



(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)



1-0041405

(51)^{2020.01} H04N 19/52; H04N 19/593

(13) B

(21) 1-2021-03571

(22) 10/02/2020

(86) PCT/CN2020/074575 10/02/2020

(87) WO2020/177505 10/09/2020

(30) 62/813,690 04/03/2019 US; 62/815,311 07/03/2019 US

(45) 25/10/2024 439

(43) 25/11/2021 404

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

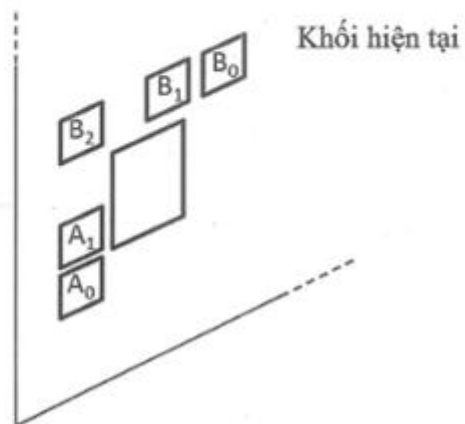
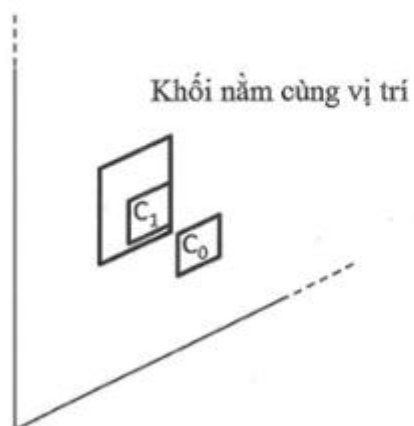
Huawei Administration Building Bantian, Longgang District Shenzhen, Guangdong 518129, China

(72) GAO, Han (CN); ESENLİK, Semih (TR); WANG, Biao (CN); KOTRA, Anand Meher (IN); CHEN, Jianle (CN).

(74) Công ty Luật TNHH Phạm và Liên danh (PHAM & ASSOCIATES)

(54) PHƯƠNG PHÁP TẠO DANH SÁCH HỢP NHẤT ỨNG VIÊN ĐỐI VỚI CHẾ ĐỘ SAO CHÉP TRONG KHỐI, BỘ MÃ HÓA, BỘ GIẢI MÃ, THIẾT BỊ GIẢI MÃ DỮ LIỆU VIDEO VÀ VẬT GHI MÁY TÍNH ĐỌC ĐƯỢC

(57) Sáng chế đề cập đến các phương pháp và các thiết bị tạo danh sách hợp nhất ứng viên đối với chế độ sao chép trong khối (Intra block copy, IBC), phương pháp bao gồm các bước: chèn vector khối của khối lân cận bên trái của khối hiện tại vào danh sách hợp nhất ban đầu của khối hiện tại, khi khối lân cận bên trái khả dụng và khối lân cận bên trái đang sử dụng chế độ IBC; chèn vector khối của khối lân cận bên trên của khối hiện tại vào danh sách hợp nhất ban đầu, khi khối lân cận bên trên khả dụng, khối lân cận bên trên đang sử dụng chế độ IBC và vector khối của khối lân cận bên trên không giống như vector khối của khối lân cận bên trái; chèn vector khối của ứng viên cuối cùng trong bộ dự báo vector chuyển động dựa trên lịch sử (history based motion vector predictor, HMVP) vào danh sách hợp nhất ban đầu, khi vector khối của khối lân cận bên trên không giống như vector khối của ứng viên cuối cùng trong HMVP và khi vector khối của khối lân cận bên trái không giống như vector khối của ứng viên cuối cùng trong HMVP; thu được vector khối của khối hiện tại theo danh sách hợp nhất ban đầu sau các tiến trình chèn nêu trên và chỉ số ứng viên hợp nhất đối với khối hiện tại.



Lĩnh vực kỹ thuật được đề cập

Sáng chế nói chung đề cập đến lĩnh vực xử lý ảnh và cụ thể hơn đến danh sách dự báo được chia sẻ.

Tình trạng kỹ thuật của sáng chế

Kỹ thuật tạo mã video (mã hóa và giải mã video) được sử dụng rộng rãi trong các ứng dụng video số, chẳng hạn TV số phát sóng, truyền video trên mạng Internet và các mạng di động, các ứng dụng hội thoại thời gian thực chẳng hạn trò chuyện video, hội nghị video, đĩa DVD và Blu-ray, các hệ thống chỉnh sửa và thu thập nội dung video, và các máy quay video cỡ nhỏ của các ứng dụng bảo mật.

Lượng dữ liệu video cần mô tả, thậm chí video tương đối ngắn, có thể tương đối lớn, có thể dẫn đến các khó khăn khi dữ liệu được truyền stream hoặc được truyền theo cách khác qua mạng truyền thông có dung lượng băng thông hữu hạn. Do vậy, dữ liệu video thường được nén trước khi được truyền qua các mạng viễn thông hiện đại. Kích thước video cũng có thể là vấn đề khi video được lưu trữ trong bộ lưu trữ do 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 tạo mã dữ liệu video trước khi truyền hoặc lưu trữ, nhờ đó giảm lượng dữ liệu cần để biểu diễn các ảnh video số. Sau đó dữ liệu bị nén được nhận ở phía đích bởi thiết bị giải nén video để giải mã dữ liệu video. Với các tài nguyên mạng hữu hạn và các nhu cầu ngày càng tăng với chất lượng video cao hơn, các kỹ thuật nén và giải nén được cải tiến để cải thiện tỷ lệ nén mà không ảnh hưởng chất lượng ảnh đang được mong muốn.

