đổi 213, bộ phận đánh giá ảnh bên trong 215, bộ phận phân tích điều khiển bộ lọc 227, và bộ phận đánh giá chuyển động 221 cho việc nén.

Bộ phận điều khiển bộ mã chung 211 được tạo cấu hình để đưa ra các quyết định liên quan đến việc mã của các hình ảnh của chuỗi video thành dòng bit theo các sự ràng buộc ứng dụng. Ví dụ, bộ phận điều khiển bộ mã chung 211 quản lý sự tối ưu hóa của tốc độ bit/kích thước dòng bit so với chất lượng tái cấu trúc. Các quyết định như vậy có thể được đưa ra dựa vào tính khả dụng không gian lưu trữ/độ rộng dải và các yêu cầu độ phân dải hình ảnh. Bộ phận điều khiển bộ mã chung 211 cũng quản lý việc sử dụng bộ đệm dựa vào tốc độ truyền để giảm bớt các vấn đề thiếu hoặc tràn bộ đệm. Để quản lý các vấn đề này, bộ phận điều khiển

bộ mã chung 211 quản lý việc phân chia, dự báo, và lọc bởi các bộ phận khác. Ví dụ, bộ phận điều khiển bộ mã chung 211 có thể tự động làm tăng độ phức tạp nén để làm tăng độ phân giải và làm tăng việc sử dụng độ rộng dải hoặc làm giảm độ phức tạp nén để làm giảm độ phân giải và việc sử dụng độ rộng dải. Do đó, bộ phận điều khiển bộ mã chung 211 điều khiển các bộ phận khác của hệ thống bộ mã hóa/giải mã 200 để cân bằng chất lượng tái cấu trúc tín hiệu video với các sự quan tâm tốc độ bit. Bộ phận điều khiển bộ mã chung 211 tạo dữ liệu điều khiển, mà điều khiển thao tác của các bộ phận khác. Dữ liệu điều khiển cũng được chuyển tiếp tới bộ phận CABAC và định dạng đoạn đầu 231 để được mã hóa ở dòng bit để truyền tín hiệu các thông số cho việc giải mã ở bộ giải mã.

Tín hiệu video được phân chia 201 cũng được gửi tới bộ phận đánh giá chuyển động 221 và bộ phận bù chuyển động 219 cho việc dự báo liên kết. Khung hoặc lát của tín hiệu video được phân chia 201 có thể được chia thành nhiều khối video. Bộ phận đánh giá chuyển động 221 và bộ phận bù chuyển động 219 thực hiện việc mã dự báo liên kết của khối video thu được liên quan đến một hoặc nhiều khối trong một hoặc nhiều khung tham chiếu để đưa ra việc dự báo theo thời gian. Hệ thống bộ mã hóa/giải mã 200 có thể thực hiện nhiều quy trình mã hóa, ví dụ, để lựa chọn chế độ mã hóa thích hợp cho mỗi khối của dữ liệu video.

Bộ phận đánh giá chuyển động 221 và bộ phận bù chuyển động 219 có thể được tích hợp cao, nhưng được minh họa một cách riêng biệt nhằm các mục đích khái niệm. Đánh giá chuyển động, được thực hiện bởi bộ phận đánh giá chuyển động 221, là quy trình tạo ra các vectơ chuyển động, mà đánh giá việc chuyển động cho các khối video. Vectơ chuyển động, ví dụ, có thể chỉ báo sự dịch chuyển của đối tượng được mã hóa liên quan đến khối dự báo. Khối dự báo là khối mà được tìm thấy trùng khớp chặt chẽ khối được mã hóa, về độ chênh lệch điểm ảnh. Khối dự báo cũng có thể được gọi là khối tham chiếu. Độ chênh lệch điểm ảnh như vậy có thể được xác định bởi tổng của độ chênh lệch tuyệt đối (sum of absolute difference - SAD), tổng của độ chênh lệch hình vuông (sum of square difference - SSD), hoặc các ma trận độ chênh lệch khác. HEVC ứng dụng nhiều đối tượng được mã hóa bao gồm CTU, các khối cây mã hóa (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 bao gồm

trong các CU. CU có thể được mã hóa là bộ phận dự báo (prediction unit - PU) chứa dữ liệu dự báo và/hoặc bộ phận biến đổi (transform unit - TU) chứa dữ liệu dư được biến đổi cho CU. Bộ phận đánh giá chuyển động 221 tạo ra các vector chuyển động, các PU, và các TU nhờ sử dụng phân tích tốc độ biến dạng là phần của quy trình tối ưu hóa tốc độ biến dạng. Ví dụ, bộ phận đánh giá chuyển động 221 có thể xác định nhiều khối tham chiếu, nhiều vector chuyển động, v.v. cho khối/khung hiện thời, và có thể lựa chọn các khối tham chiếu, các vector chuyển động, v.v. có các đặc tính tốc độ biến dạng tốt nhất. Các đặc tính tốc độ biến dạng tốt nhất cân bằng cả chất lượng của việc tái cấu trúc video (ví dụ, lượng mất mát dữ liệu bởi việc nén) với hiệu quả mã hóa (ví dụ, kích thước của việc mã hóa cuối cùng).

Theo một số ví dụ, hệ thống bộ mã hóa/giải mã 200 có thể tính toán các trị số cho các vị trí điểm ảnh nguyên con của các ảnh tham chiếu được lưu trữ trong bộ phận bộ đệm ảnh được giải mã 223. Ví dụ, hệ thống bộ mã hóa/giải mã video 200 có thể nội suy các trị số của một phần tư vị trí điểm ảnh, một phần tám vị trí điểm ảnh, hoặc các vị trí điểm ảnh phân số khác của ảnh tham chiếu. Do đó, bộ phận đánh giá chuyển động 221 có thể thực hiện việc tìm kiếm chuyển động liên quan đến các vị trí điểm ảnh đầy đủ và các vị trí điểm ảnh phân số và đưa ra vector chuyển động với độ chính xác điểm ảnh phân số. Bộ phận đánh giá chuyển động 221 tính toán vector chuyển động cho PU của khối video trong lát được mã hóa liên kết bằng cách so sánh vị trí của PU với vị trí của khối dự báo của ảnh tham chiếu. Bộ phận đánh giá chuyển động 221 đưa ra vector chuyển động được tính toán là dữ liệu chuyển động tới bộ phận CABAC và định dạng đoạn đầu 231 cho việc mã hóa và chuyển động tới bộ phận bù chuyển động 219.

Bù chuyển động, được thực hiện bởi bộ phận bù chuyển động 219, có thể bao gồm việc nạp hoặc tạo ra khối dự báo dựa vào vector chuyển động được xác định bởi bộ phận đánh giá chuyển động 221. Hơn nữa, bộ phận đánh giá chuyển động 221 và bộ phận bù chuyển động 219 có thể được tích hợp về mặt chức năng, theo một số ví dụ. Khi thu vector chuyển động cho PU của khối video hiện thời, bộ phận bù chuyển động 219 có thể bố trí khối dự báo trong đó vector chuyển động chỉ. Khối video dư sau đó được tạo nên bằng cách trừ các trị số điểm ảnh của khối dự

báo từ các trị số điểm ảnh của khối video hiện thời được mã hóa, tạo nên các trị số chênh lệch điểm ảnh. Nói chung, bộ phận đánh giá chuyển động 221 thực hiện việc đánh giá chuyển động liên quan đến các thành phần độ chói, và bộ phận bù chuyển động 219 sử dụng các vector chuyển động được tính toán dựa vào các thành phần độ chói cho cả các thành phần sắc độ và các thành phần độ chói. Khối dự báo và khối dư được chuyển tiếp tới bộ phận lượng tử hóa và chia tỷ lệ biến đổi 213.

Tín hiệu video được phân chia 201 cũng được gửi tới bộ phận đánh giá ảnh bên trong 215 và bộ phận dự báo ảnh bên trong 217. Như với bộ phận đánh giá chuyển động 221 và bộ phận bù chuyển động 219, bộ phận đánh giá ảnh bên trong 215 và bộ phận dự báo ảnh bên trong 217 có thể được tích hợp cao, nhưng được minh họa một cách riêng biệt nhằm các mục đích khái niệm. Bộ phận đánh giá ảnh bên trong 215 và bộ phận dự báo ảnh bên trong 217 dự báo bên trong khối hiện thời liên quan đến các khối trong khung hiện thời, như sự thay thế cho việc dự báo liên kết được thực hiện bởi bộ phận đánh giá chuyển động 221 và bộ phận bù chuyển động 219 giữa các khung, như được nêu trên. Cụ thể là, bộ phận đánh giá ảnh bên trong 215 xác định chế độ dự báo bên trong để sử dụng để mã hóa khối hiện thời. Theo một số ví dụ, bộ phận đánh giá ảnh bên trong 215 lựa chọn chế độ dự báo bên trong thích hợp để mã hóa khối hiện thời từ nhiều chế độ dự báo bên trong được thử nghiệm. Các chế độ dự báo bên trong được lựa chọn sau đó được chuyển tiếp tới bộ phận CABAC và định dạng đoạn đầu 231 để mã hóa.

Ví dụ, bộ phận đánh giá ảnh bên trong 215 tính toán các trị số tốc độ biến dạng nhờ sử dụng phân tích tốc độ biến dạng cho các chế độ dự báo bên trong được thử nghiệm khác nhau, và lựa chọn chế độ dự báo bên trong có các đặc tính tốc độ biến dạng tốt nhất trong số các chế độ được thử nghiệm. Phân tích tốc độ biến dạng nói chung xác định lượng biến dạng (hoặc lỗi) giữa khối được mã hóa và khối không được mã hóa gốc mà đã được mã hóa để tạo ra khối được mã hóa, cũng như tốc độ bit (ví dụ, một số bit) được sử dụng để tạo ra khối được mã hóa. Bộ phận đánh giá ảnh bên trong 215 tính toán các tỷ lệ từ các độ biến dạng và các tốc độ cho các khối được mã hóa khác nhau để xác định chế độ dự báo bên trong nào thể hiện trị số tốc độ biến dạng tốt nhất cho khối. Ngoài ra, bộ phận đánh giá ảnh bên trong 215 có thể được tạo cấu hình để mã hóa các khối độ sâu của bản đồ độ sâu nhờ sử

dụng chế độ mô hình hóa độ sâu (depth modeling mode - DMM) dựa vào sự tối ưu hóa tốc độ biến dạng (RDO).

Bộ phận dự báo ảnh bên trong 217 có thể tạo ra khối dư từ khối dự báo dựa vào các chế độ dự báo bên trong được lựa chọn được xác định bởi bộ phận đánh giá ảnh bên trong 215 khi được thực hiện trên bộ mã hóa hoặc đọc khối dư từ dòng bit khi được thực hiện trên bộ giải mã. Khối dư bao gồm độ chênh lệch về các trị số giữa khối dự báo và khối gốc, được biểu diễn như ma trận. Khối dư sau đó được chuyển tiếp tới bộ phận lượng tử hóa và chia tỷ lệ biến đổi 213. Bộ phận đánh giá ảnh bên trong 215 và bộ phận dự báo ảnh bên trong 217 có thể thao tác trên cả các thành phần độ chói và sắc độ.

Bộ phận lượng tử hóa và chia tỷ lệ biến đổi 213 được tạo cấu hình để nén thêm khối dư. Bộ phận lượng tử hóa và chia tỷ lệ biến đổi 213 áp dụng phép biến đổi, chẳng hạn như 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, tới khối dư, tạo ra khối video bao gồm các trị số hệ số biến đổi dư. Các phép biến đổi wavelet, các phép biến đổi nguyên, các phép biến đổi dải tần con hoặc các loại khác của các phép biến đổi cũng có thể được sử dụng. Phép biến đổi có thể chuyển đổi thông tin dư từ miền trị số điểm ảnh đến miền biến đổi, chẳng hạn như miền tần số. Bộ phận lượng tử hóa và chia tỷ lệ biến đổi 213 cũng được tạo cấu hình để chia tỷ lệ thông tin dư được biến đổi, ví dụ dựa vào tần số. Việc chia tỷ lệ như vậy bao gồm việc áp dụng hệ số chia tỷ lệ tới thông tin dư sao cho thông tin tần số khác nhau được lượng tử hóa ở các độ chi tiết khác nhau, mà có thể ảnh hưởng chất lượng trực quan cuối cùng của video được tái cấu trúc. Bộ phận lượng tử hóa và chia tỷ lệ biến đổi 213 cũng được tạo cấu hình để lượng tử hóa các hệ số biến đổi để làm giảm thêm tốc độ bit. Quy trình lượng tử hóa có thể làm giảm độ sâu bit được kết hợp với một số hoặc tất cả các hệ số. Mức lượng tử hóa có thể được cải biến bằng cách điều chỉnh thông số lượng tử hóa. Theo một số ví dụ, bộ phận lượng tử hóa và chia tỷ lệ biến đổi 213 sau đó có thể thực hiện việc quét ma trận bao gồm các hệ số biến đổi được lượng tử hóa. Các hệ số biến đổi được lượng tử hóa được chuyển tiếp tới bộ phận CABAC và định dạng đoạn đầu 231 để được mã hóa ở dòng bit.

Bộ phận biến đổi ngược và chia tỷ lệ 229 áp dụng thao tác ngược của bộ phận lượng tử hóa và chia tỷ lệ biến đổi 213 để hỗ trợ việc đánh giá chuyển động. Bộ phận biến đổi ngược và chia tỷ lệ 229 áp dụng việc chia tỷ lệ, phép biến đổi, và/hoặc lượng tử hóa ngược để tái cấu trúc khối dư trong miền điểm ảnh, ví dụ, cho việc sử dụng sau này như khối tham chiếu mà có thể trở thành khối dự báo cho khối hiện thời khác. Bộ phận đánh giá chuyển động 221 và/hoặc bộ phận bù chuyển động 219 có thể tính toán khối tham chiếu bằng cách bổ sung lại khối dư vào khối dự báo tương ứng cho việc sử dụng trong việc đánh giá chuyển động của khối/khung sau đó. Các bộ lọc được áp dụng tới các khối tham chiếu được tái cấu trúc để giảm bớt các rìa được tạo trong suốt thời gian chia tỷ lệ, lượng tử hóa, và biến đổi. Các rìa như vậy theo cách khác có thể gây ra việc dự báo không chính xác (và tạo các rìa bổ sung) khi các khối tiếp theo được dự báo.

Bộ phận phân tích điều khiển bộ lọc 227 và bộ phận các bộ lọc trong vòng lặp 225 áp dụng các bộ lọc tới các khối dư và/hoặc tới các khối hình ảnh được tái cấu trúc. Ví dụ, khối dư được biến đổi từ bộ phận biến đổi ngược và chia tỷ lệ 229 có thể được kết hợp với khối dự báo tương ứng từ bộ phận dự báo ảnh bên trong 217 và/hoặc bộ phận bù chuyển động 219 để tái cấu trúc khối hình ảnh gốc. Các bộ lọc sau đó có thể được áp dụng tới khối hình ảnh được tái cấu trúc. Theo một số ví dụ, các bộ lọc thay vì vậy có thể được áp dụng tới các khối dư. Như với các bộ phận khác trên Fig.2, bộ phận phân tích điều khiển bộ lọc 227 và bộ phận các bộ lọc trong vòng lặp 225 được tích hợp cao và có thể được thực hiện cùng nhau, nhưng được mô tả một cách riêng biệt nhằm các mục đích khái niệm. Các bộ lọc được áp dụng tới các khối tham chiếu được tái cấu trúc được áp dụng tới các vùng không gian cụ thể và bao gồm nhiều thông số để điều chỉnh cách thức các bộ lọc như vậy được áp dụng. Bộ phận phân tích điều khiển bộ lọc 227 phân tích các khối tham chiếu được tái cấu trúc để xác định nơi các bộ lọc như vậy sẽ được áp dụng và thiết đặt các thông số tương ứng. Dữ liệu như vậy được chuyển tiếp tới bộ phận CABAC và định dạng đoạn đầu 231 như dữ liệu điều khiển bộ lọc để mã hóa. Bộ phận các bộ lọc trong vòng lặp 225 áp dụng các bộ lọc như vậy dựa vào dữ liệu điều khiển bộ lọc. Các bộ lọc có thể bao gồm bộ lọc giải khối, bộ lọc khử nhiễu, bộ lọc SAO, và bộ lọc vòng lặp thích nghi. Các bộ lọc như vậy có thể được áp dụng trong miền

không gian/điểm ảnh (ví dụ, trên khối điểm ảnh được tái cấu trúc) hoặc trong miền tần số, tùy thuộc vào ví dụ.

Khi hoạt động như bộ mã hóa, khối hình ảnh được tái cấu trúc được lọc, khối dư, và/hoặc khối dự báo được lưu trữ trong bộ phận bộ đệm ảnh được giải mã 223 nhằm sử dụng sau này trong việc đánh giá chuyển động như được thảo luận ở trên. Khi hoạt động như bộ giải mã, bộ phận bộ đệm ảnh được giải mã 223 lưu trữ và chuyển tiếp các khối được tái cấu trúc và được lọc tới màn hình như phần của tín hiệu video đầu ra. Bộ phận bộ đệm ảnh được giải mã 223 có thể là thiết bị nhớ bất kỳ có khả năng lưu trữ các khối dự báo, các khối dư, và/hoặc các khối hình ảnh được tái cấu trúc.

Bộ phận CABAC và định dạng đoạn đầu 231 thu dữ liệu từ các bộ phận khác nhau của hệ thống bộ mã hóa/giải mã 200 và mã hóa dữ liệu như vậy thành dòng bit được mã hóa cho việc truyền về phía bộ giải mã. Cụ thể là, bộ phận CABAC và định dạng đoạn đầu 231 tạo ra các đoạn đầu khác nhau để mã hóa dữ liệu điều khiển, chẳng hạn như dữ liệu điều khiển chung và dữ liệu điều khiển bộ lọc. Hơn nữa, dữ liệu dự báo, bao gồm dữ liệu chuyển động và dự báo bên trong, cũng như dữ liệu dư dưới dạng của dữ liệu hệ số biến đổi được lượng tử hóa tất cả được mã hóa ở dòng bit. Dòng bit cuối cùng bao gồm tất cả thông tin được mong muốn bởi bộ giải mã để tái cấu trúc tín hiệu video được phân chia 201 gốc. Thông tin như vậy cũng có thể bao gồm các bảng chỉ số chế độ dự báo bên trong (cũng được gọi là các bảng ánh xạ từ mã), các định nghĩa về các ngữ cảnh mã hóa cho các khối khác nhau, các sự chỉ báo của các chế độ dự báo bên trong có thể xảy ra nhất, sự chỉ báo của thông tin phân chia, v.v. Dữ liệu như vậy có thể được mã hóa bằng cách ứng dụng việc mã entropi. Ví dụ, thông tin có thể được mã hóa bằng cách ứng dụng việc mã độ dài biến đổi thích nghi ngữ cảnh (context adaptive variable length coding - CAVLC), CABAC, mã hóa thuật toán nhị phân thích nghi ngữ cảnh dựa vào cú pháp (syntax-based context-adaptive binary arithmetic coding - SBAC), mã hóa entropi phân chia khoảng xác suất (probability interval partition entropy - PIPE), hoặc kỹ thuật mã hóa entropi khác. Tiếp theo việc mã entropi, dòng bit được mã hóa có thể được truyền tới thiết bị khác (ví dụ, bộ giải mã video) hoặc được lưu trữ cho việc truyền hoặc truy xuất sau này.

Fig.3 là sơ đồ khối minh họa bộ mã hóa video 300 ví dụ. Bộ mã hóa video 300 có thể được ứng dụng để thực hiện các chức năng mã hóa của hệ thống bộ mã hóa/giải mã 200 và/hoặc thực hiện các bước 101, 103, 105, 107, và/hoặc 109 của phương pháp thao tác 100. Bộ mã hóa 300 phân chia tín hiệu video đầu vào, dẫn đến tín hiệu video được phân chia 301, mà về cơ bản giống với tín hiệu video được phân chia 201. Tín hiệu video được phân chia 301 sau đó được nén và được mã hóa thành dòng bit bởi các bộ phận của bộ mã hóa 300.

Cụ thể là, tín hiệu video được phân chia 301 được chuyển tiếp tới bộ phận dự báo ảnh bên trong 317 cho việc dự báo bên trong. Bộ phận dự báo ảnh bên trong 317 về cơ bản có thể giống với bộ phận đánh giá ảnh bên trong 215 và bộ phận dự báo ảnh bên trong 217. Tín hiệu video được phân chia 301 cũng được chuyển tiếp tới bộ phận bù chuyển động 321 cho việc dự báo liên kết dựa vào các khối tham chiếu trong bộ phận bộ đệm ảnh được giải mã 323. Bộ phận bù chuyển động 321 về cơ bản có thể giống với bộ phận đánh giá chuyển động 221 và bộ phận bù chuyển động 219. Các khối dự báo và các khối dư từ bộ phận dự báo ảnh bên trong 317 và bộ phận bù chuyển động 321 được chuyển tiếp tới bộ phận biến đổi và lượng tử hóa 313 cho việc biến đổi và lượng tử hóa của các khối dư. Bộ phận biến đổi và lượng tử hóa 313 về cơ bản có thể giống với bộ phận lượng tử hóa và chia tỷ lệ biến đổi 213. Các khối dư và các khối dự báo được biến đổi và được lượng tử hóa tương ứng (cùng với dữ liệu điều khiển được kết hợp) được chuyển tiếp tới bộ phận mã hóa entropi 331 để mã hóa thành dòng bit. Bộ phận mã hóa entropi 331 về cơ bản có thể giống với bộ phận CABAC và định dạng đoạn đầu 231.