Bản chất kỹ thuật của sáng chế

Sáng chế đề cập đến các thiết bị và các phương pháp để mã hóa và giải mã theo các điểm yêu cầu bảo hộ độc lập.

Các mục đích khác và nêu trên đạt được bởi đối tượng của các điểm yêu cầu bảo hộ độc lập. Các dạng triển khai khác rõ ràng từ các điểm yêu cầu bảo hộ phụ thuộc, phần mô tả và các hình vẽ.

Khía cạnh thứ nhất của sáng chế bộc lộ phương pháp tạo danh sách hợp nhất ứng viên đối với chế độ sao chép trong khối (Intra block copy, IBC), phương pháp bao gồm các bước:

chèn vector khối của khối lân cận bên trái của khối hiện tại vào danh sách hợp nhất ban đầu của khối hiện tại (trong ví dụ, danh sách hợp nhất ban đầu là danh sách trống trước bước chèn này), khi khối lân cận bên trái khả dụng và khối lân cận bên trái đang sử dụng chế độ IBC;

chèn vector khối của khối lân cận bên trên của khối hiện tại vào danh sách hợp nhất ban đầu (trong ví dụ, danh sách hợp nhất ban đầu là danh sách trống trước bước này, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trái của khối hiện tại), khi khối lân cận bên trên khả dụng, khối lân cận bên trên đang sử dụng chế độ IBC và vector khối của khối lân cận bên trên không giống như vector khối của khối lân cận bên trái;

chèn vector khối của ứng viên cuối cùng trong bộ dự báo vector chuyển động dựa trên lịch sử (history based motion vector predictor, HMVP) vào danh sách hợp nhất ban đầu, khi vector khối của khối lân cận bên trên không giống như vector khối của ứng viên cuối cùng trong HMVP và khi vector khối của khối lân cận bên trái không giống như vector khối của ứng viên cuối cùng trong HMVP.

Khía cạnh thứ hai của sáng chế bộc lộ phương pháp tạo danh sách hợp nhất ứng viên đối với chế độ IBC, phương pháp bao gồm các bước:

chèn vector khối của khối lân cận của khối hiện tại vào danh sách hợp nhất ban đầu của khối hiện tại, khi khối lân cận khả dụng và khối lân cận đang sử dụng chế độ IBC;

chèn vector khối của ứng viên cuối cùng trong HMVP vào danh sách hợp nhất ban đầu, khi vector khối của khối lân cận không giống như vector khối của

ứng viên cuối cùng trong HMVP;

chèn vector khối của ứng viên khác trong HMVP vào danh sách hợp nhất ban đầu, trong đó phép xen tia đối với vector khối của ứng viên khác trong HMVP được loại bỏ.

Khía cạnh thứ ba của sáng chế bộc lộ phương pháp tạo danh sách hợp nhất ứng viên đối với chế độ IBC, phương pháp bao gồm các bước:

chèn vector khối của khối lân cận bên trái của khối hiện tại vào danh sách hợp nhất ban đầu của khối hiện tại (trong ví dụ, danh sách hợp nhất ban đầu là danh sách trống trước bước chèn này), khi khối lân cận bên trái khả dụng và khối lân cận bên trái đang sử dụng chế độ IBC;

chèn vector khối của khối lân cận bên trên của khối hiện tại vào danh sách hợp nhất ban đầu (trong ví dụ, danh sách hợp nhất ban đầu là danh sách trống trước bước này, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trái của khối hiện tại), khi khối lân cận bên trên khả dụng, khối lân cận bên trên đang sử dụng chế độ IBC và vector khối của khối lân cận bên trên không giống như vector khối của khối lân cận bên trái;

chèn vector khối của ứng viên cuối cùng trong HMVP vào danh sách hợp nhất ban đầu, khi vector khối của khối lân cận bên trên không giống như vector khối của ứng viên cuối cùng trong HMVP và khi vector khối của khối lân cận bên trái không giống như vector khối của ứng viên cuối cùng trong HMVP;

chèn vector khối của ứng viên khác trong HMVP vào danh sách hợp nhất ban đầu, trong đó phép xen tia đối với vector khối của ứng viên khác trong HMVP được loại bỏ.

Khía cạnh thứ tư của sáng chế bộc lộ phương pháp tạo danh sách hợp nhất ứng viên đối với chế độ IBC, phương pháp bao gồm các bước:

chèn vector khối của khối lân cận của khối hiện tại vào danh sách hợp nhất ban đầu của khối hiện tại, khi khối lân cận khả dụng và khối lân cận đang sử dụng chế độ IBC;

chèn vector khối của ứng viên trong HMVP vào danh sách hợp nhất ban đầu, khi vector khối của khối lân cận không giống như vector khối của ứng viên cuối cùng trong HMVP;

trong đó vector khối cuối cùng trong danh sách hợp nhất ban đầu của khối hiện tại là một vector khối của một ứng viên trong HMVP.