Các khối dư được lượng tử hóa và được biến đổi và/hoặc các khối dự báo tương ứng cũng được chuyển tiếp từ bộ phận biến đổi và lượng tử hóa 313 tới bộ phận biến đổi và lượng tử hóa ngược 329 cho việc tái cấu trúc thành các khối tham chiếu cho việc sử dụng bởi bộ phận bù chuyển động 321. Bộ phận biến đổi và lượng tử hóa ngược 329 về cơ bản có thể giống với bộ phận biến đổi ngược và chia tỷ lệ 229. Các bộ lọc trong vòng lặp trong bộ phận các bộ lọc trong vòng lặp 325 cũng được áp dụng tới các khối dư và/hoặc các khối tham chiếu được tái cấu trúc, tùy thuộc vào ví dụ. Bộ phận các bộ lọc trong vòng lặp 325 về cơ bản có thể giống với bộ phận phân tích điều khiển bộ lọc 227 và bộ phận các bộ lọc trong vòng

lập 225. Bộ phận các bộ lọc trong vòng lặp 325 có thể bao gồm nhiều bộ lọc như được thảo luận đối với bộ phận các bộ lọc trong vòng lặp 225. Các khối được lọc sau đó được lưu trữ trong bộ phận bộ đệm ảnh được giải mã 323 cho việc sử dụng như các khối tham chiếu bởi bộ phận bù chuyển động 321. Bộ phận bộ đệm ảnh được giải mã 323 về cơ bản có thể giống với bộ phận bộ đệm ảnh được giải mã 223.

Fig.4 là sơ đồ khối minh họa bộ giải mã video 400 ví dụ. Bộ giải mã video 400 có thể được ứng dụng để thực hiện các chức năng giải mã của hệ thống bộ mã hóa/giải mã 200 và/hoặc thực hiện các bước 111, 113, 115, và/hoặc 117 của phương pháp thao tác 100. Bộ giải mã 400 thu dòng bit, ví dụ từ bộ mã hóa 300, và tạo ra tín hiệu video đầu ra được tái cấu trúc dựa vào dòng bit cho việc hiển thị tới người dùng cuối.

Dòng bit được thu bởi bộ phận giải mã entropi 433. Bộ phận giải mã entropi 433 được tạo cấu hình để thực hiện sơ đồ giải mã entropi, chẳng hạn như mã hóa CAVLC, CABAC, SBAC, PIPE, hoặc các kỹ thuật mã hóa entropi khác. Ví dụ, bộ phận giải mã entropi 433 có thể ứng dụng thông tin đoạn đầu để cung cấp ngữ cảnh để diễn dịch dữ liệu bổ sung được mã hóa như các từ mã ở dòng bit. Thông tin được giải mã bao gồm thông tin được mong muốn bất kỳ để giải mã tín hiệu video, chẳng hạn như dữ liệu điều khiển chung, dữ liệu điều khiển bộ lọc, thông tin phân chia, dữ liệu chuyển động, dữ liệu dự báo, và các hệ số biến đổi được lượng tử hóa từ các khối dư. Các hệ số biến đổi được lượng tử hóa được chuyển tiếp tới bộ phận biến đổi và lượng tử hóa ngược 429 cho việc tái cấu trúc thành các khối dư. Bộ phận biến đổi và lượng tử hóa ngược 429 có thể giống với bộ phận biến đổi và lượng tử hóa ngược 329.

Các khối dư và/hoặc các khối dự báo được tái cấu trúc được chuyển tiếp tới bộ phận dự báo ảnh bên trong 417 cho việc tái cấu trúc thành các khối hình ảnh dựa vào các thao tác dự báo bên trong. Bộ phận dự báo ảnh bên trong 417 có thể giống với bộ phận đánh giá ảnh bên trong 215 và bộ phận dự báo ảnh bên trong 217. Cụ thể là, bộ phận dự báo ảnh bên trong 417 ứng dụng các chế độ dự báo để bố trí khối tham chiếu trong khung và áp dụng khối dư tới kết quả để tái cấu trúc

các khối hình ảnh được dự báo bên trong. Các khối hình ảnh được dự báo bên trong được tái cấu trúc và/hoặc các khối dư và dữ liệu dự báo liên kết tương ứng được chuyển tiếp tới bộ phận bộ đệm ảnh được giải mã 423 qua bộ phận các bộ lọc trong vòng lặp 425, mà về cơ bản có thể lần lượt giống với bộ phận bộ đệm ảnh được giải mã 223 và bộ phận các bộ lọc trong vòng lặp 225. Bộ phận các bộ lọc trong vòng lặp 425 lọc các khối hình ảnh được tái cấu trúc, các khối dư và/hoặc các khối dự báo, và thông tin như vậy được lưu trữ trong bộ phận bộ đệm ảnh được giải mã 423. Các khối hình ảnh được tái cấu trúc từ bộ phận bộ đệm ảnh được giải mã 423 được chuyển tiếp tới bộ phận bù chuyển động 421 cho việc dự báo liên kết. Bộ phận bù chuyển động 421 về cơ bản có thể giống với bộ phận đánh giá chuyển động 221 và/hoặc bộ phận bù chuyển động 219. Cụ thể là, bộ phận bù chuyển động 421 ứng dụng các vectơ chuyển động từ khối tham chiếu để tạo ra khối dự báo và áp dụng khối dư tới kết quả để tái cấu trúc khối hình ảnh. Các khối được tái cấu trúc kết quả cũng có thể được chuyển tiếp qua bộ phận các bộ lọc trong vòng lặp 425 tới bộ phận bộ đệm ảnh được giải mã 423. Bộ phận bộ đệm ảnh được giải mã 423 tiếp tục lưu trữ các khối hình ảnh được tái cấu trúc bổ sung, mà có thể được tái cấu trúc thành các khung qua thông tin phân chia. Các khung như vậy cũng có thể được đặt trong chuỗi. Chuỗi được đưa ra tới màn hình là tín hiệu video đầu ra được tái cấu trúc.

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

Dòng bit 500 bao gồm tập hợp thông số chuỗi (SPS) 510, các tập hợp thông số hình ảnh (PPS) 512, các đoạn đầu lát 514, dữ liệu hình ảnh 520, và một hoặc nhiều bản tin SEI 515. SPS 510 chứa dữ liệu chuỗi phổ biến tới tất cả các ảnh trong chuỗi video được chứa ở dòng bit 500. Dữ liệu như vậy có thể bao gồm việc định kích thước ảnh, độ sâu bit, các thông số công cụ mã hóa, các giới hạn tốc độ bit, v.v. PPS 512 chứa các thông số mà cụ thể với một hoặc nhiều ảnh tương ứng.

Do đó, mỗi ảnh trong chuỗi video có thể đề cập đến một PPS 512. PPS 512 có thể chỉ báo các công cụ mã hóa khả dụng đối với các ô trong các ảnh, các thông số lượng tử hóa, các dịch vị, các thông số công cụ mã hóa ảnh cụ thể tương ứng (ví dụ, các điều khiển bộ lọc), v.v. Đoạn đầu lát 514 chứa các thông số mà cụ thể với một hoặc nhiều lát 524 tương ứng trong ảnh. Do đó, mỗi lát 524 trong chuỗi video có thể đề cập đến đoạn đầu lát 514. Đoạn đầu lát 514 có thể chứa thông tin loại lát, các số đếm thứ tự ảnh (POC), các danh mục ảnh tham chiếu, các trọng số dự báo, các điểm mục nhập ô, các thông số giải khối, v.v. Theo một số ví dụ, các lát 524 có thể được gọi là các nhóm ô. Trong trường hợp như vậy, đoạn đầu lát 514 có thể được gọi là đoạn đầu nhóm ô. Các bản tin SEI 515 là các bản tin tùy chọn mà chứa siêu dữ liệu mà không được yêu cầu cho việc giải mã khối, nhưng có thể được ứng dụng nhằm các mục đích liên quan chẳng hạn như chỉ báo thời điểm đầu ra ảnh, các sự thiết đặt màn hình, phát hiện mất mát, che dấu mất mát, v.v.

Dữ liệu hình ảnh 520 chứa dữ liệu video được mã hóa theo việc dự báo liên kết và/hoặc việc dự báo bên trong cũng như dữ liệu dư được biến đổi và được lượng tử hóa tương ứng. Dữ liệu hình ảnh 520 như vậy được phân loại theo việc phân chia được sử dụng để phân chia hình ảnh trước khi mã hóa. Ví dụ, chuỗi video được chia thành các ảnh 521. Các ảnh 521 còn có thể được chia thành các ảnh con 522, mà được chia thành các lát 524. Các lát 524 còn có thể được chia thành các ô và/hoặc các CTU. Các CTU còn được chia thành các khối mã hóa dựa vào các cây mã hóa. Các khối mã hóa sau đó có thể được mã hóa/được giải mã theo các cơ chế dự báo. 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 đề cập đến PPS 512 và các lát 524 đề cập đến đoạn đầu lát 514. Các ảnh con 522 có thể được phân chia một cách thích hợp qua toàn bộ chuỗi video (cũng được biết đến là phân đoạn), và do đó có thể đề cập đến SPS 510. Mỗi lát 524 có thể chứa một hoặc nhiều ô. Mỗi lát 524, và do đó ảnh 521 và ảnh con 522, cũng có thể chứa các CTU.

Mỗi ảnh 521 có thể chứa toàn bộ tập hợp của dữ liệu trực quan được kết hợp với chuỗi video cho thời điểm tương ứng đúng lúc. Tuy nhiên, các ứng dụng nhất định có thể muốn hiển thị chỉ 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

người dùng của ảnh 521, mà tạo cảm giác có mặt trong khung cảnh được mô tả trong ảnh 521. Vùng người dùng có thể muốn xem không được biết đến khi dòng bit 500 được mã hóa. Theo đó, ảnh 521 có thể chứa mỗi vùng khả thi người dùng có thể xem về mặt tiềm năng là 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 qua ảnh 521 của chuỗi video không liên quan. Theo ví dụ khác nữa, các hệ thống hội nghị từ xa có thể hiển thị toàn bộ ảnh 521 của người dùng mà hiện tại đang nói và ảnh con 522 của người dùng mà hiện tại không đang nói. Theo đó, ảnh con 522 có thể chứa vùng được định rõ của ảnh 521. Ảnh con 522 mà bị ràng buộc chuyển động về mặt thời gian có thể giải mã được một cách riêng biệt với phần còn lại của ảnh 521. Cụ thể là, ảnh con bị ràng buộc chuyển động theo thời gian được mã hóa mà không tham chiếu tới các mẫu bên ngoài của ảnh con bị ràng buộc chuyển động theo thời gian, và do đó chứa đủ thông tin cho việc giải mã hoàn toàn mà không tham chiếu tới phần còn lại của ảnh 521.

Mỗi lát 524 có thể là hình chữ nhật được định rõ bởi CTU ở góc trên bên trái và CTU ở góc dưới cùng bên phải. Theo một số ví dụ, lát 524 bao gồm hàng loạt các ô và/hoặc các CTU theo thứ tự quét mảnh tiến hành từ trái sang phải và từ trên xuống dưới. Theo các ví dụ khác, lát 524 là lát hình chữ nhật. Lát hình chữ nhật có thể không đi qua toàn bộ độ rộng của ảnh theo thứ tự quét mảnh. Thay vì vậy, lát hình chữ nhật có thể chứa vùng hình chữ nhật và/vùng hình vuông của ảnh 521 và/hoặc ảnh con 522 được định rõ về CTU và/hoặc các hàng ô và CTU và/hoặc các cột ô. Lát 524 là đơn vị nhỏ nhất mà có thể được hiển thị một cách riêng biệt bởi bộ giải mã. Do đó, các lát 524 từ ảnh 521 có thể được gán tới các ảnh con 522 khác nhau để mô tả một cách riêng biệt các vùng được 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 người dùng được lựa chọn hoặc nhóm con được định rõ trước của các ảnh con 522. Ví dụ, ảnh 521 có thể được chia thành chín ảnh con 522, nhưng bộ giải mã chỉ có thể hiển thị ảnh con 523 đơn từ nhóm của các ảnh con 522. Các ảnh con 523 chứa các lát 525, mà là nhóm con được lựa chọn hoặc được định rõ

trước của các lát 524. Để cho phép hiển thị tách biệt các ảnh con 523, dòng bit con 501 có thể được trích 529 từ dòng bit 500. Việc trích 529 có thể xảy ra ở phía bộ mã hóa sao cho bộ giải mã chỉ thu dòng bit con 501. Trong các trường hợp khác, toàn bộ dòng bit 500 được truyền tới bộ giải mã và bộ giải mã trích 529 dòng bit con 501 cho việc giải mã tách biệt. Cần lưu ý rằng dòng bit con 501 cũng có thể được gọi chung là dòng bit trong một số trường hợp. Dòng bit con 501 bao gồm SPS 510, PPS 512, các ảnh con 523 được lựa chọn, cũng như các đoạn đầu lát 514, và các bản tin SEI 515 mà thích hợp với các ảnh con 523 và/hoặc các lát 525.

Sáng chế truyền tín hiệu dữ liệu khác nhau để hỗ trợ việc mã hữu hiệu của các ảnh con 522 cho sự 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 liên quan đến tập hợp hoàn thiện của các ảnh con 522. Kích thước ảnh con 531 bao gồm độ cao ảnh con trong các mẫu độ chói và độ rộng ảnh con trong các mẫu độ chói cho ảnh con 522 tương ứng. Vị trí ảnh con 532 bao gồm khoảng cách dịch vị giữa mẫu trên cùng bên trái của ảnh con 522 tương ứng và mẫu trên cùng bên trái của ảnh 521. Vị trí ảnh con 532 và kích thước ảnh con 531 định rõ sự phối trí 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 màn hình ảnh con 522 hoặc trị số được định rõ 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 mã video, dữ liệu liên quan đến 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 các ảnh con 522 có thể được sử dụng bởi các ứng dụng, chẳng hạn như các ứng dụng dựa vào ROI, các ứng dụng VR, v.v., mà tùy thuộc vào các phần chia ảnh con 522 thích hợp qua chuỗi/phân đoạn video. Như vậy, các phần chia ảnh con 522 nói chung không thay đổi trên cơ sở mỗi ảnh. Việc đặt thông tin phối trí cho các ảnh con 522 trong SPS 510 đảm bảo rằng sự phối trí chỉ được truyền tín hiệu một lần cho chuỗi/phân đoạn khác ngoài được truyền tín hiệu dư cho mỗi PPS 512 (mà có thể được truyền tín hiệu cho mỗi ảnh 521 trong một số trường hợp). Ngoài ra, việc truyền tín hiệu thông tin ảnh con 522, thay vì phụ thuộc vào bộ giải mã để dẫn ra thông tin như vậy, làm giảm khả năng lỗi trong trường hợp của các gói bị mất và

hỗ trợ tính năng bổ sung về việc trích các ảnh con 523. Theo đó, việc truyền tín hiệu sự phối trí ảnh con 522 trong SPS 510 nâng cao tính năng của bộ mã hóa và/hoặc bộ giải mã.

SPS 510 cũng chứa các cờ các ảnh con bị ràng buộc chuyển động 534 liên quan đến tập hợp hoàn thiện của các ảnh con 522. Các cờ các ảnh con bị ràng buộc chuyển động 534 chỉ báo xem mỗi ảnh con 522 là ảnh con bị ràng buộc chuyển động theo thời gian hay không. Do đó, bộ giải mã có thể đọc các cờ các ảnh con bị ràng buộc chuyển động 534 và xác định các ảnh con 522 nào có thể được trích một cách riêng biệt và được hiển thị mà không giải mã các ảnh con 522 khác. Điều này cho phép các ảnh con 522 được lựa chọn được mã hóa là các ảnh con bị ràng buộc chuyển động theo thời gian trong khi cho phép các ảnh con 522 khác được mã hóa mà không có sự giới hạn như vậy đối với hiệu quả mã hóa được tăng.

Các ID ảnh con 533 cũng được bao gồm trong các đoạn đầu lát 514. Mỗi đoạn đầu lát 514 chứa dữ liệu thích hợp với tập hợp của các lát 524 tương ứng. Theo đó, đoạn đầu lát 514 chứa chỉ các ID ảnh con 533 tương ứng với các lát 524 được kết hợp với đoạn đầu lát 514. Như vậy, bộ giải mã có thể thu lát 524, thu nhận ID ảnh con 533 từ đoạn 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 đoạn đầu lát 514 để tương quan với dữ liệu liên quan trong SPS 510. Như vậy, bộ giải mã có thể xác định cách thức để bố trí các ảnh con 522/523 và các lát 524/525 bằng cách đọc SPS 510 và các đoạn đầu lát 514 thích hợp. Điều này cho phép các ảnh con 523 và các lát 525 được giải mã thậm chí nếu một số ảnh con 522 bị mất trong khi truyền hoặc cố ý được bỏ qua để làm tăng hiệu quả mã hóa.

Bản tin SEI 515 cũng có thể chứa mức ảnh con 535. Mức ảnh con 535 chỉ báo các tài nguyên phần cứng được cần để giải mã ảnh con 522 tương ứng. Theo cách này, mỗi ảnh con 522 có thể được mã hóa một cách độc lập với các ảnh con 522 khác. Điều này đảm bảo mỗi ảnh con 522 có thể được cấp phát lượng chính xác của các tài nguyên phần cứng ở bộ giải mã. Không có mức ảnh con 535 như vậy, mỗi ảnh con 522 sẽ được cấp phát với đủ các tài nguyên để giải mã ảnh con 522 phức tạp nhất. Do đó, mức ảnh con 535 ngăn ngừa bộ giải mã khỏi cấp phát

quá các tài nguyên phần cứng nếu các ảnh con 522 được kết hợp với việc thay đổi các yêu cầu tài nguyên phần cứng thay đổi.

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

Ảnh 600 về cơ bản có thể giống với ảnh 521. Hơn nữa, ảnh 600 có thể được phân chia thành các ảnh con 622, mà về cơ bản giống với các ảnh con 522. Các ảnh con 622 đều bao gồm kích thước ảnh con 631, mà có thể được bao gồm ở dòng bit 500 như kích thước ảnh con 531. Kích thước ảnh con 631 bao gồm độ rộng ảnh con 631a và độ cao ảnh con 631b. Độ rộng ảnh con 631a là độ rộng của ảnh con 622 tương ứng trong các bộ phận của các mẫu độ chói. Độ cao ảnh con 631b là độ cao của ảnh con 622 tương ứng trong các bộ phận của các mẫu độ chói. Các ảnh con 622 đều bao gồm ID ảnh con 633, mà có thể được bao gồm ở dòng bit 500 như ID ảnh con 633. ID ảnh con 633 có thể là trị số bất kỳ mà nhận dạng duy nhất mỗi ảnh con 622. Theo ví dụ được thể hiện, ID ảnh con 633 là chỉ số ảnh con 622. Các ảnh con 622 đều bao gồm vị trí 632, mà có thể được bao gồm ở dòng bit 500 như vị trí ảnh con 532. Vị trí 632 được thể hiện là dịch vị giữa mẫu trên cùng bên trái của ảnh con 622 tương ứng và mẫu trên cùng bên trái 642 của ảnh 600.

Cũng như được thể hiện, một số ảnh con 622 có thể là các ảnh con bị ràng buộc chuyển động theo thời gian 634 và các ảnh con 622 khác có thể không. Theo ví dụ được thể hiện, ảnh con 622 với ID ảnh con 633 là năm là ảnh con bị ràng buộc chuyển động theo thời gian 634. Điều này chỉ báo rằng ảnh con 622 được nhận dạng là năm được mã hóa mà không tham chiếu tới ảnh con 622 khác bất kỳ và do đó có thể được trích và được giải mã một cách riêng biệt mà không xem xét dữ liệu từ các ảnh con 622 khác. Sự chỉ báo về việc các ảnh con 622 nào là các ảnh con bị ràng buộc chuyển động theo thời gian 634 có thể được truyền tín hiệu ở dòng bit 500 trong các cờ các ảnh con bị ràng buộc chuyển động 534.

Như được thể hiện, các ảnh con 622 có thể bị ràng buộc để bao trùm ảnh 600 mà không có khe hở hoặc chồng lấn. Khe hở là vùng của ảnh 600 mà không được bao gồm trong ảnh con 622 bất kỳ. Chồng lấn là vùng của ảnh 600 mà được bao gồm trong nhiều hơn một ảnh con 622. Theo ví dụ được thể hiện trên Fig.6, các ảnh con 622 được phân chia từ ảnh 600 để ngăn ngừa cả các khe hở và các sự chồng lấn. Các khe hở khiến các mẫu ảnh 600 được bỏ ngoài các ảnh con 622. Các sự chồng lấn khiến các lát được kết hợp được bao gồm trong nhiều ảnh con 622. Do đó, các khe hở và các sự chồng lấn có thể khiến các mẫu bị ảnh hưởng bởi sự xử lý khác biệt khi các ảnh con 622 được mã hóa khác. Nếu điều này được phép ở bộ mã hóa, thì bộ giải mã phải hỗ trợ sơ đồ mã như vậy thậm chí khi sơ đồ giải mã hiếm khi được sử dụng. Bằng cách không cho phép các khe hở và các sự chồng lấn ảnh con 622, độ phức tạp của bộ giải mã có thể được giảm khi bộ giải mã không được yêu cầu để chiếm các khe hở và các sự chồng lấn tiềm năng khi xác định các kích thước ảnh con 631 và các vị trí 632. Hơn nữa, việc không cho phép các khe hở và các sự chồng lấn ảnh con 622 làm giảm độ phức tạp của các quy trình xử lý RDO ở bộ mã hóa. Điều này là bởi vì bộ mã hóa có thể bỏ qua việc xem xét các trường hợp khe hở và chồng lấn khi lựa chọn việc mã hóa cho chuỗi video. Theo đó, việc tránh khỏi các khe hở và các sự chồng lấn có thể làm giảm việc sử dụng 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 giản lược minh họa cơ chế 700 ví dụ để liên kết các lát 724 với sự phối trí ảnh con 722. Ví dụ, cơ chế 700 có thể được áp dụng tới ảnh 600. Hơn nữa, cơ chế 700 có thể được áp dụng dựa vào dữ liệu ở dòng bit 500, ví dụ bởi hệ thống bộ mã hóa/giải mã 200, bộ mã hóa 300, và/hoặc bộ giải mã 400. Hơn nữa, cơ chế 700 có thể được ứng dụng để hỗ trợ việc mã hóa và giải mã theo phương pháp 100.

Cơ chế 700 có thể được áp dụng tới các lát 724 trong ảnh con 722, chẳng hạn như lần lượt các lát 524/525 và các ảnh con 522/523. Theo ví dụ được thể hiện, ảnh con 722 bao gồm lát thứ nhất 724a, lát thứ hai 724b, và lát thứ ba 724c. Các đoạn đầu lát cho mỗi trong số các lát 724 bao gồm ID ảnh con 733 cho ảnh con 722. Bộ giải mã có thể so khớp ID ảnh con 733 từ đoạn đầu lát với ID ảnh con 733 trong SPS. Bộ giải mã sau đó có thể xác định vị trí 732 và kích thước của ảnh con

722 từ SPS dựa vào ID ảnh con 733. Việc sử dụng vị trí 732, ảnh con 722 có thể được đặt liên quan đến mẫu trên cùng bên trái ở góc trên cùng bên trái 742 của ảnh. Kích thước có thể được sử dụng để thiết đặt độ cao và độ rộng của ảnh con 722 liên quan đến vị trí 732. Các lát 724 sau đó có thể được bao gồm trong ảnh con 722. Theo đó, các lát 724 có thể được bố trí 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 điều chỉnh lỗi khi các ảnh con bị mất khác không cảnh báo việc giải mã của ảnh con 722. Điều này cũng hỗ trợ các ứng dụng mà chỉ trích ảnh con 722 và tránh khỏi việc truyền các ảnh con khác. Do đó, các ID ảnh con 733 hỗ trợ tính năng được tăng và/hoặc hiệu quả mã hóa được tăng, mà làm giảm việc sử dụng 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 giản lược minh họa ảnh 800 ví dụ khác được phân chia thành các ảnh con 822. Ảnh 800 về cơ bản có thể giống với ảnh 600. Ngoài ra, ảnh 800 có thể được mã hóa trong và được giải mã từ dòng bit 500, ví dụ bởi hệ thống bộ mã hóa/giải mã 200, bộ mã hóa 300, và/hoặc bộ giải mã 400. Hơn nữa, ảnh 800 có thể được phân chia và/hoặc được bao gồm ở dòng bit con 501 để hỗ trợ việc mã hóa và giải mã theo phương pháp 100 và/hoặc cơ chế 700.

Ảnh 800 bao gồm các ảnh con 822, mà về cơ bản có thể giống với các ảnh con 522, 523, 622, và/hoặc 722. Các ảnh con 822 được chia thành các CTU 825. CTU 825 là đơn vị mã hóa cơ bản trong các hệ thống mã video được chuẩn hóa. CTU 825 được chia nhỏ bởi cây mã hóa thành các khối mã hóa, mà được mã hóa theo việc dự báo liên kết hoặc việc dự báo bên trong. Như được thể hiện, một số ảnh con 822a bị ràng buộc để bao gồm các độ rộng ảnh con và các độ cao ảnh con mà là các bội số của kích thước CTU 825. Theo ví dụ được thể hiện, các ảnh con 822a có độ cao là sáu CTU 825 và độ rộng là năm CTU 825. Sự ràng buộc này được loại bỏ đối với các ảnh con 822b được bố trí trên ranh giới bên phải các ảnh 801 và đối với các ảnh con 822c được bố trí trên ranh giới dưới cùng các ảnh 802. Theo ví dụ được thể hiện, các ảnh con 822b có độ rộng giữa năm và sáu CTU 825. Tuy nhiên, các ảnh con 822b mà không được bố trí trên ranh giới dưới cùng các ảnh 802 vẫn bị ràng buộc để duy trì độ cao ảnh con mà là bội số của kích thước CTU 825. Theo ví dụ được thể hiện, các ảnh con 822c có độ cao giữa sáu và bảy

CTU 825. Tuy nhiên, các ảnh con 822c mà không được bố trí trên ranh giới bên phải các ảnh 801 vẫn bị ràng buộc để duy trì độ rộng ảnh con mà là bội số 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 độ cao và các độ rộng mà là các bội số của kích thước CTU 825. Điều này có thể ngăn ngừa các ảnh con 822 khởi tạo một cách chính xác với nhiều sự phối trí ảnh, ví dụ với ảnh 800 mà chứa tổng độ rộng hoặc độ cao mà không là bội số của kích thước CTU 825. Bằng cách cho phép các ảnh con dưới cùng 822c và các ảnh con bên phải 822b lần lượt bao gồm các độ cao và các độ rộng, mà không là các bội số 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 tăng tính năng bộ mã hóa và bộ giải mã. Hơn nữa, tính năng được tăng cho phép bộ mã hóa mã hóa các ảnh một cách hữu hiệu hơn, mà làm giảm việc sử dụng 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, sáng chế mô tả các thiết kế đối với việc phân chia ảnh dựa vào ảnh con trong mã hóa video. Ảnh con là diện tích hình chữ nhật nằm trong ảnh mà có thể được giải mã một cách độc lập nhờ sử dụng quy trình giải mã giống như được sử dụng cho ảnh. Sáng chế đề cập đến việc truyền tín hiệu của các ảnh con trong chuỗi video và/hoặc dòng bit được mã hóa cũng như quy trình xử lý cho việc trích ảnh con. Các phần mô tả của các kỹ thuật dựa vào VVC bởi JVET của ITU-T và ISO/IEC. Tuy nhiên, các kỹ thuật cũng áp dụng tới các thông số kỹ thuật bộ mã hóa/giải mã video khác. Phần sau đây là các phương án ví dụ được mô tả ở đây. Các phương án như vậy có thể được áp dụng riêng hoặc kết hợp.

Thông tin liên quan đến các ảnh con mà có thể có mặt trong chuỗi video được mã hóa (coded video sequence - CVS) có thể được truyền tín hiệu trong tập hợp thông số mức chuỗi, chẳng hạn như SPS. Việc truyền tín hiệu như vậy có thể bao gồm thông tin sau đây. Số lượng các ảnh con mà có mặt trong mỗi ảnh của CVS có thể được truyền tín hiệu trong SPS. Trong ngữ cảnh của SPS hoặc CVS, các ảnh con được sắp xếp cho tất cả các bộ phận truy cập (access unit - AU) có thể

được gọi chung là chuỗi ảnh con. Vòng lặp để xác định thêm thông tin 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 cùng bên trái của ảnh con và mẫu độ chói góc trên cùng bên trái của ảnh), và kích thước của ảnh con. Ngoài ra, SPS có thể truyền tín hiệu xem mỗi ảnh con là ảnh con bị ràng buộc chuyển động (chứa tính năng của MCTS) hay không. Thông tin hồ sơ, cấp, và mức cho mỗi ảnh con cũng có thể được truyền tín hiệu hoặc có thể dẫn ra ở bộ giải mã. Thông tin như vậy có thể được ứng dụng để xác định thông tin hồ sơ, cấp, và mức cho dòng bit được tạo bằng cách trích các ảnh con từ dòng bit gốc. Hồ sơ và cấp của mỗi ảnh con có thể được dẫn ra giống như toàn bộ hồ sơ và cấp của dòng bit. Mức cho mỗi ảnh con có thể được truyền tín hiệu một cách tường minh. Việc truyền tín hiệu như vậy có thể có mặt trong vòng lặp được chứa trong SPS. Các thông số bộ giải mã tham chiếu giả thuyết (hypothetical reference decoder - HRD) mức chuỗi có thể được truyền tín 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 truyền tín hiệu ở dòng bit. Khi ảnh con của các ảnh trong CVS được trích, mỗi bộ phận truy cập ở dòng bit mới có thể không chứa ảnh con. Trong trường hợp này, ảnh ở mỗi AU ở dòng bit mới không được phân chia thành nhiều ảnh con. Vì vậy không có nhu cầu truyền tín hiệu các đặc tính ảnh con chẳng hạn như vị trí và kích thước trong SPS do thông tin như vậy có thể được dẫn ra từ các đặc tính ảnh. Tuy nhiên, việc nhận dạng ảnh con vẫn có thể được truyền tín hiệu khi ID có thể được đề cập tới bởi các bộ phận VCL NAL/các nhóm ô mà được bao gồm trong ảnh con được trích. Điều này có thể cho phép các ID ảnh con vẫn giống nhau khi trích ảnh con.

Vị trí của ảnh con trong ảnh (dịch vị x và dịch vị y) có thể được truyền tín hiệu trong các bộ phận của các mẫu độ chói. Vị trí biểu diễn khoảng cách giữa mẫu độ chói góc trên cùng bên trái của ảnh con và mẫu độ chói góc trên cùng bên trái của ảnh. Theo cách khác, vị trí của ảnh con trong ảnh có thể được truyền tín

hiệu trong các bộ phận của kích thước khối độ chói mã hóa nhỏ nhất (MinCbSizeY). Theo cách khác, bộ phận của các dịch vị vị trí ảnh con có thể được chỉ báo một cách tường minh bởi thành phần cú pháp trong tập hợp thông số. Bộ phận có thể là CtbSizeY, MinCbSizeY, mẫu sắc độ, hoặc các trị số khác.

Kích thước của ảnh con (độ rộng ảnh con và độ cao ảnh con) có thể được truyền tín hiệu trong các bộ phận của các mẫu độ chói. Theo cách khác, kích thước của ảnh con có thể được truyền tín hiệu trong các bộ phận của kích thước khối độ chói mã hóa nhỏ nhất (MinCbSizeY). Theo cách khác, bộ phận của các trị số kích thước ảnh con có thể được chỉ báo một cách tường minh bởi thành phần cú pháp trong tập hợp thông số. Bộ phận có thể là CtbSizeY, MinCbSizeY, mẫu sắc độ, hoặc các trị số khác. Khi ranh giới bên phải của ảnh con không trùng khớp với ranh giới bên phải của ảnh, độ rộng của ảnh con có thể được yêu cầu là bội số nguyên của kích thước CTU độ chói (CtbSizeY). Tương tự như vậy, khi ranh giới dưới cùng của ảnh con không trùng khớp với ranh giới dưới cùng của ảnh, độ cao của ảnh con có thể được yêu cầu là bội số nguyên của kích thước CTU độ chói (CtbSizeY). Nếu độ rộng của ảnh con không là bội số nguyên của kích thước CTU độ chói, thì ảnh con có thể được yêu cầu để được bố trí ở vị trí ngoài cùng bên phải trong ảnh. Tương tự như vậy, nếu độ cao của ảnh con không là bội số nguyên của kích thước CTU độ chói, thì ảnh con có thể được yêu cầu để được bố trí ở vị trí dưới cùng nhất trong ảnh. Trong một số trường hợp, độ rộng của ảnh con có thể được truyền tín hiệu trong các bộ phận của kích thước CTU độ chói, nhưng độ rộng của ảnh con không là bội số nguyên của kích thước CTU độ chói. Trong trường hợp này, độ rộng thực tế trong các mẫu độ chói có thể được dẫn ra dựa vào vị trí dịch vị của ảnh con. Độ rộng của ảnh con có thể được dẫn ra dựa vào kích thước CTU độ chói và độ cao của ảnh có thể được dẫn ra dựa vào các mẫu độ chói. Tương tự như vậy, độ cao của ảnh con có thể được truyền tín hiệu trong các bộ phận của kích thước CTU độ chói, nhưng độ cao của ảnh con không là bội số nguyên của kích thước CTU độ chói. Trong trường hợp như vậy, độ cao thực tế trong mẫu sắc độ có thể được dẫn ra dựa vào vị trí dịch vị của ảnh con. Độ cao của ảnh con có thể được dẫn ra dựa vào kích thước CTU độ chói và độ cao của ảnh có thể được dẫn ra dựa vào các mẫu độ chói.

Đối với ảnh con bất kỳ, ID ảnh con có thể khác với chỉ số ảnh con. Chỉ số ảnh con có thể là chỉ số của ảnh con khi được truyền tín hiệu trong vòng lặp của các ảnh con trong SPS. ID ảnh con có thể là chỉ số của ảnh con theo thứ tự quét mảnh ảnh con trong ảnh. Khi trị số của ID ảnh con của mỗi ảnh con giống với chỉ số ảnh con, ID ảnh con có thể được truyền tín hiệu hoặc được dẫn ra. Khi ID ảnh con của mỗi ảnh con khác với chỉ số ảnh con, ID ảnh con được truyền tín hiệu một cách tường minh. Số lượng các bit cho việc truyền tín hiệu của các ID ảnh con có thể được truyền tín hiệu trong cùng tập hợp thông số mà chứa các đặc tính ảnh con (ví dụ, trong SPS). Một số trị số đối với ID ảnh con có thể được giữ lại nhằm các mục đích nhất định. Ví dụ, khi các đoạn đầu nhóm ô chứa các ID ảnh con để xác định ảnh con nào chứa nhóm ô, trị số không có thể được giữ lại và không được sử dụng cho các ảnh con để đảm bảo rằng một số ít bit thứ nhất của đoạn đầu nhóm ô tất cả không là các không để tránh khỏi sự bao gồm tình cờ của mã ngăn ngừa giả lập. Trong các trường hợp tùy chọn trong đó các ảnh con của ảnh không bao trùm toàn bộ diện tích của ảnh không có khe hở và không chồng lấn, trị số (ví dụ, trị số một) có thể được giữ lại cho các nhóm ô mà không có phần của ảnh con bất kỳ. Theo cách khác, ID ảnh con của diện tích còn lại được truyền tín hiệu một cách tường minh. Số lượng các bit để truyền tín hiệu ID ảnh con có thể bị ràng buộc như sau. Khoảng trị số sẽ đủ để nhận dạng duy nhất tất cả các ảnh con trong ảnh, bao gồm các trị số được giữ lại của ID ảnh con. Ví dụ, số lượng nhỏ nhất của các bit cho ID ảnh con có thể là trị số của $\text{Ceil}(\log_2(\text{số các ảnh con trong ảnh} + \text{số ID ảnh con được giữ lại}))$.

Có thể bị ràng buộc rằng sự liên kết của các ảnh con phải bao trùm toàn bộ ảnh không có khe hở và không chồng lấn. Khi sự ràng buộc này được áp dụng, cho mỗi ảnh con, có thể có mặt để xác định xem ảnh con là ảnh con bị ràng buộc chuyển động hay không, mà chỉ báo ảnh con có thể được trích. Theo cách khác, sự liên kết của các ảnh con có thể không bao trùm toàn bộ ảnh, nhưng các sự chồng lấn có thể không được phép.

Các ID ảnh con có thể có mặt ngay sau đoạn đầu đơn vị NAL để hỗ trợ quy trình trích ảnh con mà không yêu cầu bộ trích phân tách phần còn lại của các bit đơn vị NAL. Đối với các bộ phận VCL NAL, ID ảnh con có thể có mặt trong

các bit thứ nhất của các đoạn đầu nhóm ô. Đối với đơn vị NAL không VCL, phần sau đây có thể áp dụng. Đối với SPS, ID ảnh con không cần có mặt ngay sau đoạn đầu đơn vị NAL. Đối với PPS, nếu tất cả các nhóm ô của cùng ảnh bị ràng buộc để đề cập đến cùng PPS, thì ID ảnh con không cần có mặt ngay sau đoạn đầu đơn vị NAL của nó. Nếu các nhóm ô của cùng ảnh được phép đề cập đến các PPS khác nhau, thì ID ảnh con có thể có mặt trong các bit thứ nhất của PPS (ví dụ, ngay sau đoạn đầu đơn vị NAL). Trong trường hợp này, các nhóm ô bất kỳ của một ảnh có thể được phép dùng chung cùng PPS. Theo cách khác, khi các nhóm ô của cùng ảnh được phép đề cập đến các PPS khác nhau, và nhóm ô khác của cùng ảnh cũng được phép dùng chung cùng PPS, không có ID ảnh con có thể có mặt trong cú pháp PPS. Theo cách khác, khi các nhóm ô của cùng ảnh được phép đề cập đến các PPS khác nhau, và nhóm ô khác của cùng ảnh cũng được phép dùng chung cùng PPS, danh mục của các ID ảnh con có thể có mặt trong cú pháp PPS. Danh mục chỉ báo các ảnh con trong đó PPS áp dụng. Đối với các đơn vị NAL không VCL khác, nếu bộ phận không VCL áp dụng tới mức ảnh trở nên (ví dụ, dấu phân cách bộ phận truy cập, kết thúc chuỗi, kết thúc dòng bit, v.v.), sau đó ID ảnh con có thể không có mặt ngay sau đoạn đầu đơn vị NAL. Theo cách khác, ID ảnh con có thể có mặt ngay sau đoạn đầu đơn vị NAL.

Với việc truyền tín hiệu SPS ở trên, việc phân chia ô nằm trong các ảnh con riêng có thể được truyền tín hiệu trong PPS. Các nhóm ô nằm trong cùng ảnh có thể được phép đề cập đến các PPS khác nhau. Trong trường hợp này, việc tạo nhóm ô chỉ có thể nằm trong mỗi ảnh con. Khái niệm tạo nhóm ô là phân chia ảnh con thành các ô.

Theo cách khác, tập hợp thông số để mô tả việc phân chia ô nằm trong các ảnh con riêng được định rõ. Tập hợp thông số như vậy có thể được gọi là tập hợp thông số ảnh con (Sub-Picture Parameter Set - SPPS). SPPS đề cập đến SPS. Thành phần cú pháp đề cập đến SPS ID có mặt trong SPPS. SPPS có thể chứa ID ảnh con. Nhằm các mục đích trích ảnh con, thành phần cú pháp đề cập đến ID ảnh con là thành phần cú pháp thứ nhất trong SPPS. SPPS chứa cấu trúc ô (ví dụ, một số cột, một số hàng, khoảng cách ô đồng nhất, v.v.) SPPS có thể chứa cờ để chỉ báo xem bộ lọc vòng lặp được phép qua các ranh giới ảnh con được kết hợp hay không.

Theo cách khác, các đặc tính ảnh con cho mỗi ảnh con có thể được truyền tín hiệu trong SPPS thay vì trong SPS. Việc phân chia ô nằm trong các ảnh con riêng vẫn có thể được truyền tín hiệu trong PPS. Các nhóm ô nằm trong cùng ảnh được phép đề cập đến các PPS khác nhau. Khi SPPS được kích hoạt, SPPS kéo dài cho chuỗi của các AU liên tiếp theo thứ tự giải mã. Tuy nhiên, SPPS có thể được hủy kích hoạt/được kích hoạt ở AU mà không bắt đầu của CVS. Ở thời điểm bất kỳ trong suốt thời gian quy trình giải mã của dòng bit lớp đơn với nhiều ảnh con ở một số AU, nhiều SPPS có thể hoạt động. SPPS có thể được dùng chung bởi các ảnh con khác nhau của AU. Theo cách khác, SPPS và PPS có thể được hợp nhất vào một tập hợp thông số. Trong trường hợp như vậy, tất cả các nhóm ô của cùng ảnh có thể không được yêu cầu để đề cập đến cùng PPS. Sự ràng buộc có thể được áp dụng sao cho tất cả các nhóm ô trong cùng ảnh con có thể đề cập đến cùng tập hợp thông số xuất phát từ bộ hợp nhất giữa SPPS và PPS.

Số lượng các bit được sử dụng để truyền tín hiệu ID ảnh con có thể được truyền tín hiệu ở đoạn đầu đơn vị NAL. Khi có mặt ở đoạn đầu đơn vị NAL thông tin như vậy có thể hỗ trợ các quy trình trích ảnh con trong việc phân tách trị số ID ảnh con ở đầu của tải trọng của đơn vị NAL (ví dụ, một số ít bit thứ nhất ngay sau đoạn đầu đơn vị NAL). Đối với việc truyền tín hiệu như vậy, một vài trong số các bit được giữ lại (ví dụ, bảy bit được giữ lại) ở đoạn đầu đơn vị NAL có thể được sử dụng để tránh khỏi việc làm tăng độ dài của đoạn đầu đơn vị NAL. Số lượng các bit để truyền tín hiệu như vậy có thể bao trùm trị số của sub-picture-ID-bit-len. Ví dụ, bốn bit trong số bảy bit được giữ lại của đoạn đầu đơn vị NAL các VVC có thể được sử dụng nhằm mục đích này.

Khi giải mã ảnh con, vị trí của mỗi khối cây mã hóa (ví dụ, xCtb và yCtb) có thể được điều chỉnh tới vị trí mẫu độ chói thực tế trong ảnh thay vì vị trí mẫu độ chói trong ảnh con. Theo cách này, việc trích của ảnh con được đồng bộ trí từ mỗi ảnh tham chiếu có thể được tránh khỏi khi khối cây mã hóa được giải mã dựa vào ảnh thay vì ảnh con. Để điều chỉnh vị trí của khối cây mã hóa, các biến SubpictureXOffset và SubpictureYOffset có thể được dẫn ra dựa vào vị trí ảnh con (subpic_x_offset và subpic_y_offset). Các trị số của các biến có thể lần lượt được

bổ sung vào các trị số của vị trí mẫu độ chói các tọa độ x và y, của mỗi khối cây mã hóa trong ảnh con.