Khía cạnh thứ năm của sáng chế bộc lộ phương pháp tạo danh sách hợp nhất ứng viên đối với chế độ IBC, phương pháp bao gồm các bước:

chèn vector khối của khối lân cận bên trái của khối hiện tại vào danh sách hợp nhất ban đầu của khối hiện tại (trong ví dụ, danh sách hợp nhất ban đầu là danh sách trống trước bước chèn này), khi khối lân cận bên trái khả dụng và khối lân cận bên trái đang sử dụng chế độ IBC;

chèn vector khối của khối lân cận bên trên của khối hiện tại vào danh sách hợp nhất ban đầu (trong ví dụ, danh sách hợp nhất ban đầu là danh sách trống trước bước này, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trái của khối hiện tại), khi khối lân cận bên trên khả dụng, khối lân cận bên trên đang sử dụng chế độ IBC và vector khối của khối lân cận bên trên không giống như vector khối của khối lân cận bên trái;

chèn vector khối của ứng viên cuối cùng trong HMVP vào danh sách hợp nhất ban đầu, khi vector khối của khối lân cận bên trên không giống như vector khối của ứng viên cuối cùng trong HMVP và khi vector khối của khối lân cận bên trái không giống như vector khối của ứng viên cuối cùng trong HMVP;

chèn vector khối của ứng viên khác trong HMVP vào danh sách hợp nhất ban đầu, trong đó phép xen tia đối với vector khối của ứng viên khác trong HMVP được loại bỏ, và

trong đó vector khối cuối cùng trong danh sách hợp nhất ban đầu của khối hiện tại là vector khối của ứng viên khác trong HMVP.

Theo một triển khai khả thi đối với khía cạnh bất kỳ trong các khía cạnh thứ nhất đến khía cạnh thứ năm, phương pháp còn bao gồm bước: thu được vector khối của khối hiện tại theo danh sách hợp nhất ban đầu sau các tiến trình chèn nêu trên và chỉ số ứng viên hợp nhất đối với khối hiện tại.

Theo một triển khai khả thi đối với khía cạnh bất kỳ trong các khía cạnh thứ nhất đến khía cạnh thứ năm, trong đó danh sách hợp nhất ban đầu là danh sách trống trước tiến trình chèn thứ nhất.

Theo một triển khai khả thi đối với khía cạnh bất kỳ trong các khía cạnh thứ

nhất đến khía cạnh thứ năm, các tiến trình chèn nêu trên được thực hiện theo thứ tự.

Khía cạnh thứ sáu của sáng chế bộc lộ bộ mã hóa (20) bao gồm hệ mạch xử lý để thực hiện một trong các phương án thực hiện phương pháp trước đó.

Khía cạnh thứ bảy của sáng chế bộc lộ bộ giải mã (30) bao gồm hệ mạch xử lý để thực hiện một trong các phương án thực hiện phương pháp trước đó.

Khía cạnh thứ tám của sáng chế bộc lộ sản phẩm chương trình máy tính bao gồm mã chương trình để thực hiện một trong các phương án thực hiện phương pháp trước đó.

Khía cạnh thứ chín của sáng chế bộc lộ bộ giải mã hoặc bộ mã hóa, bao gồm:

một hoặc nhiều bộ xử lý; và

vật ghi máy tính đọc được bất biến được ghép nối với các bộ xử lý và lưu trữ chương trình để thực thi bởi các bộ xử lý, trong đó chương trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện một trong các phương án thực hiện phương pháp trước đó.

Các chi tiết của một hoặc nhiều phương án thực hiện được nêu trong các hình vẽ đi kèm và phần mô tả dưới đây. Các dấu hiệu, các đối tượng, và các ưu điểm khác sẽ trở nên rõ ràng trong phần mô tả, các hình vẽ, và các điểm yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

Các phương án thực hiện sau của sáng chế được mô tả chi tiết hơn dựa vào các hình vẽ đi kèm, trong đó:

Fig.1A là sơ đồ khối thể hiện ví dụ của hệ thống tạo mã video được tạo cấu hình để thực hiện các phương án thực hiện sáng chế;

Fig.1B là sơ đồ khối thể hiện ví dụ khác của hệ thống tạo mã video được tạo cấu hình để thực hiện các phương án thực hiện sáng chế;

Fig.2 là sơ đồ khối thể hiện ví dụ của bộ mã hóa video được tạo cấu hình để thực hiện các phương án thực hiện sáng chế;

Fig.3 là sơ đồ khối thể hiện cấu trúc ví dụ của bộ giải mã video được tạo cấu hình để thực hiện các phương án thực hiện sáng chế;

Fig.4 là sơ đồ khối minh họa ví dụ của thiết bị mã hóa hoặc thiết bị giải mã;

Fig.5 là sơ đồ khối minh họa ví dụ khác của thiết bị mã hóa hoặc thiết bị giải mã;

Fig.6 là sơ đồ khối minh họa ví dụ của các khối lân cận của khối hiện tại.

Fig.7 là hình vẽ thể hiện ví dụ về phương án thực hiện sáng chế;

Fig.8 là sơ đồ khối thể hiện cấu trúc ví dụ của hệ thống cung cấp nội dung 3100 thực hiện dịch vụ truyền nội dung; và

Fig.9 là sơ đồ khối thể hiện cấu trúc của ví dụ của thiết bị đầu cuối.

Dưới đây, các dấu chỉ dẫn giống nhau đề cập đến các dấu hiệu tương đương ít nhất về mặt chức năng hoặc giống hệt nếu không có thông báo khác.

Mô tả chi tiết sáng chế

Trong phần mô tả sau, có thể thực hiện tham khảo các hình vẽ sau, mà tạo thành một phần của sáng chế, và thể hiện, thông qua minh họa, các khía cạnh cụ thể của các phương án thực hiện sáng chế hoặc các khía cạnh cụ thể trong đó các phương án thực hiện sáng chế có thể được sử dụng. Cần hiểu rằng các phương án thực hiện sáng chế có thể được sử dụng theo các khía cạnh khác và bao gồm các thay đổi cấu trúc hoặc logic không được mô tả trong các hình vẽ. Do vậy, phần mô tả chi tiết sau không được hiểu theo nghĩa giới hạn, và phạm vi của sáng chế được định nghĩa bởi các điểm yêu cầu bảo hộ đi kèm.