Quy trình trích ảnh con có thể được định rõ như sau. Đầu vào tới quy trình xử lý là ảnh con đích được trích. Điều này có thể dưới dạng của 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 quyết bằng cách phân tách thông tin ảnh con trong SPS. Đối với các đơn vị NAL không VCL, phần sau đây áp dụng. Các thành phần cú pháp trong SPS liên quan đến kích thước và mức ảnh có thể được cập nhật với thông tin kích thước và mức của ảnh con. Các đơn vị NAL không VAL sau đây được giữ mà không thay đổi: PPS, dấu phân cách bộ phận truy cập (Access Unit Delimiter - AUD), kết thúc chuỗi (End of Sequence - EOS), kết thúc dòng bit (End of Bitstream - EOB), và các đơn vị NAL không VAL bất kỳ mà có thể áp dụng được tới mức ảnh trở nên. Các đơn vị NAL không VAL còn lại với ID ảnh con không bằng ID ảnh con đích có thể được loại bỏ. Các bộ phận VCL NAL với ID ảnh con không bằng ID ảnh con đích cũng có thể được loại bỏ.

Bản tin SEI lồng ảnh con mức chuỗi có thể được sử dụng cho việc lồng của các bản tin SEI mức AU hoặc mức ảnh con cho tập hợp của các ảnh con. Điều này có thể bao gồm giai đoạn đệm, thời điểm ảnh, và các bản tin SEI không HRD. Cú pháp và ngữ nghĩa của bản tin SEI lồng ảnh con này có thể như sau. Đối với các thao tác 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 xem có thể được yêu cầu và được giải mã bởi trình phát OMAF. Do đó, bản tin SEI mức chuỗi được sử dụng để mang thông tin của tập hợp của các chuỗi ảnh con mà bao trùm chung vùng ảnh hình chữ nhật. Thông tin có thể được sử dụng bởi các hệ thống, và thông tin là chỉ báo của khả năng giải mã được yêu cầu cũng như tốc độ bit của tập hợp của các chuỗi ảnh con. Thông tin chỉ báo mức của dòng bit chỉ bao gồm tập hợp của các chuỗi ảnh con. Thông tin này cũng chỉ báo tốc độ bit của dòng bit chứa chỉ tập hợp của các chuỗi ảnh con. Một cách tùy chọn, quy trình trích dòng bit con có thể được quy định cho tập hợp của các chuỗi ảnh con. Lợi ích của việc làm này là dòng bit chỉ bao gồm tập hợp của các chuỗi ảnh con cũng có thể thích hợp. Nhược điểm ở chỗ xem xét các khả năng kích thước

công xem khác nhau có thể có nhiều tập hợp như vậy ngoài các số khả thi lớn sẵn có của các chuỗi ảnh con riêng.

Theo ví dụ khác, một hoặc nhiều trong số các ví dụ được bộc lộ có thể được thực hiện như sau. Ảnh con có thể được định rõ là vùng hình chữ nhật của một hoặc nhiều nhóm ô nằm trong ảnh. Quy trình phân tách nhị phân được phép có thể được định rõ như sau. Các đầu vào tới quy trình xử lý này là: chế độ phân tách nhị phân `btSplit`, độ rộng khối mã hóa `cbWidth`, độ cao khối mã hóa `cbHeight`, vị trí (`x0`, `y0`) của mẫu độ chói trên cùng bên trái của khối mã hóa được xem xét liên quan đến mẫu độ chói trên cùng bên trái của ảnh, độ sâu cây nhiều loại `mttDepth`, độ sâu cây nhiều loại lớn nhất với dịch vị `maxMttDepth`, kích thước cây nhị phân lớn nhất `maxBtSize`, và chỉ số phân chia `partIdx`. Đầu ra của quy trình xử lý này là biến `allowBtSplit`.

	<code>btSplit</code> <code>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>

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

Các biến `parallelTtSplit` và `cbSize` được dẫn ra như được quy định ở trên. Biến `allowBtSplit` được dẫn ra như sau. Nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, thì `allowBtSplit` được thiết đặt bằng sai (FALSE): `cbSize` nhỏ hơn hoặc bằng `MinBtSizeY`, `cbWidth` lớn hơn `maxBtSize`, `cbHeight` lớn hơn `maxBtSize`, và `mttDepth` lớn hơn hoặc bằng `maxMttDepth`. Theo cách khác, nếu tất cả các điều kiện sau đây là đúng (true), thì `allowBtSplit` được thiết đặt bằng sai: `btSplit` bằng `SPLIT_BT_VER`, và `y0 + cbHeight` lớn hơn `SubPicBottomBorderInPic`. Theo cách khác, nếu tất cả các điều kiện sau đây là đúng, thì `allowBtSplit` được thiết đặt bằng sai, `btSplit` bằng `SPLIT_BT_HOR`,

$x0 + cbWidth$ lớn hơn `SubPicRightBorderInPic`, và $y0 + cbHeight$ nhỏ hơn hoặc bằng `SubPicBottomBorderInPic`. Theo cách khác, nếu tất cả các điều kiện sau đây là đúng, thì `allowBtSplit` được thiết đặt bằng sai: `mttDepth` lớn hơn không, `partIdx` bằng một, và `MttSplitMode[ x0 ][ y0 ][ mttDepth - 1 ]` bằng `parallelTtSplit`. Theo cách khác nếu tất cả các điều kiện sau đây là đúng, thì `allowBtSplit` được thiết đặt bằng sai: `btSplit` bằng `SPLIT_BT_VER`, `cbWidth` nhỏ hơn hoặc bằng `MaxTbSizeY`, và `cbHeight` lớn hơn `MaxTbSizeY`. Theo cách khác nếu tất cả các điều kiện sau đây là đúng, thì `allowBtSplit` được thiết đặt bằng sai: `btSplit` bằng `SPLIT_BT_HOR`, `cbWidth` lớn hơn `MaxTbSizeY`, và `cbHeight` nhỏ hơn hoặc bằng `MaxTbSizeY`. Theo cách khác, `allowBtSplit` được thiết đặt bằng đúng.

Quy trình phân tách bậc ba được phép có thể được định rõ như sau. Các đầu vào tới quy trình xử lý này là: chế độ phân tách bậc ba `ttSplit`, độ rộng khối mã hóa `cbWidth`, độ cao khối mã hóa `cbHeight`, vị trí ($x0, y0$) của mẫu độ chói trên cùng bên trái của khối mã hóa được xem xét liên quan đến mẫu độ chói trên cùng bên trái của ảnh, độ sâu cây nhiều loại `mttDepth`, độ sâu cây nhiều loại lớn nhất với dịch vị `maxMttDepth`, và kích thước cây nhị phân lớn nhất `maxTtSize`. Đầu ra của quy trình xử lý này là biến `allowTtSplit`.

	<code>ttSplit == SPLIT_TT_VER</code>	<code>ttSplit == SPLIT_TT_HOR</code>
<code>cbSize</code>	<code>cbWidth</code>	<code>cbHeight</code>

Bản mô tả của `cbSize` dựa vào `ttSplit`.

Biến `cbSize` được dẫn ra như được quy định ở trên. Biến `allowTtSplit` được dẫn ra như sau. Nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, thì `allowTtSplit` được thiết đặt bằng sai: `cbSize` nhỏ hơn hoặc bằng $2 * \text{MinTtSizeY}$, `cbWidth` lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$, `cbHeight` lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$, `mttDepth` lớn hơn hoặc bằng `maxMttDepth`, $x0 + cbWidth$ lớn hơn `SubPicRightBorderInPic`, và $y0 + cbHeight$ lớn hơn `SubPicBottomBorderInPic`. Theo cách khác, `allowTtSplit` được thiết đặt bằng đúng.

Cú pháp và ngữ nghĩa tập hợp thông số chuỗi RBSP như sau.

seq_parameter_set_rbsp() {	Ký hiệu
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` xác định độ rộng của mỗi ảnh được giải mã trong các bộ phận 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` xác định độ cao của mỗi ảnh được giải mã trong các bộ phận 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 xác định số lượng các ảnh con được phân chia trong các ảnh được mã hóa thuộc về chuỗi video được mã hóa. `Subpic_id_len_minus1` cộng 1 xác định số lượng các bit được sử dụng để biểu diễn thành phần cú pháp `subpic_id[i]` trong SPS, `spps_subpic_id` trong SPPS đề cập đến SPS, và `tile_group_subpic_id` trong các đoạn đầu nhóm ô đề cập đến SPS. Trị số của `subpic_id_len_minus1` sẽ trong khoảng từ $\text{Ceil}(\text{Log2}(\text{num_subpic_minus1} + 2))$ đến tám. `Subpic_id[i]` xác định ID ảnh con của ảnh con thứ i trong số các ảnh đề cập đến SPS. Độ dài của `subpic_id[i]` là `subpic_id_len_minus1 + 1` bit. Trị số của `subpic_id[i]` sẽ lớn hơn không. `Subpic_level_idc[i]` chỉ báo mức trong đó CVS được xuất phát từ việc trích của các ảnh con thứ i phù hợp với các yêu cầu tài nguyên được quy định. Các dòng bit sẽ không chứa các trị số của `subpic_level_idc[i]` khác ngoài các dòng bit được quy định. Các trị số khác của `subpic_level_idc[i]` được giữ lại. Khi không có mặt, trị số của `subpic_level_idc[i]` được suy luận bằng trị số của `general_level_idc`.

`Subpic_x_offset[i]` xác định dịch vị theo chiều ngang của góc trên cùng bên trái của ảnh con thứ i liên quan đến góc trên cùng bên trái của ảnh. Khi không có mặt, trị số của `subpic_x_offset[i]` được suy luận bằng 0. Trị số của dịch vị x ảnh con được dẫn ra như sau: `SubpictureXOffset[i] = subpic_x_offset[i]`. `Subpic_y_offset[i]` xác định dịch vị theo chiều thẳng đứng của góc trên cùng bên trái của ảnh con thứ i liên quan đến góc trên cùng bên trái của ảnh. Khi không có mặt, trị số của `subpic_y_offset[i]` được suy luận bằng không. Trị số của dịch vị y ảnh con được dẫn ra như sau: `SubpictureYOffset[i] = subpic_y_offset[i]`. `Subpic_width_in_luma_samples[i]` xác định độ rộng của ảnh con thứ i được giải

mã trong đó SPS này là SPS hoạt động. Khi tổng của $\text{SubpictureXOffset}[i]$ và $\text{subpic_width_in_luma_samples}[i]$ nhỏ hơn $\text{pic_width_in_luma_samples}$, trị số của $\text{subpic_width_in_luma_samples}[i]$ sẽ là bội số nguyên của CtbSizeY . Khi không có mặt, trị số của $\text{subpic_width_in_luma_samples}[i]$ được suy luận bằng trị số của $\text{pic_width_in_luma_samples}$. $\text{Subpic_height_in_luma_samples}[i]$ xác định độ cao của ảnh con thứ i được giải mã trong đó SPS này là SPS hoạt động. Khi tổng của $\text{SubpictureYOffset}[i]$ và $\text{subpic_height_in_luma_samples}[i]$ nhỏ hơn $\text{pic_height_in_luma_samples}$, trị số của $\text{subpic_height_in_luma_samples}[i]$ sẽ là bội số nguyên của CtbSizeY . Khi không có mặt, trị số của $\text{subpic_height_in_luma_samples}[i]$ được suy luận bằng trị số của $\text{pic_height_in_luma_samples}$.

Đây là yêu cầu của sự phù hợp dòng bit rằng sự liên kết của các ảnh con sẽ bao trùm toàn bộ diện tích của ảnh mà không có khe hở và chồng lấn. $\text{Subpic_motion_constrained_flag}[i]$ bằng một xác định ảnh con thứ i là ảnh con bị ràng buộc chuyển động theo thời gian. $\text{Subpic_motion_constrained_flag}[i]$ bằng không xác định ảnh con thứ i có thể hoặc có thể không là ảnh con bị ràng buộc chuyển động theo thời gian. Khi không có mặt, trị số của $\text{subpic_motion_constrained_flag}$ được suy luận bằng không.

Các biến $\text{SubpicWidthInCtbsY}$, $\text{SubpicHeightInCtbsY}$, SubpicSizeInCtbsY , $\text{SubpicWidthInMinCbsY}$, $\text{SubpicHeightInMinCbsY}$, $\text{SubpicSizeInMinCbsY}$, $\text{SubpicSizeInSamplesY}$, $\text{SubpicWidthInSamplesC}$, và $\text{SubpicHeightInSamplesC}$ được dẫn ra như sau:

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

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

$$\text{SubPicRightBorderInPic}[i] =$$

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

$$\text{SubPicBottomBorderInPic}[i] =$$

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

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

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

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

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

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

$$\text{SubpicSizeInMinCbsY}[i] =$$

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

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

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

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

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

Cú pháp và ngữ nghĩa tập hợp thông số ảnh con RBSP như sau.

sub_pic_parameter_set_rbsp() {	Ký hiệu
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_column_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_column_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. Độ dài của sps_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. Trị số của sps_subpic_parameter_set_id sẽ trong khoảng từ không đến sáu mươi ba. Spps_seq_parameter_set_id xác định trị số của sps_seq_parameter_set_id cho SPS hoạt động. Trị số của sps_seq_parameter_set_id sẽ trong khoảng từ không đến mười lăm. Single_tile_in_subpic_flag bằng một xác định rằng chỉ có một ô ở mỗi ảnh con đề cập đến SPPS. Single_tile_in_subpic_flag bằng không xác định rằng có nhiều hơn một ô ở mỗi ảnh con đề cập đến SPPS. Num_tile_column_minus1 cộng 1 xác định số lượng các cột ô phân chia ảnh con. Num_tile_column_minus1 sẽ trong khoảng từ không đến $\text{PicWidthInCtbsY}[\text{spps_subpic_id}] - 1$. Khi không có mặt, trị số của num_tile_column_minus1 được suy luận bằng không. Num_tile_rows_minus1 cộng 1 xác định số lượng các hàng ô phân chia ảnh con. Num_tile_rows_minus1 sẽ trong khoảng từ không đến

$\text{PicHeightInCtbsY}[\text{spps_subpic_id}] - 1$. Khi không có mặt, trị số của $\text{num_tile_rows_minus1}$ được suy luận bằng không. Biến NumTilesInPic được thiết đặt bằng $(\text{num_tile_column_minus1} + 1) * (\text{num_tile_rows_minus1} + 1)$.

Khi $\text{single_tile_in_subpic_flag}$ bằng không, NumTilesInPic sẽ lớn hơn không. $\text{Uniform_tile_spacing_flag}$ bằng một xác định rằng các ranh giới cột ô và tương tự các ranh giới hàng ô được phân phối đồng đều qua ảnh con. $\text{Uniform_tile_spacing_flag}$ bằng không xác định rằng các ranh giới cột ô và tương tự các ranh giới hàng ô không được phân phối đồng đều qua ảnh con nhưng được truyền tín hiệu một cách tường minh nhờ sử dụng các thành phần cú pháp $\text{tile_column_width_minus1}[i]$ và $\text{tile_row_height_minus1}[i]$. Khi không có mặt, trị số của $\text{uniform_tile_spacing_flag}$ được suy luận bằng một. $\text{Tile_column_width_minus1}[i]$ cộng 1 xác định độ rộng của cột ô thứ i trong các bộ phận của các CTB. $\text{Tile_row_height_minus1}[i]$ cộng 1 xác định độ cao của hàng ô thứ i trong các bộ phận của các CTB.

Các ô sau đây được dẫn ra bằng cách truy xuất quy trình chuyển đổi quét ô và mảnh CTB: danh mục $\text{colWidth}[i]$ cho i trong khoảng từ không đến $\text{num_tile_column_minus1}$, xác định độ rộng của cột ô thứ i trong các bộ phận của các CTB; danh mục $\text{RowHeight}[j]$ cho j trong khoảng từ không đến $\text{num_tile_rows_minus1}$, xác định độ cao của hàng ô thứ j trong các bộ phận của các CTB; danh mục $\text{colBd}[i]$ cho i trong khoảng từ không đến $\text{num_tile_column_minus1} + 1$, xác định vị trí của ranh giới cột ô thứ i trong các bộ phận của các CTB; danh mục $\text{RowBd}[j]$ cho j trong khoảng từ không đến $\text{num_tile_rows_minus1} + 1$, xác định vị trí của ranh giới hàng ô thứ j trong các bộ phận của các CTB; danh mục $\text{ctbAddrRsToTs}[\text{ctbAddrRs}]$ cho ctbAddrRs trong khoảng từ không đến $\text{PicSizeInCtbsY} - 1$, xác định việc chuyển đổi từ địa chỉ CTB trong việc quét mảnh CTB của ảnh đến địa chỉ CTB trong việc quét ô; danh mục $\text{ctbAddrTsToRs}[\text{ctbAddrTs}]$ cho ctbAddrTs trong khoảng từ không đến $\text{PicSizeInCtbsY} - 1$, xác định việc chuyển đổi từ địa chỉ CTB trong việc quét ô đến địa chỉ CTB trong việc quét mảnh CTB của ảnh; danh mục $\text{TileId}[\text{ctbAddrTs}]$ cho ctbAddrTs trong khoảng từ không đến $\text{PicSizeInCtbsY} - 1$, xác định việc chuyển đổi từ địa chỉ CTB trong việc quét ô đến ID ô; danh mục $\text{NumCtusInTile}[\text{tileIdx}]$

cho `tileIdx` trong khoảng từ không đến `PicSizeInCtbsY - 1`, xác định việc chuyển đổi từ chỉ số ô đến số lượng các CTU trong ô; danh mục `FirstCtbAddrTs[ tileIdx ]` cho `tileIdx` trong khoảng từ không đến `NumTilesInPic - 1`, xác định việc chuyển đổi từ ID ô đến địa chỉ CTB trong việc quét ô của CTB thứ nhất trong ô; danh mục `columnWidthInLumaSamples[ i ]` cho `i` trong khoảng từ không đến `num_tile_column_minus1`, xác định độ rộng của cột ô thứ `i` trong các bộ phận của các mẫu độ chói; và danh mục `RowHeightInLumaSamples[ j ]` cho `j` trong khoảng từ không đến `num_tile_rows_minus1`, xác định độ cao của hàng ô thứ `j` trong các bộ phận của các mẫu độ chói. Các trị số của `ColumnWidthInLumaSamples[ i ]` cho `i` trong khoảng từ không đến `num_tile_column_minus1`, và `RowHeightInLumaSamples[ j ]` cho `j` trong khoảng từ không đến `num_tile_rows_minus1`, tất cả sẽ lớn hơn không.

`Loop_filter_across_tiles_enabled_flag` bằng một xác định rằng các thao tác lọc trong vòng lặp có thể được thực hiện qua các ranh giới ô trong các ảnh con đề cập đến SPPS. `Loop_filter_across_tiles_enabled_flag` bằng không xác định rằng các thao tác lọc trong vòng lặp không được thực hiện qua các ranh giới ô trong các ảnh con đề cập đến SPPS. Các thao tác lọc trong vòng lặp bao gồm bộ lọc giải khối, bộ lọc dịch vị thích nghi mẫu, và các thao tác bộ lọc vòng lặp thích nghi. Khi không có mặt, trị số của `loop_filter_across_tiles_enabled_flag` được suy luận bằng một. `Loop_filter_across_subpic_enabled_flag` bằng một xác định rằng các thao tác lọc trong vòng lặp có thể được thực hiện qua các ranh giới ảnh con trong các ảnh con đề cập đến SPPS. `Loop_filter_across_subpic_enabled_flag` bằng không xác định rằng các thao tác lọc trong vòng lặp không được thực hiện qua các ranh giới ảnh con trong các ảnh con đề cập đến SPPS. Các thao tác lọc trong vòng lặp bao gồm bộ lọc giải khối, bộ lọc dịch vị thích nghi mẫu, và các thao tác bộ lọc vòng lặp thích nghi. Khi không có mặt, trị số của `loop_filter_across_subpic_enabled_flag` được suy luận bằng trị số của `loop_filter_across_tiles_enabled_flag`.

Cú pháp và ngữ nghĩa đoạn đầu nhóm ô chung như sau.

<code>tile_group_header() {</code>	Ký hiệu
-------------------------------------	---------

tile_group_subpic_id	u(v)
tile_group_subpic_parameter_set_id	u(v)
...	
}	

Trị số của thành phần cú pháp đoạn đầu nhóm ô tile_group_pic_parameter_set_id và tile_group_pic_order_cnt_lsb sẽ giống nhau trong tất cả các đoạn đầu nhóm ô của ảnh được mã hóa. Trị số của thành phần cú pháp đoạn đầu nhóm ô tile_group_subpic_id sẽ giống nhau trong tất cả các đoạn đầu nhóm ô của ảnh con được mã hóa. Tile_group_subpic_id nhận dạng ảnh con mà nhóm ô thuộc về. Độ dài của tile_group_subpic_id là subpic_id_len_minus1 + 1 bit. Tile_group_subpic_parameter_set_id xác định trị số của sps_subpic_parameter_set_id cho SPPS đang sử dụng. Trị số của tile_group_spps_parameter_set_id sẽ trong khoảng từ không đến sáu mươi ba.

Các ô sau đây được dẫn ra và ghi đè các biến tương ứng được dẫn 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]

$\text{PicSizeInSamplesY} = \text{SubPicSizeInSamplesY}[\text{tile_group_subpic_id}]$

$\text{PicWidthInSamplesC} = \text{SubPicWidthInSamplesC}[\text{tile_group_subpic_id}]$

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

Cú pháp đơn vị cây mã hóa như sau.

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

<code>dual_tree_implicit_qt_split(x0, y0, log2CbSize, cqtDepth) {</code>	Ký hiệu
<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,</code> <code> cqtDepth + 1)</code>	
<code> if(x1 < SubPicRightBorderInPic)</code>	
<code> dual_tree_implicit_qt_split(x1, y0, log2CbSize - 1,</code> <code> cqtDepth + 1)</code>	

if(y1 < SubPicBottomBorderInPic)	
dual_tree_implicit_qt_split(x0, y1, log2CbSize - 1, cqtDepth + 1)	
if(x1 < SubPicRightBorderInPic && y1 < SubPicBottomBorderInPic)	
dual_tree_implicit_qt_split(x1, y1, log2CbSize - 1, cqtDepth + 1)	
} else {	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_LUMA)	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_CHROMA)	
}	
}	

Cú pháp và ngữ nghĩa cây tứ phân mã hóa như sau.

coding_quadtree(x0, y0, log2CbSize, cqtDepth, treeType) {	Ký hiệu
minQtSize = (treeType == DUAL_TREE_CHROMA) ? MinQtSizeC : MinQtSizeY	
maxBtSize = (treeType == DUAL_TREE_CHROMA) ? MaxBtSizeC : MaxBtSizeY	

<pre> if(((x0 + (1 << log2CbSize) <= PicWidthInLumaSamples) ? 1 : 0) + ((y0 + (1 << log2CbSize) <= PicHeightInLumaSamples) ? 1 : 0) + (((1 << log2CbSize) <= maxBtSize) ? 1 : 0)) >= 2 && (1 << log2CbSize) > minQtSize) </pre>	
<pre> qt_split_cu_flag[x0][y0] </pre>	ae(v)
<pre> if(cu_qp_delta_enabled_flag && cqtDepth <= diff_cu_qp_delta_depth) { </pre>	
<pre> IsCuQpDeltaCoded = 0 </pre>	
<pre> CuQpDeltaVal = 0 </pre>	
<pre> CuQgTopLeftX = x0 </pre>	
<pre> CuQgTopLeftY = y0 </pre>	
<pre> } </pre>	
<pre> if(qt_split_cu_flag[x0][y0]) { </pre>	
<pre> x1 = x0 + (1 << (log2CbSize - 1)) </pre>	
<pre> y1 = y0 + (1 << (log2CbSize - 1)) </pre>	
<pre> coding_quadtree(x0, y0, log2CbSize - 1, cqtDepth + 1, treeType) </pre>	
<pre> if(x1 < SubPicRightBorderInPic) </pre>	
<pre> coding_quadtree(x1, y0, log2CbSize - 1, cqtDepth + 1, treeType) </pre>	
<pre> if(y1 < SubPicBottomBorderInPic) </pre>	

coding_quadtree(x0, y1, log2CbSize - 1, cqtDepth + 1, treeType)	
if(x1 < SubPicRightBorderInPic && y1 < SubPicBottomBorderInPic)	
coding_quadtree(x1, y1, log2CbSize - 1, cqtDepth + 1, treeType)	
} else	
multi_type_tree(x0, y0, 1 << log2CbSize, 1 << log2CbSiz e, cqtDepth, 0, 0, 0, treeType)	
}	

Qt_split_cu_flag[x0][y0] xác định xem bộ phận mã hóa được phân tách thành các bộ phận mã hóa với nửa kích thước theo chiều ngang và theo chiều thẳng đứng hay không. Các chỉ số mảng x0, y0 xác định vị trí (x0, y0) của mẫu độ chói trên cùng bên trái của khối mã hóa được xem xét liên quan đến mẫu độ chói 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 trong số các điều kiện sau đây là đúng, thì trị số của qt_split_cu_flag[x0][y0] được suy luận bằng một. $x0 + (1 \ll \log2CbSize)$ lớn hơn SubPicRightBorderInPic và $(1 \ll \log2CbSize)$ lớn hơn MaxBtSizeC nếu treeType bằng DUAL_TREE_CHROMA hoặc lớn hơn MaxBtSizeY theo cách khác. $y0 + (1 \ll \log2CbSize)$ lớn hơn SubPicBottomBorderInPic và $(1 \ll \log2CbSize)$ lớn hơn MaxBtSizeC nếu treeType bằng DUAL_TREE_CHROMA hoặc lớn hơn MaxBtSizeY theo cách khác.

Theo cách khác, nếu tất cả các điều kiện sau đây là đúng, thì trị số của qt_split_cu_flag[x0][y0] được suy luận bằng 1: $x0 + (1 \ll \log2CbSize)$ lớn hơn SubPicRightBorderInPic, $y0 + (1 \ll \log2CbSize)$ lớn hơn SubPicBottomBorderInPic, và $(1 \ll \log2CbSize)$ lớn hơn MinQtSizeC nếu treeType bằng DUAL_TREE_CHROMA hoặc lớn hơn MinQtSizeY theo cách

khác. Theo cách khác, trị số của `qt_split_cu_flag[ x0 ][ y0 ]` được suy luận bằng không.

Cú pháp và ngữ nghĩa cây nhiều loại như sau.

<code>multi_type_tree(x0, y0, cbWidth, cbHeight, cqtDepth, mttDepth, depthOffset, partIdx, treeType) {</code>	Ký hiệu
<code> if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && (x0 + cbWidth <= SubPicRightBorderInPic) && (y0 + cbHeight <= SubPicBottomBorderInPic))</code>	
<code> mtt_split_cu_flag</code>	ae(v)
<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>	ae(v)

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

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 xác định rằng bộ phận mã hóa không được phân tách. Mtt_split_cu_flag bằng một xác định rằng bộ phận mã hóa được phân tách thành hai bộ phận mã hóa nhờ sử dụng việc phân tách nhị phân hoặc thành ba bộ phận mã hóa nhờ sử dụng việc phân tách bậc ba như được chỉ báo bởi thành phần cú pháp mtt_split_cu_binary_flag. Phân tách nhị phân hay bậc ba có thể hoặc theo chiều thẳng đứng hoặc theo chiều ngang như được chỉ báo bởi thành phần cú pháp mtt_split_cu_vertical_flag. Khi mtt_split_cu_flag không có mặt, trị số của mtt_split_cu_flag được suy luận như sau. Nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, thì trị số của mtt_split_cu_flag được suy luận bằng 1: $x0 + cbWidth$ lớn hơn SubPicRightBorderInPic, và $y0 + cbHeight$ lớn hơn SubPicBottomBorderInPic. Theo cách khác, trị số của mtt_split_cu_flag được suy luận bằng không.

Quy trình dẫn cho việc dự báo vector chuyển động độ chói theo thời gian như sau. Các đầu ra của quy trình xử lý này là: dự báo vector chuyển động mvLXCol theo 1/16 độ chính xác mẫu phân số, và cờ khả dụng availableFlagLXCol. Biến currCb xác định khối mã hóa độ chói hiện thời ở vị trí độ chói (xCb, yCb). Các biến mvLXCol và availableFlagLXCol được dẫn ra như sau. Nếu tile_group_temporal_mvp_enabled_flag bằng không, hoặc nếu ảnh tham chiếu là ảnh hiện thời, thì cả hai bộ phận của mvLXCol được thiết đặt bằng không và availableFlagLXCol được thiết đặt bằng không. Theo cách khác

(tile_group_temporal_mvp_enabled_flag bằng một và ảnh tham chiếu không là ảnh hiện thời), các bước theo thứ tự sau đây áp dụng. Vector chuyển động được sắp xếp dưới cùng bên phải được dẫn ra như sau:

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

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

Nếu $yCb \gg CtbLog2SizeY$ bằng $yColBr \gg CtbLog2SizeY$, $yColBr$ nhỏ hơn $SubPicBottomBorderInPic$ và $xColBr$ nhỏ hơn $SubPicRightBorderInPic$, thì phần sau đây áp dụng. Biến $colCb$ xác định khối mã hóa độ chói bao trùm vị trí được cải biến được đưa ra bởi $((xColBr \gg 3) \ll 3, (yColBr \gg 3) \ll 3)$ bên trong ảnh được sắp xếp được quy định bởi $ColPic$. Vị trí độ chói $(xColCb, yColCb)$ được thiết đặt bằng mẫu trên cùng bên trái của khối mã hóa độ chói được sắp xếp được quy định bởi $colCb$ liên quan đến mẫu độ chói trên cùng bên trái của ảnh được sắp xếp được quy định bởi $ColPic$. Quy trình dẫn ra cho các vector chuyển động được sắp xếp được truy xuất với $currCb$, $colCb$, $(xColCb, yColCb)$, $refIdxLX$ và $sbFlag$ được thiết đặt bằng không là các đầu vào, và đầu ra được gán tới $mvLXCol$ và $availableFlagLXCol$. Theo cách khác, cả hai bộ phận của $mvLXCol$ được thiết đặt bằng không và $availableFlagLXCol$ được thiết đặt bằng không.

Quy trình dẫn ra cho các ứng viên hợp nhất hình tam giác theo thời gian như sau. Các biến $mvLXColC0$, $mvLXColC1$, $availableFlagLXColC0$ và $availableFlagLXColC1$ được dẫn ra như sau. Nếu $tile_group_temporal_mvp_enabled_flag$ bằng không, thì cả hai bộ phận của $mvLXColC0$ và $mvLXColC1$ được thiết đặt bằng không và $availableFlagLXColC0$ và $availableFlagLXColC1$ được thiết đặt bằng không. Theo cách khác ($tile_group_temporal_mvp_enabled_flag$ bằng 1), các bước theo thứ tự sau đây áp dụng. Vector chuyển động được sắp xếp dưới cùng bên phải $mvLXColC0$ được dẫn ra như sau:

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

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

Nếu $yCb \gg CtbLog2SizeY$ bằng $yColBr \gg CtbLog2SizeY$, $yColBr$ nhỏ hơn $SubPicBottomBorderInPic$ và $xColBr$ nhỏ hơn $SubPicRightBorderInPic$, thì phần sau đây áp dụng. Biến $colCb$ xác định khối mã hóa độ chói bao trùm vị trí được cải biến được đưa ra bởi $((xColBr \gg 3) \ll 3, (yColBr \gg 3) \ll 3)$ bên trong ảnh được sắp xếp được quy định bởi $ColPic$. Vị trí độ chói $(xColCb, yColCb)$ được thiết đặt bằng mẫu trên cùng bên trái của khối mã hóa độ chói được sắp xếp được quy định bởi $colCb$ liên quan đến mẫu độ chói trên cùng bên trái của ảnh được sắp xếp được quy định bởi $ColPic$. Quy trình dẫn ra cho các vector chuyển động được sắp xếp được truy xuất với $currCb$, $colCb$, $(xColCb, yColCb)$, $refIdxLXC0$ và $sbFlag$ được thiết đặt bằng không là các đầu vào, và đầu ra được gán tới $mvLXColC0$ và $availableFlagLXColC0$. Theo cách khác, cả hai bộ phận của $mvLXColC0$ được thiết đặt bằng không và $availableFlagLXColC0$ được thiết đặt bằng không.

Quy trình dẫn ra cho các ứng viên hợp nhất vector chuyển động điểm điều khiển afin được cấu trúc như sau. Vector chuyển động điểm điều khiển (được sắp xếp dưới cùng bên phải) thứ tư $cpMvLXCorner[3]$, chỉ số tham chiếu $refIdxLXCorner[3]$, cờ sử dụng danh mục dự báo $predFlagLXCorner[3]$ và cờ khả dụng $availableFlagCorner[3]$ với X là 0 và 1 được dẫn ra như sau. Các chỉ số tham chiếu cho ứng viên hợp nhất theo thời gian, $refIdxLXCorner[3]$, với X là không hoặc một, được thiết đặt bằng không. Các biến $mvLXCol$ và $availableFlagLXCol$, với X là không hoặc một, được dẫn ra như sau. Nếu $tile_group_temporal_mvp_enabled_flag$ bằng không, thì cả hai bộ phận của $mvLXCol$ được thiết đặt bằng không và $availableFlagLXCol$ được thiết đặt bằng không. Theo cách khác ($tile_group_temporal_mvp_enabled_flag$ bằng một), phần sau đây áp dụng:

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

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

Nếu $yCb \gg CtbLog2SizeY$ bằng $yColBr \gg CtbLog2SizeY$, $yColBr$ nhỏ hơn $SubPicBottomBorderInPic$ và $xColBr$ nhỏ hơn $SubPicRightBorderInPic$, thì phần sau đây áp dụng. Biến $colCb$ xác định khối mã hóa độ chói bao trùm vị trí được cải biến được đưa ra bởi $((xColBr \gg 3) \ll 3, (yColBr \gg 3) \ll 3)$ bên trong ảnh được sắp xếp được quy định bởi $ColPic$. Vị trí độ chói $(xColCb, yColCb)$ được thiết đặt bằng mẫu trên cùng bên trái của khối mã hóa độ chói được sắp xếp được quy định bởi $colCb$ liên quan đến mẫu độ chói trên cùng bên trái của ảnh được sắp xếp được quy định bởi $ColPic$. Quy trình dẫn ra cho các vectơ chuyển động được sắp xếp được truy xuất với $currCb$, $colCb$, $(xColCb, yColCb)$, $refIdxLX$ và $sbFlag$ được thiết đặt bằng không là các đầu vào, và đầu ra được gán tới $mvLXCol$ và $availableFlagLXCol$. Theo cách khác, cả hai bộ phận của $mvLXCol$ được thiết đặt bằng 0 và $availableFlagLXCol$ được thiết đặt bằng không. Thay thế tất cả các lần xuất hiện của $pic_width_in_luma_samples$ với $PicWidthInLumaSamples$. Thay thế tất cả các lần xuất hiện của $pic_height_in_luma_samples$ với $PicHeightInLumaSamples$.

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

<code>seq_parameter_set_rbsp() {</code>	Ký hiệu
<code>sps_seq_parameter_set_id</code>	$ue(v)$
<code>pic_width_in_luma_samples</code>	$ue(v)$
<code>pic_height_in_luma_samples</code>	$ue(v)$
<code>num_subpic_minus1</code>	$ue(v)$
<code>subpic_id_len_minus1</code>	$ue(v)$
<code>for (i = 0; i <= num_subpic_minus1; i++) {</code>	
<code>subpic_id[i]</code>	$u(v)$

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 xác định số lượng các bit được sử dụng để biểu diễn thành phần cú pháp subpic_id[i] trong SPS, sps_subpic_id trong SPPS đề cập đến SPS, và tile_group_subpic_id trong các đoạn đầu nhóm ô đề cập đến SPS. Trị số của subpic_id_len_minus1 sẽ trong khoảng từ $\text{Ceil}(\text{Log}_2(\text{num_subpic_minus1} + 3))$ đến tám. Đây là yêu cầu của sự phù hợp dòng bit rằng sẽ không có sự chồng lấn trong số sub-picture[i] cho i từ 0 đến num_subpic_minus1. Mỗi ảnh con có thể là ảnh con bị ràng buộc chuyển động theo thời gian.

Ngữ nghĩa đoạn đầu nhóm ô chung như sau. Tile_group_subpic_id nhận dạng ảnh con mà nhóm ô thuộc về. Độ dài của tile_group_subpic_id là subpic_id_len_minus1 + 1 bit. Tile_group_subpic_id bằng một chỉ báo nhóm ô không thuộc về ảnh con bất kỳ.

Theo phương án ví dụ thứ ba, cú pháp và ngữ nghĩa đoạn đầu đơn vị NAL như sau.

nal_unit_header() {	Ký hiệu
forbidden_zero_bit	f(1)
nal_unit_type	u(5)
nuh_temporal_id_plus1	u(3)
nuh_subpicture_id_len	u(4)
nuh_reserved_zero_4bits	u(3)
}	

Nuh_subpicture_id_len xác định số lượng các bit được sử dụng để biểu diễn thành phần cú pháp xác định ID ảnh con. Khi trị số của nuh_subpicture_id_len lớn hơn không, các bit nuh_subpicture_id_len-th thứ nhất trong nuh_reserved_zero_4bits sau đó xác định ID của ảnh con mà tải trọng của đơn vị NAL thuộc về. Khi nuh_subpicture_id_len lớn hơn không, trị số của nuh_subpicture_id_len sẽ bằng trị số của subpic_id_len_minus1 trong SPS hoạt động. Trị số của nuh_subpicture_id_len đối với các đơn vị NAL không VCL bị ràng buộc như sau. Nếu nal_unit_type bằng SPS_NUT hoặc PPS_NUT, thì nuh_subpicture_id_len sẽ bằng không. Nuh_reserved_zero_3bits sẽ bằng '000'. Các bộ giải mã sẽ bỏ qua (ví dụ, loại bỏ từ dòng bit và hủy bỏ) các đơn vị NAL với các trị số của nuh_reserved_zero_3bits không bằng '000'.

Theo phương án ví dụ thứ tư, cú pháp lồng ảnh con như sau.

sub-picture_nesting(payloadSize) {	Ký hiệu
all_sub_pictures_flag	u(1)

if(!all_sub_pictures_flag) {	
nesting_num_sub_pictures_minus1	ue(v)
for(i = 0; i <= nesting_num_sub_pictures_minus1; i++)	
nesting_sub_picture_id[i]	u(v)
}	
while(!byte_aligned())	
sub_picture_nesting_zero_bit /* equal to 0 */	u(1)
do	
sei_message()	
while(more_rbsp_data())	
}	

All_sub_pictures_flag bằng một xác định rằng các bản tin SEI được lồng áp dụng tới tất cả các ảnh con. All_sub_pictures_flag bằng một xác định rằng các ảnh con trong đó các bản tin SEI được lồng áp dụng được truyền tín hiệu một cách tường minh bởi các thành phần cú pháp tiếp theo. Nesting_num_sub_pictures_minus1 cộng 1 xác định số lượng các ảnh con trong đó các bản tin SEI được lồng áp dụng. Nesting_sub_picture_id[i] chỉ báo ID ảnh con của ảnh con thứ i trong đó các bản tin SEI được lồng áp dụng. Thành phần cú pháp nesting_sub_picture_id[i] được biểu diễn bởi $\text{Ceil}(\text{Log}_2(\text{nesting_num_sub_pictures_minus1} + 1))$ bit. Sub_picture_nesting_zero_bit sẽ bằng không.

Fig.9 là hình vẽ giản lược của thiết bị mã video 900 ví dụ. Thiết bị mã video 900 thích hợp cho việc thực hiện các ví dụ/các phương án được bộc lộ như

được mô tả ở đây. Thiết bị mã video 900 bao gồm các cổng dòng xuống 920, các cổng dòng lên 950, và/hoặc các bộ phận thu phát (Tx/Rx) 910, bao gồm các bộ truyền và/hoặc các bộ thu để truyền thông dòng lên và/hoặc dòng xuống dữ liệu qua mạng. Thiết bị mã video 900 cũng bao gồm bộ xử lý 930 bao gồm bộ phận logic và/hoặc bộ xử lý trung tâm (central processing unit - CPU) để xử lý dữ liệu và bộ nhớ 932 để lưu trữ dữ liệu. Thiết bị mã video 900 cũng có thể bao gồm các bộ phận điện, quang-điện (optical-to-electrical - OE), các bộ phận điện-quang (electrical-to-optical - EO), và/hoặc các bộ phận truyền thông không dây được ghép nối với các cổng dòng lên 950 và/hoặc các cổng dòng xuống 920 cho việc truyền thông của dữ liệu qua các mạng truyền thông điện, quang, hoặc không dây. Thiết bị mã video 900 cũng có thể bao gồm các thiết bị đầu vào và/hoặc đầu ra (input and/or output - I/O) 960 để truyền thông dữ liệu tới và từ người dùng. Các thiết bị I/O 960 có thể bao gồm các thiết bị đầu ra chẳng hạn như màn hình cho việc hiển thị dữ liệu video, các loa để đưa ra dữ liệu âm thanh, 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 bởi phần cứng và phần mềm. Bộ xử lý 930 có thể được thực hiện như một hoặc nhiều chip CPU, nhiều lõi (ví dụ, như bộ xử lý nhiều lõi), các mảng cổng lập trình được dạng trường (field-programmable gate array - FPGA), các mạch tích hợp chuyên dụng (application specific integrated circuit - ASIC), và các bộ xử lý tín hiệu kỹ thuật số (digital signal processor - DSP). Bộ xử lý 930 truyền thông với các cổng dòng xuống 920, Tx/Rx 910, các cổng dòng lên 950, và bộ nhớ 932. Bộ xử lý 930 bao gồm môđun mã hóa 914. Môđun mã hóa 914 thực hiện các phương án được bộc lộ nêu trên, chẳng hạn như các phương pháp 100, 1000, 1100, và/hoặc cơ chế 700, mà có thể ứng dụng dòng bit 500, ảnh 600, và/hoặc ảnh 800. Môđun mã 914 cũng có thể thực hiện phương pháp/cơ chế khác bất kỳ được mô tả ở đây. Hơn nữa, môđun mã 914 có thể thực hiện hệ thống bộ mã hóa/giải mã 200, bộ mã hóa 300, và/hoặc bộ giải mã 400. Ví dụ, môđun mã 914 có thể được ứng dụng để truyền tín hiệu và/hoặc thu nhận các kích thước và các vị trí ảnh con trong SPS. Theo ví dụ khác, môđun mã 914 có thể ràng buộc các độ rộng ảnh con và các độ cao ảnh con là các bội số của kích thước CTU trừ khi các ảnh

con như vậy lần lượt được bố trí ở ranh giới bên phải của ảnh hoặc ranh giới dưới cùng của ảnh. Theo ví dụ khác, môđun mã 914 có thể ràng buộc các ảnh con để bao trùm ảnh mà không có khe hở hoặc chồng lấn. Theo ví dụ khác, môđun mã 914 có thể được ứng dụng để truyền tín hiệu và/hoặc thu nhận dữ liệu chỉ báo một số ảnh con là các ảnh con bị ràng buộc chuyển động theo thời gian và các ảnh con khác không. Theo ví dụ khác, môđun mã 914 có thể truyền tín hiệu tập hợp của các ID ảnh con hoàn thiện trong SPS và bao gồm ID ảnh con ở mỗi đoạn đầu lát để chỉ báo ảnh con mà chứa các lát tương ứng. Theo ví dụ khác, môđun mã 914 có thể truyền tín hiệu các mức cho mỗi ảnh con. Như vậy, môđun mã 914 khiến thiết bị mã video 900 cung cấp tính năng bổ sung, tránh khỏi việc xử lý nhất định để làm giảm phí tổn xử lý, và/hoặc làm tăng hiệu quả mã hóa khi phân chia và mã hóa dữ liệu video. Theo đó, môđun mã 914 nâng cao tính năng của thiết bị mã video 900 cũng như xử lý các vấn đề mà cụ thể với các lĩnh vực mã video. Hơn nữa, môđun mã 914 ảnh hưởng phép biến đổi của thiết bị mã video 900 tới trạng thái khác. Theo cách khác, môđun mã 914 có thể được thực hiện như các lệnh được lưu trữ trong bộ nhớ 932 và được thực hiện bởi bộ xử lý 930 (ví dụ, như sản phẩm chương trình máy tính được lưu trữ trên phương tiện bất biến).