Chẳng hạn, cần hiểu rằng sáng chế cùng với phương pháp được mô tả cũng có thể đúng đối với thiết bị hoặc hệ thống tương ứng được tạo cấu hình để thực hiện phương pháp và ngược lại. Chẳng hạn, nếu một hoặc nhiều bước phương pháp cụ thể được mô tả, thiết bị tương ứng có thể bao gồm một hoặc nhiều khối, chẳng hạn, các khối chức năng, để thực hiện một hoặc nhiều bước phương pháp được mô tả (chẳng hạn, một khối thực hiện một hoặc nhiều bước, hoặc các khối mà mỗi khối thực hiện một hoặc nhiều bước trong các bước), thậm chí nếu một hoặc nhiều khối không được mô tả hoặc minh họa rõ ràng trên các hình vẽ. Tuy nhiên, chẳng hạn, nếu thiết bị cụ thể được mô tả dựa trên

một hoặc nhiều khối, chẳng hạn, các khối chức năng, phương pháp tương ứng có thể bao gồm một bước để thực hiện chức năng của một hoặc nhiều khối (chẳng hạn, một bước thực hiện chức năng của một hoặc nhiều khối, hoặc các bước mà mỗi bước thực hiện chức năng của một hoặc nhiều khối trong các khối), thậm chí nếu một hoặc nhiều bước này không được mô tả hoặc minh họa rõ ràng trên các hình vẽ. Ngoài ra, cần hiểu rằng các dấu hiệu của các phương án thực hiện lấy làm ví dụ khác và/hoặc các khía cạnh được mô tả ở đây có thể được kết hợp với nhau, trừ khi có thông báo khác.

Kỹ thuật tạo mã video thường đề cập đến kỹ thuật xử lý chuỗi ảnh, tạo video hoặc chuỗi video. Thay cho cụm từ “ảnh”, cụm từ “khung” hoặc “hình ảnh” có thể được sử dụng làm các từ đồng nghĩa trong lĩnh vực tạo mã video. Kỹ thuật tạo mã video (hoặc tạo mã nói chung) bao gồm hai phần mã hóa video và giải mã video. Mã hóa video được thực hiện ở phía nguồn, thường bao gồm xử lý (chẳng hạn, bằng cách nén) các ảnh video gốc để giảm lượng dữ liệu cần để biểu diễn các ảnh video (để lưu trữ và/hoặc truyền hiệu quả hơn). Kỹ thuật giải mã video được thực hiện ở phía đích và thường bao gồm xử lý ngược lại so với bộ mã hóa để tái tạo các ảnh video. Các phương án thực hiện đề cập đến việc “tạo mã” các ảnh video (hoặc ảnh nói chung) cần hiểu là liên quan đến “mã hóa” hoặc “giải mã” các ảnh video hoặc các chuỗi video tương ứng. Việc kết hợp phần mã hóa và phần giải mã cũng được gọi là CODEC (Coding and Decoding).

Trong trường hợp tạo mã video không tổn hao, các ảnh video gốc có thể được tái tạo, tức là, các ảnh video được tái tạo có cùng chất lượng với các ảnh video gốc (giả sử không tổn hao khi truyền hoặc tổn hao dữ liệu khác trong khi lưu trữ hoặc truyền). Trong trường hợp tạo mã video tổn hao, bước nén tiếp, chẳng hạn, bằng cách lượng tử hóa, được thực hiện, để giảm lượng dữ liệu biểu diễn các ảnh video, không thể được tái tạo hoàn toàn ở bộ giải mã, tức là, chất lượng của các ảnh video được tái tạo thấp hơn hoặc kém hơn so với chất lượng của các ảnh video gốc.

Một số chuẩn tạo mã video thuộc nhóm “bộ codec video lai tổn hao” (tức là, kết hợp dự báo không gian và thời gian trong miền mẫu và tạo mã biến đổi

2D để áp dụng lượng tử hóa trong miền biến đổi). Mỗi ảnh của chuỗi video thường được phân vùng thành tập hợp khối không trùng lặp và việc tạo mã thường được thực hiện ở cấp độ khối. Nói cách khác, ở bộ mã hóa, video thường được xử lý, tức là, được mã hóa, trên cấp độ khối (khối video), chẳng hạn, bằng cách sử dụng dự báo không gian (trong ảnh) và/hoặc dự báo thời gian (ngoài ảnh) để tạo khối dự báo, lấy khối hiện tại trừ đi khối dự báo (khối đang được xử lý/cần được xử lý) để thu được khối dư, biến đổi khối dư và lượng tử hóa khối dư trong miền biến đổi để giảm lượng dữ liệu cần được truyền (nén), cùng lúc ở bộ giải mã xử lý ngược lại so với bộ mã hóa được áp dụng cho khối được mã hóa hoặc được nén để tái tạo khối hiện tại để biểu diễn. Ngoài ra, bộ mã hóa sao chép vòng xử lý bộ giải mã sao cho cả hai sẽ tạo các dự báo giống hệt (chẳng hạn, các dự báo trong và ngoài) và/hoặc các tái tạo để xử lý, tức là, tạo mã, các khối tiếp theo.

Theo các phương án thực hiện sau, hệ thống tạo mã video 10, bộ mã hóa video 20 và bộ giải mã video 30 được mô tả dựa trên các hình vẽ từ Fig.1 đến Fig.3.