Bộ nhớ 932 bao gồm một hoặc nhiều loại bộ nhớ chẳng hạn như các đĩa, các ổ đĩa băng, các ổ đĩa bán dẫn, bộ nhớ chỉ đọc (read only memory - ROM), bộ nhớ truy cập ngẫu nhiên (random access memory - RAM), bộ nhớ tia chớp, bộ nhớ nội dung địa chỉ 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 như thiết bị lưu trữ dữ liệu tràn, để lưu trữ các chương trình khi các chương trình như vậy được lựa chọn cho việc thực hiện, và để lưu trữ các lệnh và dữ liệu mà được đọc trong suốt thời gian thực hiện chương trình.

Fig.10 là lưu đồ của phương pháp 1000 ví dụ về việc mã hóa dòng bit, chẳng hạn như dòng bit 500 và/hoặc dòng bit con 501, của các ảnh con, chẳng hạn như các ảnh con 522, 523, 622, 722, và/hoặc 822, với các sự ràng buộc vị trí. Phương pháp 1000 có thể được ứng dụng bởi bộ mã hóa, chẳng hạn như hệ thống bộ mã hóa/giải mã 200, bộ mã hóa 300, và/hoặc thiết bị 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 các ảnh và xác định mã hóa chuỗi video đó thành dòng bit, ví dụ dựa vào đầu vào người dùng. Chuỗi video được phân chia thành các ảnh/các hình ảnh/các khung để phân chia thêm trước khi mã hóa. Ở bước 1001, ảnh được phân chia thành các ảnh con theo các sự ràng buộc vị trí. Cụ thể là, sự liên kết của các ảnh con kết quả bao trùm tổng diện tích của ảnh mà không có khe hở và/hoặc không có sự chồng lấn. Theo một số ví dụ, sự ràng buộc kích thước thích nghi cũng được áp dụng. Ví dụ, khi ảnh con hiện thời bao gồm ranh giới bên phải mà không trùng khớp với ranh giới bên phải của ảnh, độ rộng ảnh con của ảnh con hiện thời bị ràng buộc là bội số nguyên của CTU (ví dụ, các ảnh con 822 khác ngoài 822b). Theo đó, ít nhất một trong số các ảnh con (ví dụ, các ảnh con 822b) có thể bao gồm độ rộng ảnh con mà không là bội số nguyên của kích thước CTU khi mỗi ảnh con bao gồm ranh giới bên phải mà trùng khớp với ranh giới bên phải của ảnh. Hơn nữa, khi ảnh con hiện thời bao gồm ranh giới dưới cùng mà không trùng khớp với ranh giới dưới cùng của ảnh, độ cao ảnh con của ảnh con hiện thời bị ràng buộc là bội số nguyên của CTU (ví dụ, các ảnh con 822 khác ngoài 822c). Theo đó, ít nhất một trong số các ảnh con (ví dụ, các ảnh con 822c) có thể bao gồm độ cao ảnh con mà không là bội số nguyên của kích thước CTU khi mỗi ảnh con bao gồm ranh giới dưới cùng mà trùng khớp với ranh giới dưới cùng của ảnh.

Ở bước 1003, SPS được mã hóa thành dòng bit. SPS có thể bao gồm kích thước ảnh con cho các ảnh con và vị trí ảnh con cho mỗi trong số các ảnh con. SPS cũng có thể bao gồm các ID ảnh con cho mỗi trong số các ảnh con. SPS cũng có thể bao gồm sự chỉ báo rằng một hoặc nhiều trong số các ảnh con là các ảnh con bị ràng buộc chuyển động.

Ở bước 1005, một hoặc nhiều trong số các ảnh con được mã hóa thành dòng bit. Dòng bit được lưu trữ cho việc truyền thông về phía bộ giải mã ở bước 1007. Dòng bit sau đó có thể được truyền về phía bộ giải mã như được mong muốn. Theo một số ví dụ, dòng bit con có thể được trích từ dòng bit được mã hóa. Trong trường hợp như vậy, dòng bit được truyền là dòng bit con. Theo các ví dụ khác, dòng bit được mã hóa có thể được truyền cho việc trích dòng bit con ở bộ giải mã. Theo các ví dụ khác nữa, dòng bit được mã hóa có thể được giải mã và được hiển

thị mà không trích dòng bit con. Theo ví dụ bất kỳ trong số các ví dụ, các sự ràng buộc vị trí đảm bảo rằng các khe hở và/hoặc các sự chồng lấn ảnh con không được phép. Bằng cách loại bỏ các tùy chọn mã hóa này độ phức tạp của quy trình RDO được giảm. Hơn nữa, độ phức tạp giải mã cũng được giảm. Kết quả là, việc tránh khỏi các khe hở và các sự chồng lấn giữa các ảnh con có thể làm giảm việc sử dụng 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 1100 ví dụ về việc giải mã dòng bit, chẳng hạn như dòng bit 500 và/hoặc dòng bit con 501, của các ảnh con, chẳng hạn như các ảnh con 522, 523, 622, 722, và/hoặc 822, với các sự ràng buộc vị trí. Phương pháp 1100 có thể được ứng dụng bởi bộ giải mã, chẳng hạn như hệ thống bộ mã hóa/giải mã 200, bộ giải mã 400, và/hoặc thiết bị mã video 900 khi thực hiện phương pháp 100. Ví dụ, phương pháp 1100 có thể được áp dụng để giải mã dòng bit được tạo từ phương pháp 1000.

Phương pháp 1100 có thể bắt đầu khi bộ giải mã bắt đầu thu dòng bit chứa các ảnh con. Dòng bit có thể bao gồm chuỗi video hoàn thiện hoặc dòng bit có thể là dòng bit con chứa tập hợp được giảm của các ảnh con cho việc trích riêng biệt. Ở bước 1101, dòng bit được thu. Dòng bit bao gồm các ảnh con được phân chia từ ảnh theo các sự ràng buộc vị trí. Cụ thể là, sự liên kết của các ảnh con kết quả bao trùm tổng diện tích của ảnh mà không có khe hở và/hoặc không có sự chồng lấn. Theo một số ví dụ, sự ràng buộc kích thước thích nghi cũng được áp dụng. Ví dụ, khi ảnh con hiện thời bao gồm ranh giới bên phải mà không trùng khớp với ranh giới bên phải của ảnh, độ rộng ảnh con của ảnh con hiện thời bị ràng buộc là bội số nguyên của CTU (ví dụ, các ảnh con 822 khác ngoài 822b). Theo đó, ít nhất một trong số các ảnh con (ví dụ, các ảnh con 822b) có thể bao gồm độ rộng ảnh con mà không là bội số nguyên của kích thước CTU khi mỗi ảnh con bao gồm ranh giới bên phải mà trùng khớp với ranh giới bên phải của ảnh. Hơn nữa, khi ảnh con hiện thời bao gồm ranh giới dưới cùng mà không trùng khớp với ranh giới dưới cùng của ảnh, độ cao ảnh con của ảnh con hiện thời bị ràng buộc là bội số nguyên của CTU (ví dụ, các ảnh con 822 khác ngoài 822c). Theo đó, ít nhất một trong số các ảnh con (ví dụ, các ảnh con 822c) có thể bao gồm độ cao ảnh con mà

không là bội số nguyên của kích thước CTU khi mỗi ảnh con bao gồm ranh giới dưới cùng mà trùng khớp với ranh giới dưới cùng của ảnh.

Ở bước 1103, SPS ở dòng bit được phân tách để thu nhận kích thước ảnh con cho các ảnh con, vị trí ảnh con cho mỗi trong số các ảnh con, các ID ảnh con cho mỗi trong số các ảnh con, và/hoặc chỉ báo rằng một hoặc nhiều trong số các ảnh con là các ảnh con bị ràng buộc chuyển động.

Ở bước 1105, dòng bit được phân tách để thu nhận một hoặc nhiều ảnh con. Ở bước 1107, một hoặc nhiều ảnh con được giải mã để tạo chuỗi video. Chuỗi video sau đó có thể được chuyển tiếp cho việc hiển thị. Các sự ràng buộc vị trí đảm bảo rằng các khe hở và/hoặc các sự chồng lấn ảnh con không được phép. Bằng cách loại bỏ các tùy chọn mã hóa này độ phức tạp của quy trình giải mã được giảm. Kết quả là, việc tránh khỏi các khe hở và các sự chồng lấn giữa các ảnh con có thể làm giảm việc sử dụng các tài nguyên bộ nhớ và/hoặc các tài nguyên xử lý ở bộ giải mã.

Fig.12 là hình vẽ giản lược của hệ thống 1200 ví dụ về việc truyền tín hiệu dòng bit, chẳng hạn như dòng bit 500 và/hoặc dòng bit con 501, của các ảnh con, chẳng hạn như các ảnh con 522, 523, 622, 722, và/hoặc 822, với các sự ràng buộc vị trí. Hệ thống 1200 có thể được thực hiện bởi bộ mã hóa và bộ giải mã chẳng hạn như hệ thống bộ mã hóa/giải mã 200, bộ mã hóa 300, bộ giải mã 400, và/hoặc thiết bị mã video 900. Hơn nữa, hệ thống 1200 có thể được ứng 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 các ảnh con sao cho sự liên kết của các ảnh con bao trùm tổng diện tích của ảnh mà không chồng lấn. Bộ mã hóa video 1202 còn bao gồm môđun mã hóa 1203 để mã hóa một hoặc nhiều trong số các ảnh con thành dòng bit. Bộ mã hóa video 1202 còn bao gồm môđun lưu trữ 1205 để lưu trữ dòng bit cho việc truyền thông về phía bộ giải mã. Bộ mã hóa video 1202 còn bao gồm môđun truyền 1207 để truyền dòng bit bao gồm các ảnh con về phía bộ giải mã. Bộ mã hóa video 1202 còn có thể 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 dòng bit bao gồm các ảnh con được phân chia từ ảnh sao cho sự liên kết của các ảnh con bao trùm tổng diện tích của ảnh mà không chồng lấn. Bộ giải mã video 1210 còn bao gồm môđun phân tách 1213 để phân tách dòng bit để thu nhận một hoặc nhiều ảnh con. Bộ giải mã video 1210 còn bao gồm môđun giải mã 1215 để giải mã một hoặc nhiều ảnh con để tạo chuỗi video. Bộ giải mã video 1210 còn bao gồm môđun chuyển tiếp 1217 để chuyển tiếp chuỗi video cho việc hiển thị. Bộ giải mã video 1210 còn có thể được tạo cấu hình để thực hiện bước bất kỳ trong số các bước của phương pháp 1100.

Bộ phận thứ nhất được ghép nối trực tiếp với bộ phận thứ hai khi không có các bộ phận xen kẽ, ngoại trừ đường dẫn, đường quét, hoặc phương tiện khác giữa bộ phận thứ nhất và bộ phận thứ hai. Bộ phận thứ nhất được ghép nối gián tiếp với bộ phận thứ hai khi có các bộ phận xen kẽ khác ngoài đường dẫn, đường quét, hoặc phương tiện khác giữa bộ phận thứ nhất và bộ phận thứ hai. Thuật ngữ “được ghép nối” và các biến thể của nó bao gồm cả được ghép nối trực tiếp và được ghép nối gián tiếp. Việc sử dụng thuật ngữ “khoảng” nghĩa là khoảng bao gồm $\pm 10\%$ của số tiếp theo nếu không được quy định khác.

Cần hiểu rằng các bước theo các phương pháp làm ví dụ được đưa ra ở đây không cần được yêu cầu được thực hiện theo thứ tự được mô tả, và thứ tự của các bước theo các phương pháp như vậy nên được hiểu chỉ là làm ví dụ. Tương tự như vậy, các bước bổ sung có thể được bao gồm trong các phương pháp như vậy, và các bước nhất định có thể được bỏ qua hoặc được kết hợp, theo các phương pháp phù hợp với các phương án khác nhau của sáng chế.

Trong khi nhiều phương án đã được đề xuất theo sáng chế, có thể được hiểu rằng các hệ thống và các phương pháp được bộc lộ có thể được bao gồm dưới dạng cụ thể khác mà không chệch khỏi bản chất hoặc phạm vi của sáng chế. Các ví dụ hiện tại được coi là minh họa và không giới hạn, và dự định không giới hạn ở các chi tiết được đưa ra ở đây. Ví dụ, các thành phần hoặc các bộ phận khác nhau có thể được kết hợp hoặc được tích hợp vào 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 được mô tả và được minh họa theo các phương án khác nhau là rời rạc hoặc tách biệt có thể được kết hợp hoặc được tích hợp với các hệ thống, các bộ phận, các kỹ thuật, hoặc các phương pháp khác mà không chệch khỏi phạm vi của sáng chế. Các ví dụ khác về các sự thay đổi, các sự thay thế, và các sự biến đổi chắc chắn có thể hiểu được bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật tương ứng và có thể được thực hiện mà không chệch khỏi bản chất và phạm vi được bộc lộ ở đây.

YÊU CẦU BẢO HỘ

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

thu (1101), bởi bộ thu của bộ giải mã, dòng bit mà bao gồm dữ liệu được mã hóa của các ảnh con được phân từ ảnh sao cho sự liên kết của các ảnh con bao trùm tổng diện tích của ảnh mà không chồng lấn; trong đó dòng bit bao gồm tập hợp thông số chuỗi (sequence parameter set - SPS), SPS bao gồm tập hợp hoàn chỉnh của các ký hiệu nhận dạng (ID) ảnh con, đối với các ảnh con, trong đó mỗi ảnh con tương ứng với ID ảnh con, và SPS còn bao gồm kích cỡ ảnh con và vị trí ảnh con đối với mỗi ảnh con;

phân tách (1103), bởi bộ xử lý của bộ giải mã, SPS để thu nhận ID ảnh con, kích cỡ ảnh con và vị trí ảnh con đối với mỗi ảnh con;

phân tách (1105), bởi bộ xử lý của bộ giải mã, dòng bit để thu nhận dữ liệu được mã hóa của một hoặc nhiều ảnh con;

giải mã (1107), bởi bộ xử lý, dữ liệu được mã hóa của một hoặc nhiều ảnh con để tạo ra chuỗi video.

2. Phương pháp theo điểm 1, trong đó sự liên kết của các ảnh con còn bao trùm tổng diện tích của ảnh mà không có khoảng trống.

3. Phương pháp theo điểm 1 hoặc 2, trong đó nếu ảnh con thứ nhất bao gồm ranh giới bên phải mà không trùng khớp với ranh giới bên phải của ảnh, ảnh con thứ nhất bao gồm độ rộng ảnh con mà là bội nguyên của kích cỡ đơn vị cây mã hóa (coding tree unit - CTU).

4. Phương pháp theo điểm bất kỳ trong số các điểm 1 đến 3, trong đó nếu ảnh con thứ hai bao gồm ranh giới dưới cùng mà không trùng khớp với ranh giới dưới cùng của ảnh, ảnh con thứ hai bao gồm độ cao ảnh con mà là bội nguyên của kích cỡ CTU.

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

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

cho sự liên kết của các ảnh con bao trùm tổng diện tích của ảnh mà không chồng lấn;

mã hóa (1003), bởi bộ xử lý, tập hợp thông số chuỗi (sequence parameter set - SPS), thành dòng bit, trong đó SPS bao gồm tập hoàn chỉnh của các ký hiệu nhận dạng (ID) ảnh con, đối với các ảnh con, trong đó mỗi ảnh con tương ứng với ID ảnh con, và SPS còn bao gồm kích cỡ ảnh con và vị trí ảnh con đối với mỗi ảnh con;

mã hóa (1005), bởi bộ xử lý, một hoặc nhiều ảnh con thành dòng bit.

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

lưu trữ, trong bộ nhớ của bộ mã hóa, dòng bit cho việc truyền thông tới bộ giải mã.

7. Phương pháp theo điểm 5 hoặc 6, trong đó sự liên kết của các ảnh con còn bao trùm tổng diện tích của ảnh mà không có khoảng trống.

8. Phương tiện bất biến đọc được bởi máy tính bao gồm sản phẩm chương trình máy tính để sử dụng bởi thiết bị mã video, sản phẩm chương trình máy tính chứa các lệnh có thể được thực thi bởi máy tính được lưu trữ trên phương tiện bất biến đọc được bởi máy tính sao cho khi được thực thi bởi bộ xử lý làm cho thiết bị mã video thực hiện phương pháp theo điểm bất kỳ trong số các điểm 1 đến 7.

9. Bộ giải mã, bao gồm:

bộ thu (1211), có cấu trúc để thu dòng bit mà bao gồm dữ liệu được mã hóa của các ảnh con được phân từ ảnh sao cho sự liên kết của các ảnh con bao trùm tổng diện tích của ảnh mà không chồng lấn; trong đó dòng bit bao gồm tập hợp thông số chuỗi (sequence parameter set - SPS), SPS bao gồm tập hợp hoàn chỉnh của các ký hiệu nhận dạng (ID) ảnh con, đối với các ảnh con, trong đó mỗi ảnh con tương ứng với ID ảnh con, và SPS còn bao gồm kích cỡ ảnh con và vị trí ảnh con đối với mỗi ảnh con;

bộ phân tách (1213), có cấu trúc để phân tách dòng bit để thu nhận dữ liệu được mã hóa của một hoặc nhiều ảnh con;

bộ phân tích (1213), có cấu trúc để phân tích SPS để thu nhận ID ảnh con, kích cỡ ảnh con và vị trí ảnh con, đối với mỗi ảnh con;

bộ giải mã (1215), có cấu trúc để giải mã dữ liệu được mã hóa của một hoặc nhiều ảnh con để tạo ra chuỗi video.

10. Bộ giải mã theo điểm 9, trong đó bộ giải mã còn có cấu trúc để thực hiện phương pháp theo điểm bất kỳ trong số các điểm 1 đến 4.

11. Bộ mã hóa, bao gồm:

bộ phân chia (1201), có cấu trúc để phân chia ảnh thành nhiều ảnh con sao cho sự liên kết của các ảnh con bao trùm tổng diện tích của ảnh mà không chồng lấn;

bộ mã hóa (1203), có cấu trúc để mã hóa một hoặc nhiều ảnh con thành dòng bit;

bộ mã hóa (1203), có cấu trúc để mã hóa tập hợp thông số chuỗi (sequence parameter set - SPS), thành dòng bit, trong đó SPS bao gồm tập hoàn chỉnh của các ký hiệu nhận dạng (ID) ảnh con, đối với các ảnh con, trong đó mỗi ảnh con tương ứng với ID ảnh con, và SPS còn bao gồm kích cỡ ảnh con và vị trí ảnh con đối với mỗi ảnh con.

12. Bộ mã hóa theo điểm 11, bộ mã hóa còn bao gồm:

bộ lưu trữ (1205), có cấu trúc để lưu trữ dòng bit cho việc truyền thông tới bộ giải mã.

13. Bộ mã hóa theo điểm 11 hoặc 12, trong đó bộ mã hóa còn được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm 5 đến 7.

14. Phương tiện bất biến đọc được bởi máy tính chứa dòng bit bao gồm một hoặc nhiều ảnh con được phân chia từ ảnh sao cho sự liên kết của các ảnh con bao trùm tổng diện tích của ảnh mà không chồng lấn; trong đó dòng bit này khác biệt ở chỗ:

dòng bit còn bao gồm tập hợp thông số chuỗi (sequence parameter set - SPS), trong đó SPS bao gồm tập hoàn chỉnh của các ký hiệu nhận dạng (ID) ảnh con, đối với các ảnh con, trong đó mỗi ảnh con tương ứng với ID ảnh con, và SPS còn bao gồm kích cỡ ảnh con và vị trí ảnh con đối với mỗi ảnh con.