Fig.1A là sơ đồ khối minh họa hệ thống tạo mã lấy làm ví dụ 10, chẳng hạn, hệ thống tạo mã video 10 (hoặc gọi tắt là hệ thống tạo mã 10) có thể sử dụng các kỹ thuật của sáng chế. Bộ mã hóa video 20 (hoặc gọi tắt là bộ mã hóa 20) và bộ giải mã video 30 (hoặc gọi tắt là bộ giải mã 30) của hệ thống tạo mã video 10 là các ví dụ của các thiết bị mà có thể được tạo cấu hình để thực hiện các kỹ thuật theo các ví dụ khác nhau được mô tả theo sáng chế.

Như được thể hiện trên Fig.1A, hệ thống tạo mã 10 bao gồm thiết bị nguồn 12 được tạo cấu hình để cấp dữ liệu ảnh được mã hóa 21 chẳng hạn, đến thiết bị đích 14 để giải mã dữ liệu ảnh được mã hóa 13.

Thiết bị nguồn 12 bao gồm bộ mã hóa 20, và có thể bao gồm thêm, tức là tùy chọn, nguồn ảnh 16, bộ tiền xử lý (hoặc khối tiền xử lý) 18, chẳng hạn, bộ tiền xử lý ảnh 18, và giao diện truyền thông hoặc khối truyền thông 22.

Nguồn ảnh 16 có thể bao gồm là loại thiết bị chụp ảnh bất kỳ, chẳng hạn camera để chụp ảnh thế giới thực, và/hoặc loại bất kỳ của thiết bị tạo ảnh, chẳng hạn bộ xử lý đồ họa máy tính để tạo ảnh động bằng máy tính, hoặc loại

thiết bị khác để thu được và/hoặc tạo ảnh thể giới thực, ảnh được tạo bằng máy tính (chẳng hạn, nội dung màn hình, ảnh thực tại ảo (virtual reality, VR)) và/hoặc tổ hợp bất kỳ của nó (chẳng hạn, ảnh thực tế tăng cường (augmented reality, AR)). Nguồn ảnh có thể là loại bộ nhớ hoặc bộ lưu trữ bất kỳ lưu trữ ảnh bất kỳ trong các ảnh nêu trên.

Để phân biệt với bộ tiền xử lý 18 và xử lý được thực hiện bởi khối tiền xử lý 18, ảnh hoặc dữ liệu ảnh 17 cũng có thể được gọi là ảnh thô hoặc dữ liệu ảnh thô 17.

Bộ tiền xử lý 18 được tạo cấu hình để nhận dữ liệu ảnh (thô) 17 và để thực hiện tiền xử lý trên dữ liệu ảnh 17 để thu được ảnh được tiền xử lý 19 hoặc dữ liệu ảnh được tiền xử lý 19. Tiền xử lý được thực hiện bởi bộ tiền xử lý 18 có thể, chẳng hạn, bao gồm cắt tỉa, biến đổi định dạng màu (chẳng hạn, từ RGB sang YCbCr), hiệu chỉnh màu, hoặc khử nhiễu. Có thể được hiểu rằng khối tiền xử lý 18 có thể là thành phần tùy chọn.

Bộ mã hóa video 20 được tạo cấu hình để nhận dữ liệu ảnh được tiền xử lý 19 và cấp dữ liệu ảnh được mã hóa 21 (các chi tiết sau sẽ được mô tả dưới đây, chẳng hạn, dựa trên Fig.2).

Giao diện truyền thông 22 của thiết bị nguồn 12 có thể được tạo cấu hình để nhận dữ liệu ảnh được mã hóa 21 và để truyền dữ liệu ảnh được mã hóa 21 (hoặc phiên bản được xử lý tiếp bất kỳ của nó) trên kênh truyền thông 13 đến thiết bị khác, chẳng hạn, thiết bị đích 14 hoặc thiết bị khác bất kỳ, để lưu trữ hoặc tái tạo trực tiếp.

Thiết bị đích 14 bao gồm bộ giải mã 30 (chẳng hạn, bộ giải mã video 30), và có thể bao gồm thêm, tức là tùy chọn, giao diện truyền thông hoặc khối truyền thông 28, bộ hậu xử lý 32 (hoặc khối hậu xử lý 32) và thiết bị hiển thị 34.

Giao diện truyền thông 28 của thiết bị đích 14 được tạo cấu hình để nhận dữ liệu ảnh được mã hóa 21 (hoặc phiên bản được xử lý tiếp bất kỳ của nó), chẳng hạn, trực tiếp từ thiết bị nguồn 12 hoặc từ thiết bị nguồn khác bất kỳ, chẳng hạn, bộ lưu trữ, chẳng hạn, bộ lưu trữ dữ liệu ảnh được mã hóa, và cấp dữ liệu ảnh được mã hóa 21 cho bộ giải mã 30.

Giao diện truyền thông 22 và giao diện truyền thông 28 có thể được tạo cấu hình để truyền hoặc nhận dữ liệu ảnh được mã hóa 21 hoặc dữ liệu được mã hóa 13 qua liên kết truyền thông trực tiếp giữa thiết bị nguồn 12 và thiết bị đích 14, chẳng hạn, kết nối không dây hoặc hữu tuyến trực tiếp, hoặc qua loại mạng bất kỳ, chẳng hạn, mạng không dây hoặc hữu tuyến hoặc tổ hợp bất kỳ của nó, hoặc loại mạng bất kỳ trong mạng riêng tư và công khai, hoặc loại tổ hợp bất kỳ của nó.

Giao diện truyền thông 22 có thể, chẳng hạn, được tạo cấu hình để đóng gói dữ liệu ảnh được mã hóa 21 thành định dạng thích hợp, chẳng hạn, gói, và/hoặc xử lý dữ liệu ảnh được mã hóa nhờ sử dụng hoạt động bất kỳ mã hóa hoặc xử lý truyền để truyền trên liên kết truyền thông hoặc mạng truyền thông.

Giao diện truyền thông 28, tạo thành bản sao của giao diện truyền thông 22, có thể, chẳng hạn, được tạo cấu hình để nhận dữ liệu được truyền và xử lý dữ liệu truyền nhờ sử dụng hoạt động bất kỳ giải mã hoặc xử lý truyền tương ứng và/hoặc tháo gói để thu được dữ liệu ảnh được mã hóa 21.

Cả giao diện truyền thông 22 lẫn giao diện truyền thông 28 có thể được tạo cấu hình dưới dạng các giao diện truyền thông đơn hướng như được chỉ báo bởi mũi tên cho kênh truyền thông 13 trên Fig.1A chỉ từ thiết bị nguồn 12 đến thiết bị đích 14, hoặc các giao diện truyền thông hai hướng, và có thể được tạo cấu hình, chẳng hạn, để gửi và nhận các thông điệp, chẳng hạn, để thiết lập kết nối, để báo nhận và trao đổi thông tin khác bất kỳ liên quan đến liên kết truyền thông và/hoặc truyền dữ liệu, chẳng hạn, truyền dữ liệu ảnh được mã hóa.

Bộ giải mã 30 được tạo cấu hình để nhận dữ liệu ảnh được mã hóa 21 và cấp dữ liệu ảnh được giải mã 31 hoặc ảnh được giải mã 31 (các chi tiết sau sẽ được mô tả dưới đây, chẳng hạn, dựa trên Fig.3 hoặc Fig.5).

Bộ hậu xử lý 32 của thiết bị đích 14 được tạo cấu hình để hậu xử lý dữ liệu ảnh được giải mã 31 (cũng được gọi là dữ liệu ảnh được tái tạo), chẳng hạn, ảnh được giải mã 31, để thu được dữ liệu ảnh được hậu xử lý 33, chẳng hạn, ảnh được hậu xử lý 33. Hoạt động hậu xử lý được thực hiện bởi khối hậu xử lý 32 có thể bao gồm, chẳng hạn, biến đổi định dạng màu (chẳng hạn, từ YCbCr sang RGB), hiệu chỉnh màu, cắt tia, hoặc lấy mẫu lại, hoặc hoạt động xử lý

khác bất kỳ, chẳng hạn, để chuẩn bị dữ liệu ảnh được giải mã 31 để hiển thị, chẳng hạn, bởi bộ phận hiển thị 34.

Bộ phận hiển thị 34 của thiết bị đích 14 được tạo cấu hình để nhận dữ liệu ảnh được xử lý 33 để hiển thị ảnh, chẳng hạn, cho người dùng hoặc người xem. Bộ phận hiển thị 34 có thể là hoặc bao gồm loại bộ phận hiển thị bất kỳ để biểu diễn ảnh được tái tạo, chẳng hạn, bộ phận hiển thị hoặc màn hình tích hợp hoặc bên ngoài. Các bộ phận hiển thị có thể, chẳng hạn, bao gồm bộ phận hiển thị tinh thể lỏng (liquid crystal display, LCD), bộ phận hiển thị điốt phát quang hữu cơ (organic light emitting diode, OLED), bộ phận hiển thị plasma, máy chiếu, bộ phận hiển thị micro LED, tinh thể lỏng trên silic (liquid crystal on silicon, LCoS), bộ xử lý ánh sáng số (digital light processor, DLP) hoặc loại hiển thị khác bất kỳ.

Mặc dù Fig.1A mô tả thiết bị nguồn 12 và thiết bị đích 14 dưới dạng các thiết bị riêng rẽ, song theo các phương án thực hiện, các thiết bị cũng có thể bao gồm cả hai chức năng, thiết bị nguồn 12 hoặc chức năng tương ứng và thiết bị đích 14 hoặc chức năng tương ứng. Theo các phương án thực hiện này, thiết bị nguồn 12 hoặc chức năng tương ứng và thiết bị đích 14 hoặc chức năng tương ứng có thể được triển khai nhờ sử dụng phần cứng và/hoặc phần mềm tương tự hoặc bằng phần cứng và/hoặc phần mềm riêng rẽ hoặc tổ hợp bất kỳ của nó.

Như người có hiểu biết trung bình trong lĩnh vực sẽ hiểu dựa trên phần mô tả, sự tồn tại và tách ra (chính xác) các chức năng của các khối khác nhau hoặc các chức năng trong thiết bị nguồn 12 và/hoặc thiết bị đích 14 như được thể hiện trên Fig.1A có thể thay đổi tùy thuộc vào thiết bị và ứng dụng thực.