1/12

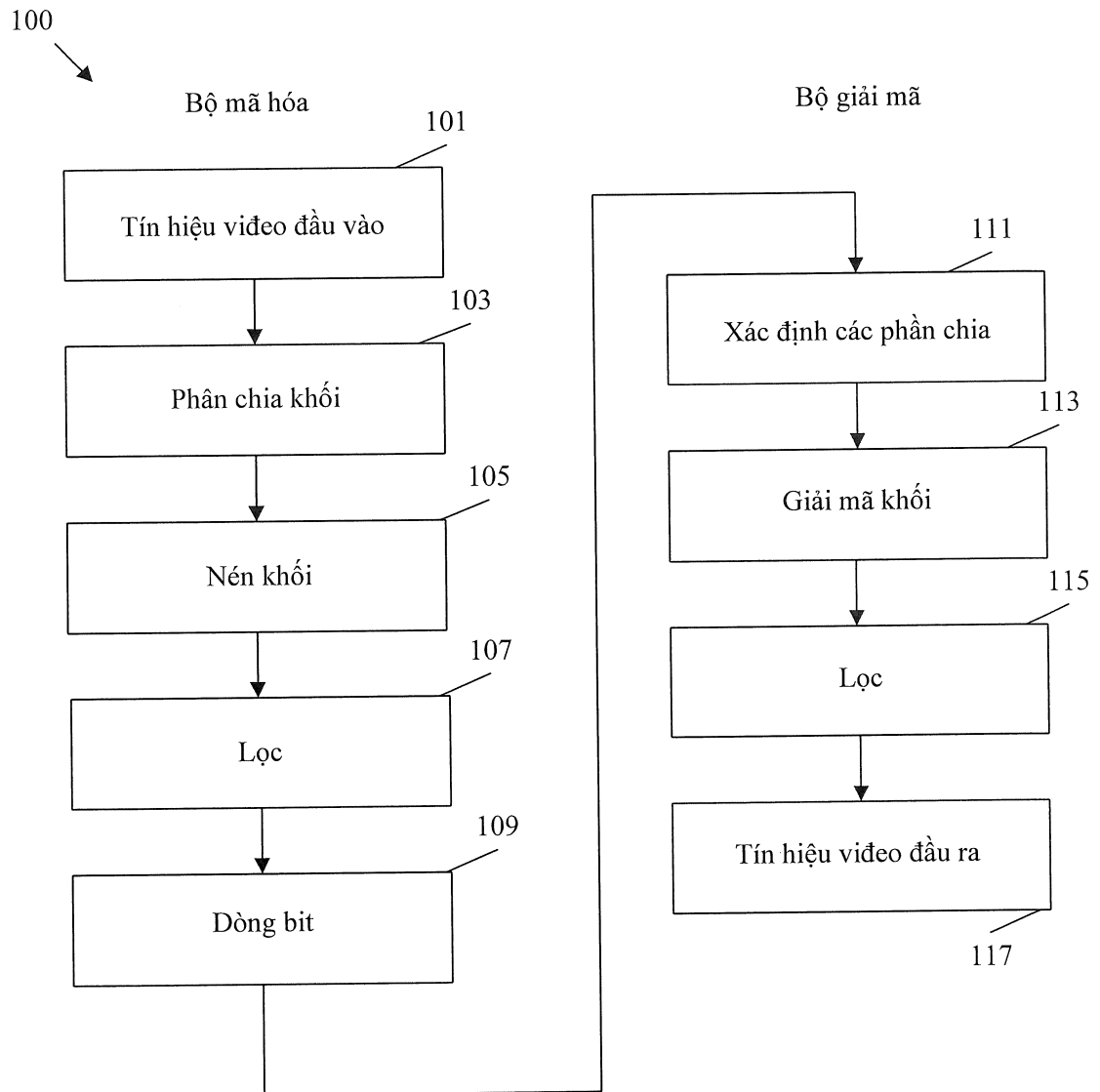


FIG. 1

2/12

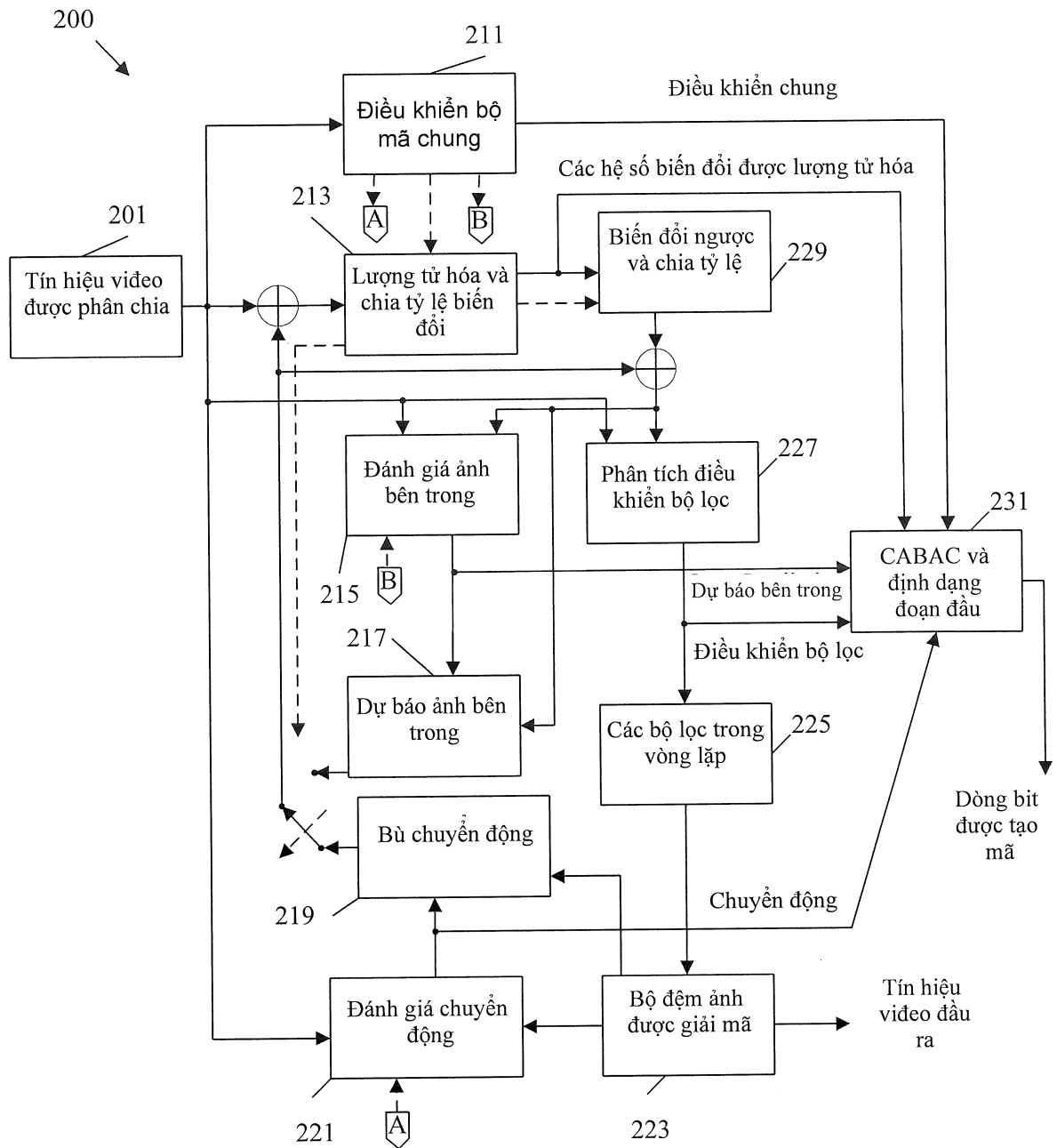


FIG. 2

3/12

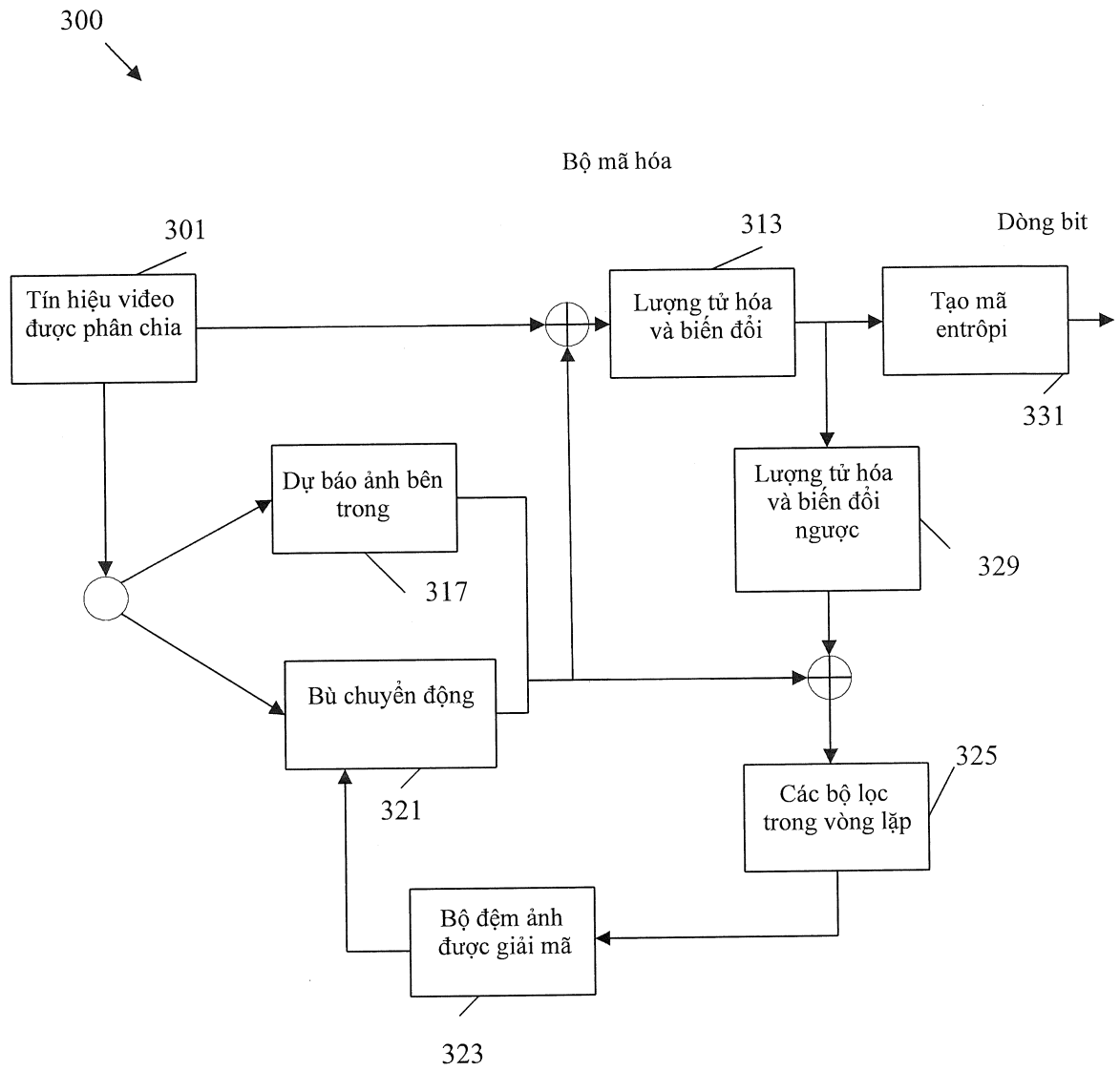


FIG. 3

4/12

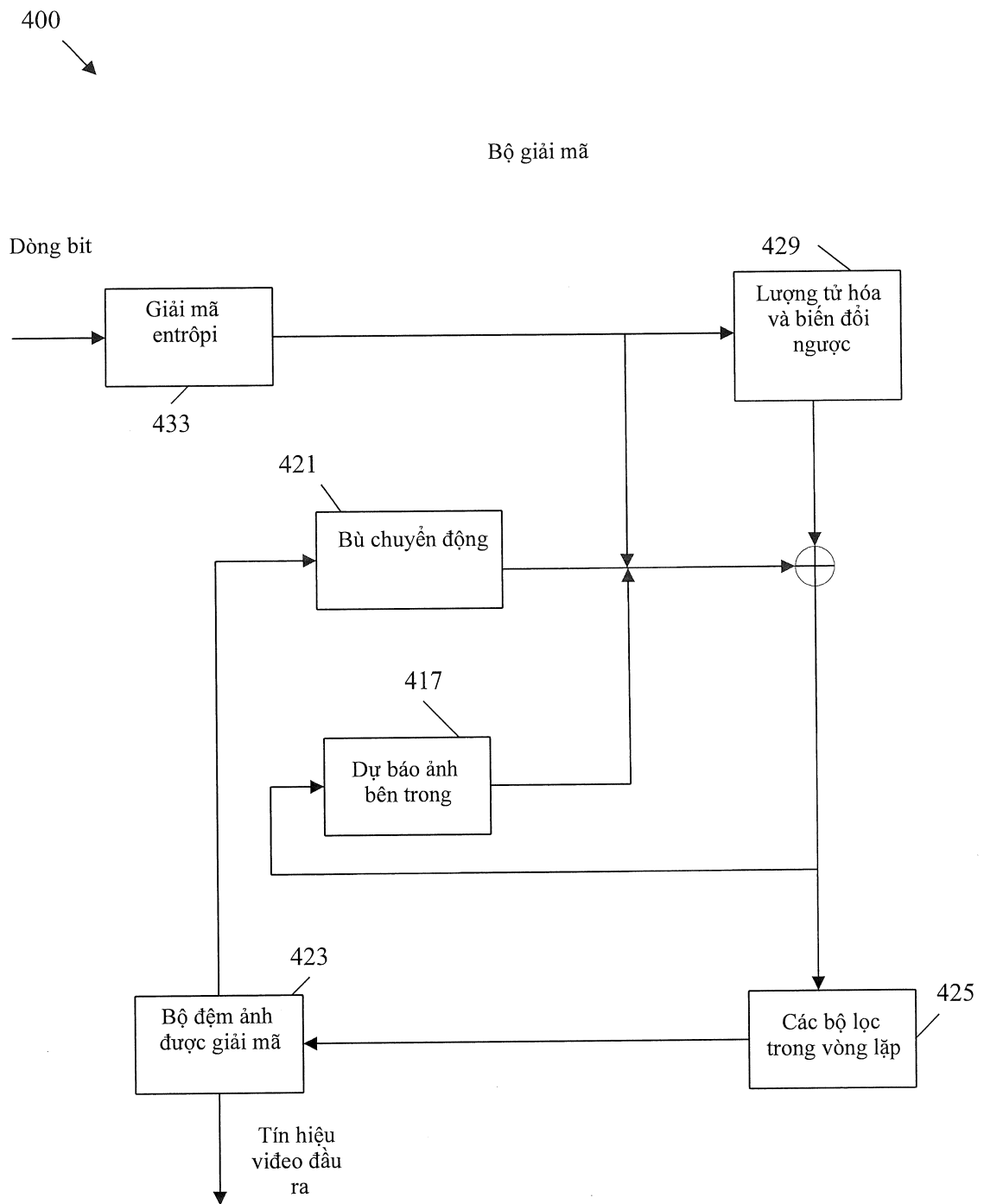


FIG. 4

5/12

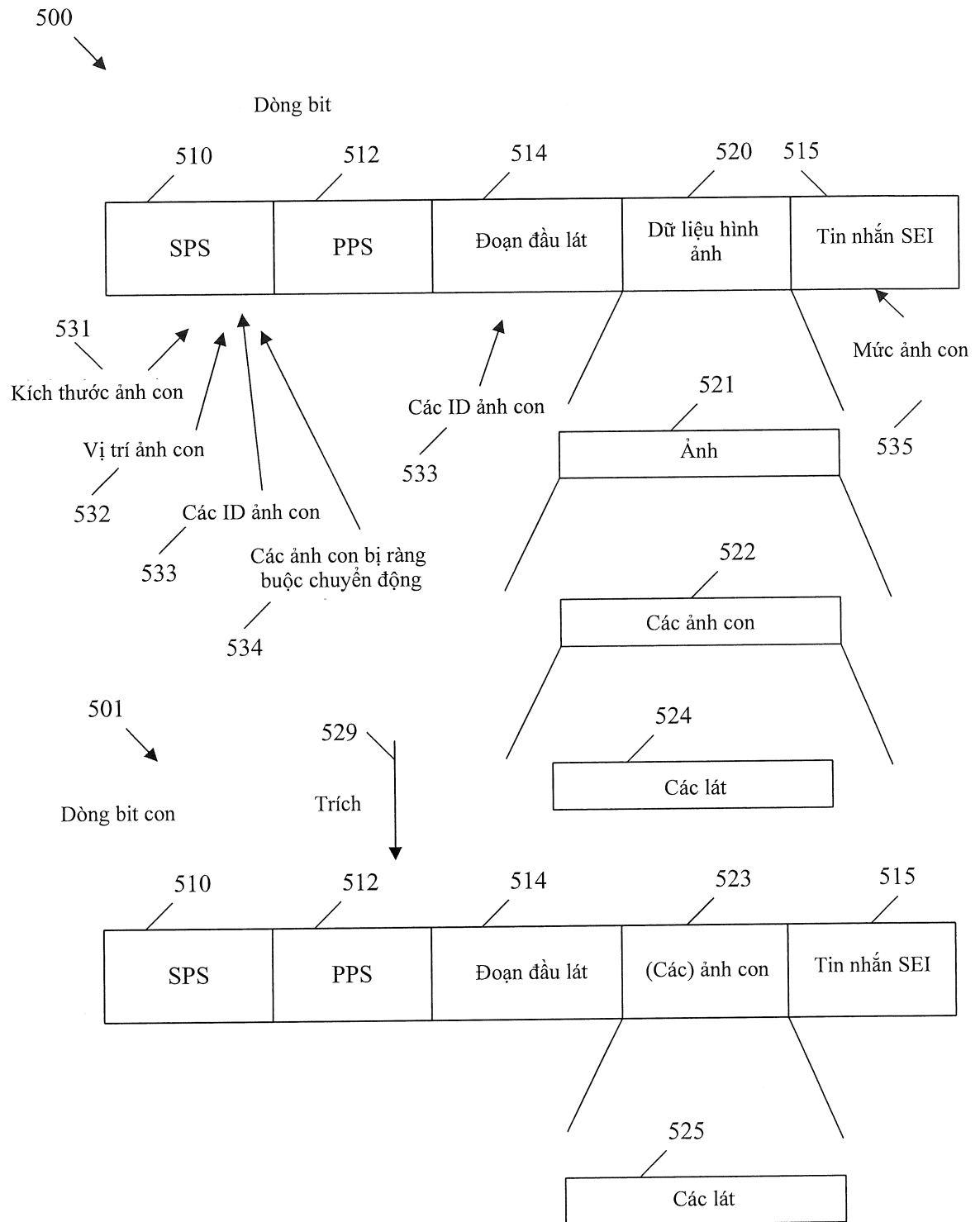


FIG. 5

6/12

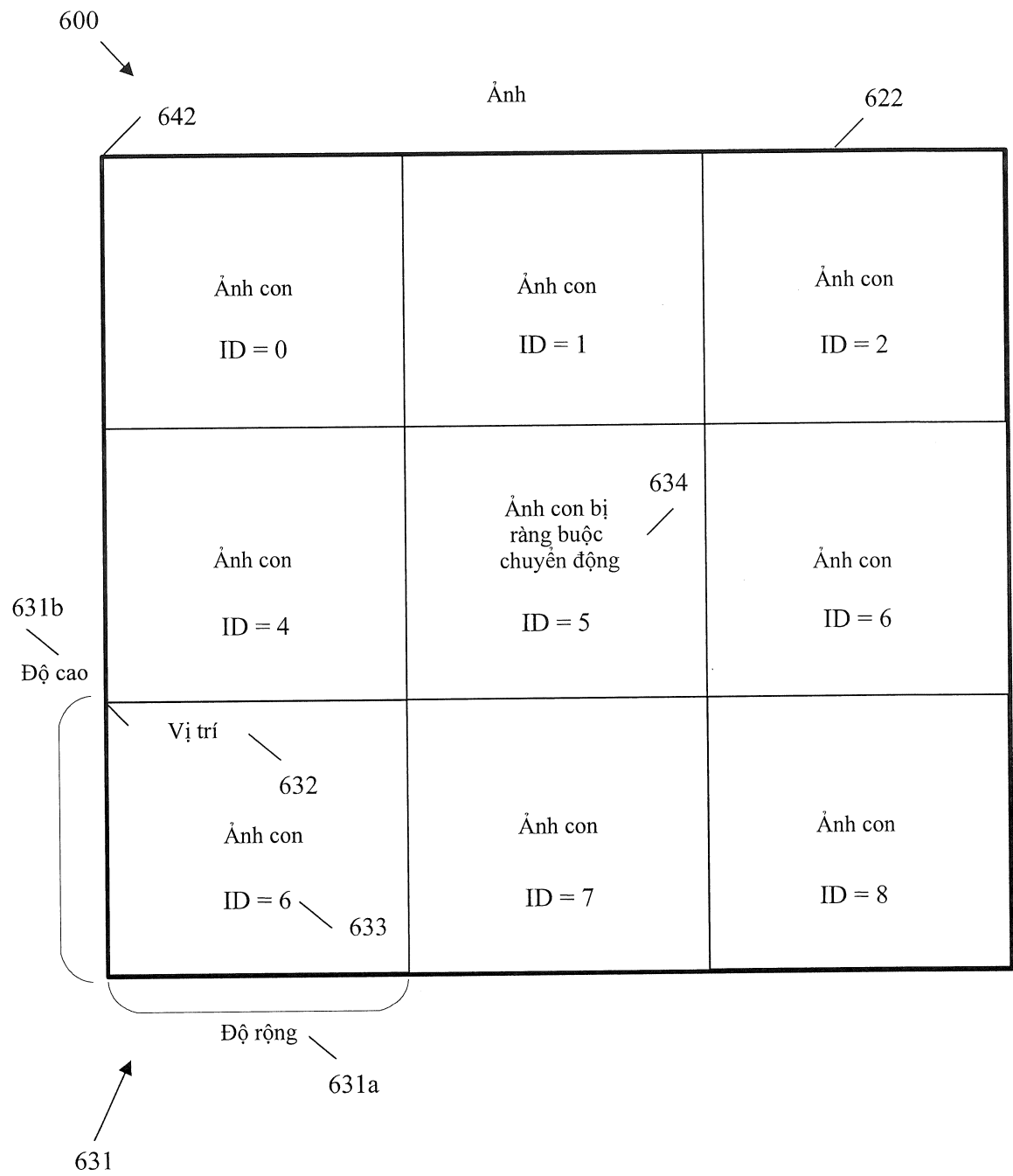


FIG. 6

7/12