Bộ mã hóa 20 (chẳng hạn, bộ mã hóa video 20) hoặc bộ giải mã 30 (chẳng hạn, bộ giải mã video 30) hoặc cả bộ mã hóa 20 lẫn bộ giải mã 30 có thể được triển khai qua hệ mạch xử lý như được thể hiện trên Fig.1B, chẳng hạn một hoặc nhiều bộ vi xử lý, bộ xử lý tín hiệu số (digital signal processor, DSP), mạch tích hợp ứng dụng cụ thể (application-specific integrated circuit, ASIC), mảng cổng dạng trường lập trình được (field-programmable gate array, FPGA), logic rời rạc, phần cứng, tạo mã video dành riêng hoặc các tổ hợp bất kỳ của

nó. Bộ mã hóa 20 có thể được triển khai qua hệ mạch xử lý 46 để thực hiện các môđun khác nhau như được nêu dựa vào bộ mã hóa 20 trên Fig.2 và/hoặc hệ thống hoặc hệ thống phụ của bộ mã hóa khác bất kỳ được nêu ở đây. Bộ giải mã 30 có thể được triển khai qua hệ mạch xử lý 46 để thực hiện các môđun khác nhau như được nêu dựa vào bộ giải mã 30 trên Fig.3 và/hoặc hệ thống hoặc hệ thống phụ của bộ giải mã khác bất kỳ được nêu ở đây. Hệ mạch xử lý có thể được tạo cấu hình để thực hiện các hoạt động khác nhau như được nêu sau đây. Như được thể hiện trên Fig.5, nếu các kỹ thuật được triển khai một phần trong phần mềm, thiết bị có thể lưu trữ các lệnh đối với phần mềm trong vật lưu trữ máy tính đọc được thích hợp, bất biến và có thể thực thi các lệnh trong phần cứng nhờ sử dụng một hoặc nhiều bộ xử lý để thực hiện các kỹ thuật của sáng chế. Một trong bộ mã hóa video 20 và bộ giải mã video 30 có thể được tích hợp dưới dạng một phần của bộ mã hóa/bộ giải mã kết hợp (CODEC) trong một thiết bị, chẳng hạn, như được thể hiện trên Fig.1B.

Thiết bị nguồn 12 và thiết bị đích 14 có thể bao gồm thiết bị bất kỳ trong các thiết bị, bao gồm loại thiết bị bất kỳ trong các thiết bị cầm tay hoặc tĩnh, chẳng hạn, máy tính notebook hoặc máy tính xách tay, điện thoại di động, điện thoại thông minh, máy tính bảng, camera, máy tính để bàn, hộp tín hiệu truyền hình số, TV, bộ phận hiển thị, máy phát media số, các bảng điều khiển trò chơi video, các dịch vụ phát trực tuyến video (streaming) (chẳng hạn các máy chủ dịch vụ nội dung hoặc các máy chủ phát nội dung), bộ thu phát song, bộ truyền phát sóng, hoặc tương tự và có thể không sử dụng hoặc sử dụng loại hệ điều hành bất kỳ. Trong một số trường hợp, thiết bị nguồn 12 và thiết bị đích 14 có thể được trang bị để truyền thông không dây. Do vậy, thiết bị nguồn 12 và thiết bị đích 14 có thể là các thiết bị truyền thông không dây.

Trong một số trường hợp, hệ thống tạo mã video 10 được minh họa trên Fig.1A chỉ là ví dụ và các kỹ thuật của sáng chế có thể áp dụng cho các thiết lập tạo mã video (chẳng hạn, mã hóa video hoặc giải mã video) mà không nhất thiết bao gồm truyền thông dữ liệu bất kỳ giữa các thiết bị mã hóa và giải mã. Trong các ví dụ khác, dữ liệu được truy xuất từ bộ nhớ cục bộ, được phát trực tuyến trên mạng, hoặc tương tự. Thiết bị mã hóa video có thể mã hóa và

lưu trữ dữ liệu vào bộ nhớ, và/hoặc thiết bị giải mã video có thể truy xuất và giải mã dữ liệu từ bộ nhớ. Trong một số ví dụ, việc mã hóa và giải mã được thực hiện bằng các thiết bị không truyền thông với nhau, mà chỉ mã hóa dữ liệu vào bộ nhớ và/hoặc truy xuất và giải mã dữ liệu từ bộ nhớ.

Để tiện mô tả, các phương án thực hiện sáng chế được mô tả ở đây, chẳng hạn, bằng cách viện dẫn đến kỹ thuật tạo mã video hiệu suất cao (High-Efficiency Video Coding, HEVC) hoặc đến phần mềm tham chiếu của kỹ thuật tạo mã video đa dụng (Versatile Video Coding, VVC), chuẩn tạo mã video thế hệ tiếp theo được phát triển bởi nhóm cộng tác chung về tạo mã video (Joint Collaboration Team on video coding, JCT-VC) của nhóm chuyên gia tạo mã video ITU-T (Video Coding Experts Group, VCEG) và nhóm chuyên gia ảnh động ISO/IEC (Motion Picture Experts Group, MPEG). Người có hiểu biết trung bình trong lĩnh vực sẽ hiểu rằng các phương án thực hiện sáng chế không bị giới hạn ở HEVC hoặc VVC.

Bộ mã hóa và phương pháp mã hóa

Fig. 2 thể hiện sơ đồ khối của bộ mã hóa video 20 lấy làm ví dụ được tạo cấu hình để thực hiện các kỹ thuật của sáng chế. Ở ví dụ trên Fig.2, bộ mã hóa video 20 bao gồm đầu vào 201 (hoặc giao diện đầu vào 201), khối tính toán phần dư 204, khối xử lý biến đổi 206, khối lượng tử hóa 208, khối lượng tử hóa ngược 210, và khối xử lý biến đổi ngược 212, khối tái tạo 214, khối lọc vòng 220, bộ đệm ảnh được giải mã (decoded picture buffer, DPB) 230, khối chọn chế độ 260, khối mã hóa entropy 270 và đầu ra 272 (hoặc giao diện đầu ra 272). Khối chọn chế độ 260 có thể bao gồm khối dự báo ngoài 244, khối dự báo trong 254 và khối phân vùng 262. Khối dự báo ngoài 244 có thể bao gồm khối ước tính chuyển động và khối bù chuyển động (không được thể hiện trên hình vẽ). Bộ mã hóa video 20 như được thể hiện trên Fig.2 cũng có thể được gọi là bộ mã hóa video lai hoặc bộ mã hóa video theo codec video lai.