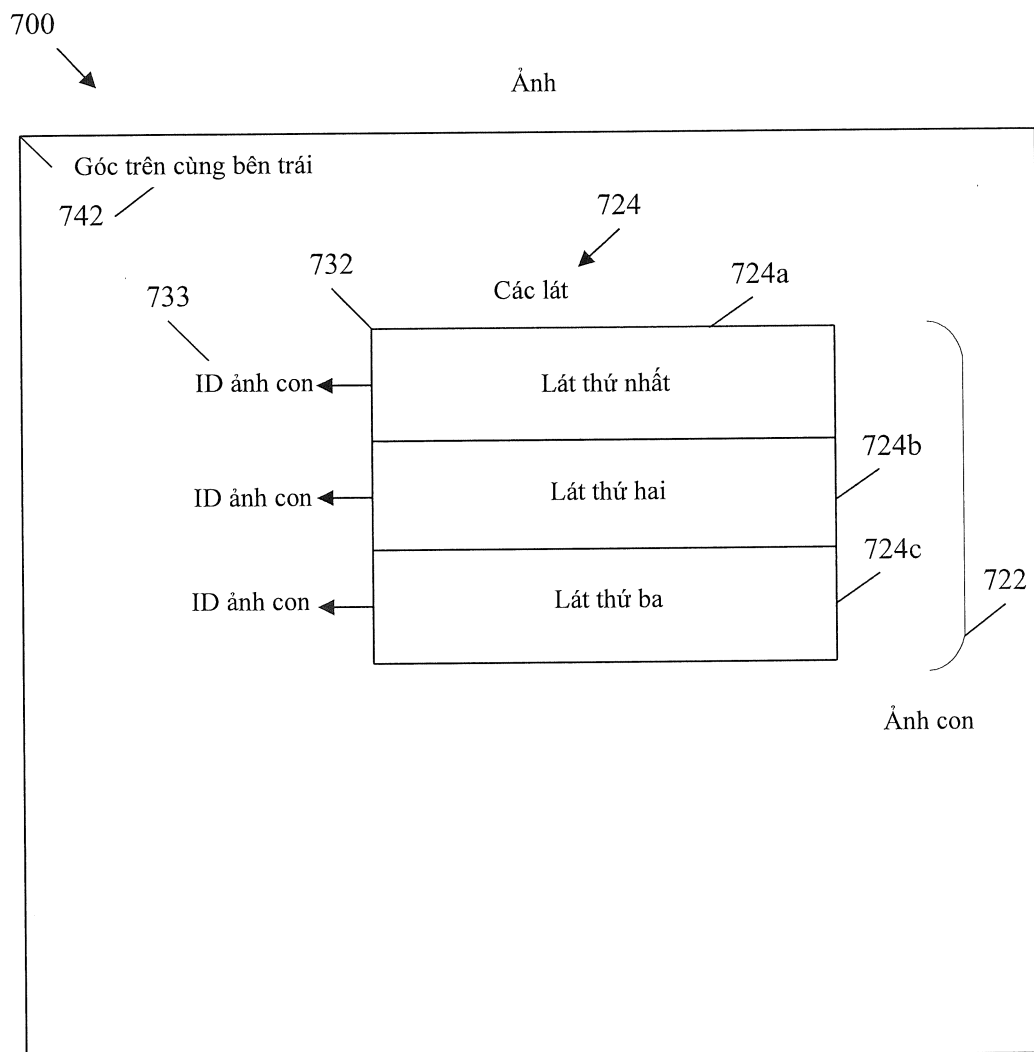


FIG. 7

8/12

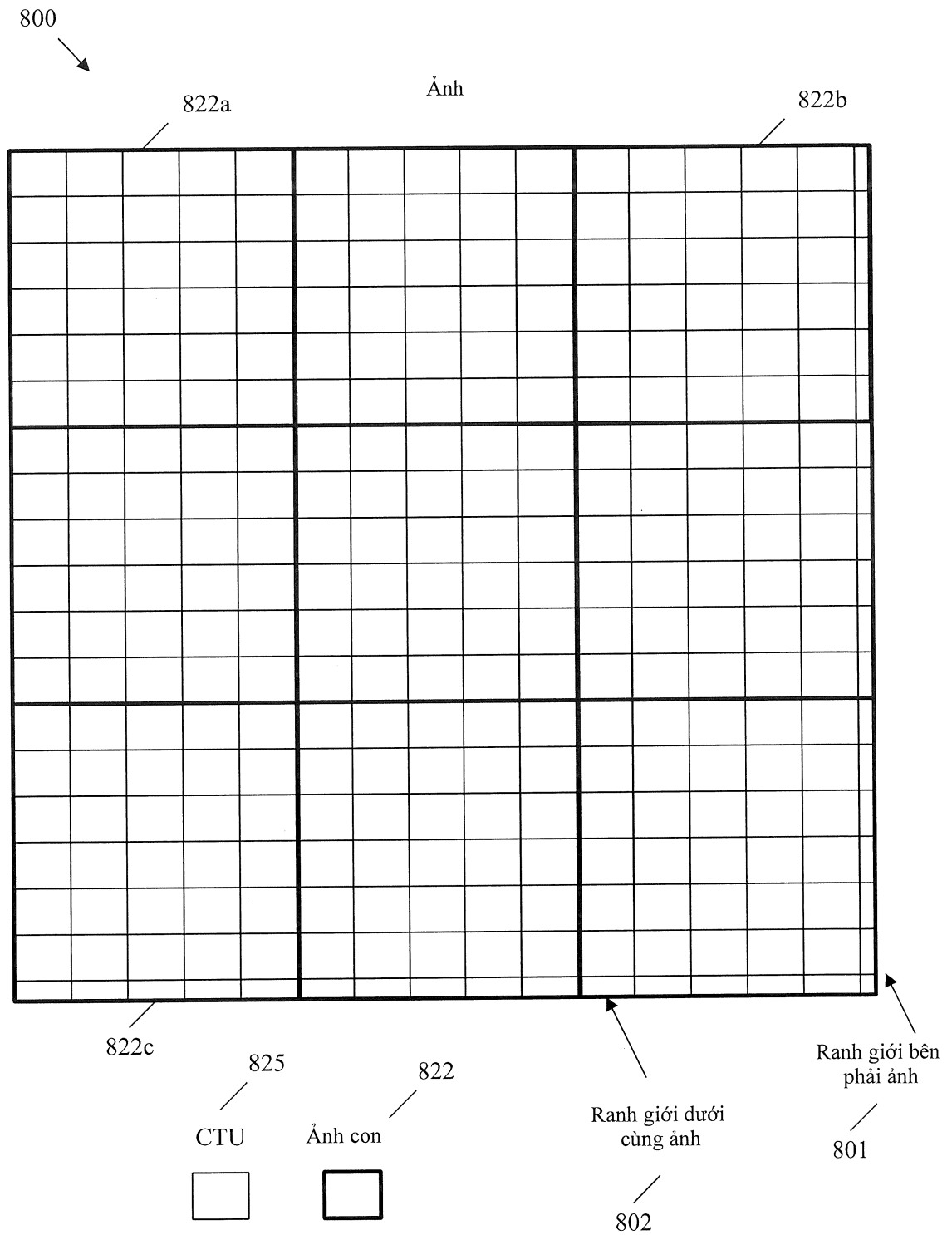


FIG. 8

9/12

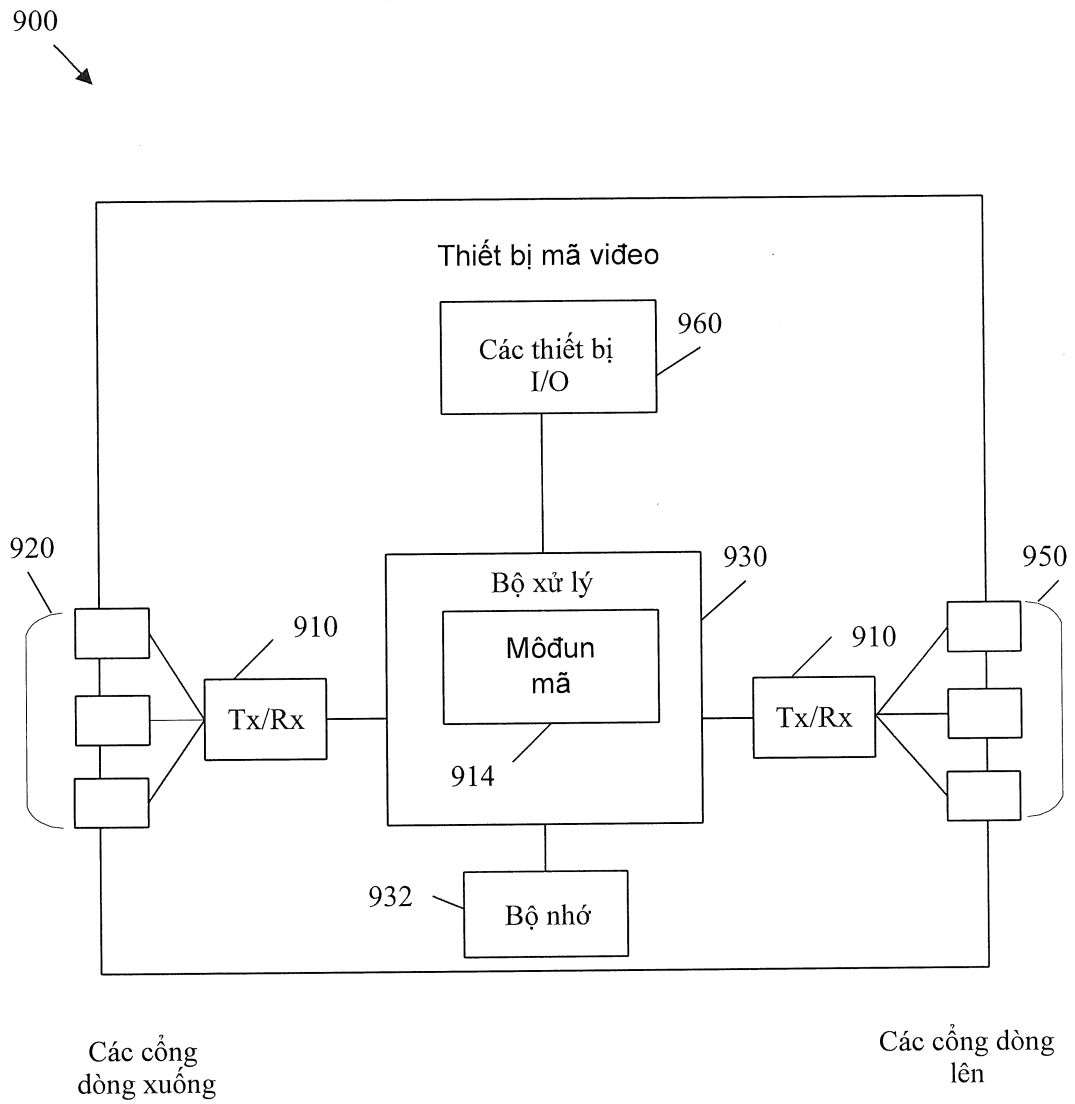


FIG. 9

10/12

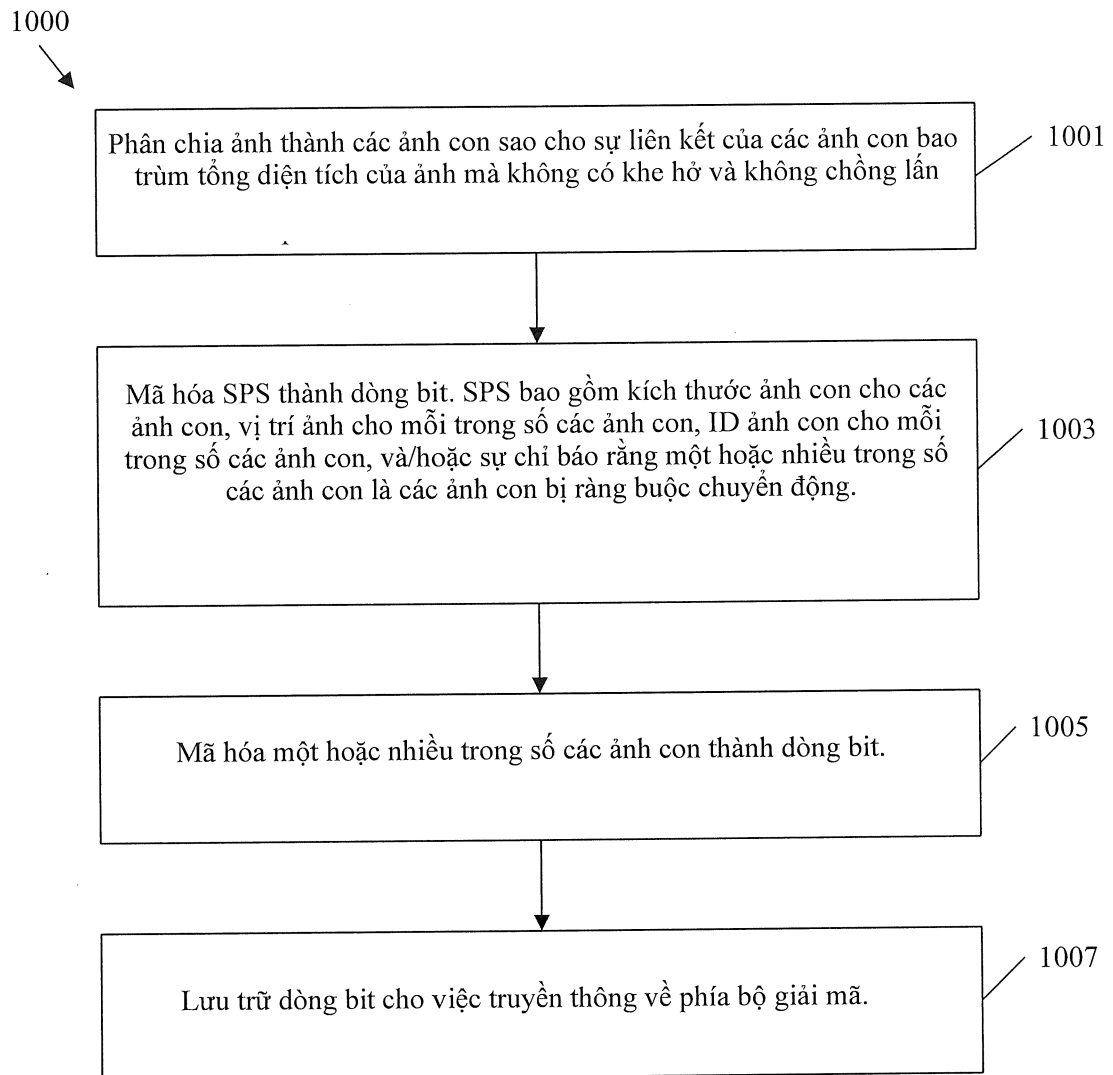


FIG. 10

11/12

1100

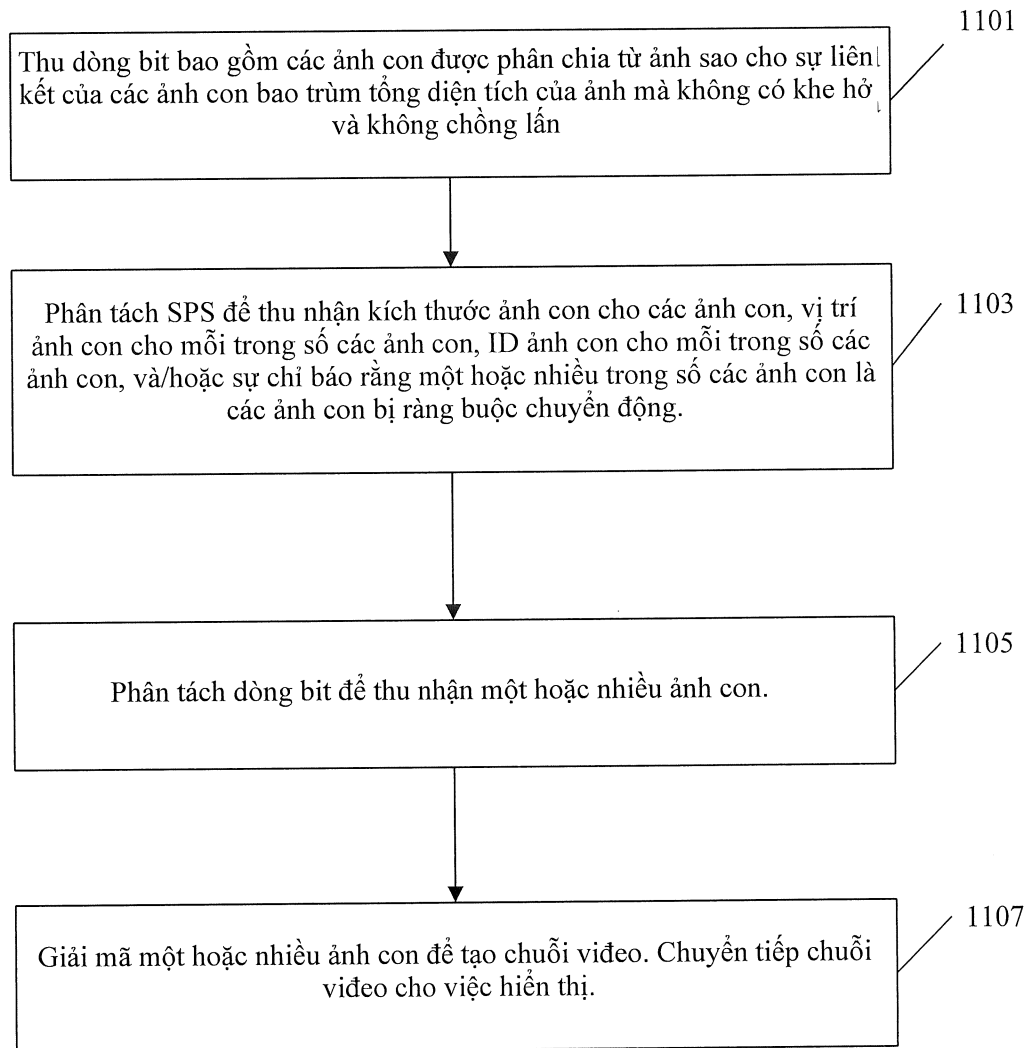


FIG. 11

12/12

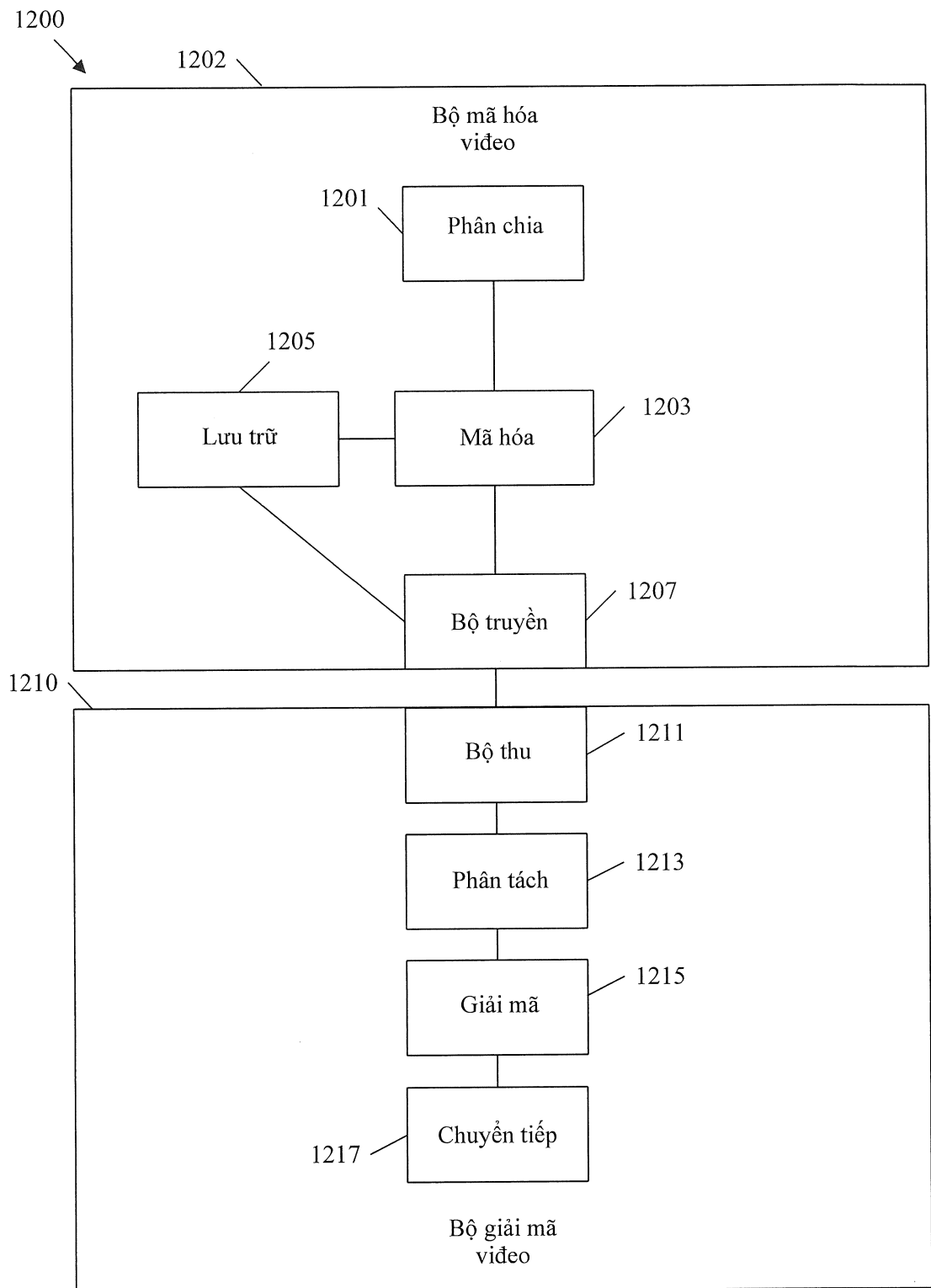


FIG. 12