Khối tính toán phần dư 204, khối xử lý biến đổi 206, khối lượng tử hóa 208, khối chọn chế độ 260 có thể được gọi là tạo thành đường tín hiệu chuyển tiếp của bộ mã hóa 20, trong đó khối lượng tử hóa đảo 210, khối xử lý biến đổi

ngược 212, khối tái tạo 214, bộ đệm 216, khối lọc vòng 220, DPB 230, khối dự báo ngoài 244 và khối dự báo trong 254 có thể được gọi là tạo đường tín hiệu lùi của bộ mã hóa video 20, trong đó đường tín hiệu lùi của bộ mã hóa video 20 tương ứng với đường tín hiệu của bộ giải mã (xem bộ giải mã video 30 trên Fig.3). Khối lượng tử hóa đảo 210, khối xử lý biến đổi ngược 212, khối tái tạo 214, khối lọc vòng 220, DPB 230, khối dự báo ngoài 244 và khối dự báo trong 254 cũng được gọi là tạo “bộ giải mã cài sẵn” của bộ mã hóa video 20.

Ảnh và phân vùng ảnh (Ảnh và khối)

Bộ mã hóa 20 có thể được tạo cấu hình để nhận, chẳng hạn, qua đầu vào 201, ảnh 17 (hoặc dữ liệu ảnh 17), chẳng hạn, ảnh trong chuỗi ảnh tạo thành video hoặc chuỗi video. Ảnh hoặc dữ liệu ảnh được nhận cũng có thể là ảnh được tiền xử lý 19 (hoặc dữ liệu ảnh được tiền xử lý 19). Để đơn giản, phần mô tả sau đề cập đến ảnh 17. Ảnh 17 cũng có thể được gọi là ảnh hiện tại hoặc ảnh cần được tạo mã (cụ thể khi tạo mã video để phân biệt ảnh hiện tại với các ảnh khác, chẳng hạn, ảnh được giải mã và/hoặc được mã hóa trước đó của cùng chuỗi video, tức là, chuỗi video cũng bao gồm ảnh hiện tại).

Ảnh (số) là hoặc có thể được xem là mảng hai chiều hoặc ma trận mẫu có các giá trị cường độ. Mẫu trong mảng cũng có thể được gọi là pixel (viết tắt của phần tử ảnh) hoặc pel. Số lượng mẫu theo chiều ngang và dọc (hoặc trục) của mảng hoặc ảnh định nghĩa kích thước và/hoặc độ phân giải của ảnh. Để biểu diễn màu, thường ba thành phần màu được sử dụng, tức là, ảnh có thể được biểu diễn hoặc bao gồm ba mảng mẫu. Ở định dạng RGB hoặc không gian màu, ảnh bao gồm mảng mẫu đỏ, lục và lam tương ứng. Tuy nhiên, khi tạo mã video, mỗi pixel thường được biểu diễn ở định dạng độ sáng và sắc độ hoặc không gian màu, chẳng hạn, YCbCr, bao gồm thành phần độ sáng được chỉ báo bởi Y (đôi lúc L cũng được sử dụng thay thế) và hai thành phần sắc độ được chỉ báo bởi Cb và Cr. Thành phần độ sáng (hoặc gọi tắt là luma) Y là độ sáng hoặc cường độ mức xám (chẳng hạn, tương tự trong ảnh thang xám), trong khi hai thành phần sắc độ (hoặc viết tắt là chroma) Cb và Cr là sắc sai hoặc thành phần thông tin màu. Do đó, ảnh ở định dạng YCbCr bao gồm mảng mẫu độ

sáng của các giá trị mẫu độ sáng (Y), và hai mảng mẫu sắc độ của các giá trị sắc độ (Cb và Cr). Các ảnh ở định dạng RGB có thể được biến đổi hoặc chuyển đổi thành định dạng YCbCr và ngược lại, quá trình này cũng được biết đến là chuyển đổi hoặc biến đổi màu. Nếu ảnh là đơn sắc, ảnh có thể bao gồm chỉ mảng mẫu độ sáng. Do đó, ảnh có thể là, chẳng hạn, mảng của các mẫu độ sáng ở định dạng đơn sắc hoặc mảng của các mẫu độ sáng và hai mảng tương ứng của các mẫu sắc độ ở định dạng màu 4:2:0, 4:2:2, và 4:4:4.

Theo các phương án thực hiện, bộ mã hóa video 20 có thể bao gồm khối phân vùng ảnh (không được mô tả trên Fig.2) được tạo cấu hình để phân vùng ảnh 17 thành các khối ảnh (thường là không trùng lặp) 203. Các khối này cũng có thể được gọi là các khối gốc, các khối lớn (H.264/AVC) hoặc các khối cây mã (coding tree block, CTB) hoặc các đơn vị cây mã (coding tree unit, CTU) (H.265/HEVC và VVC). Khối phân vùng ảnh có thể được tạo cấu hình để sử dụng cùng kích thước khối đối với tất cả ảnh của chuỗi video và lưới tương ứng định nghĩa kích thước khối, hoặc để thay đổi kích thước khối giữa các ảnh hoặc các tập hợp con hoặc các nhóm ảnh, và phân vùng mỗi ảnh thành các khối tương ứng.

Theo các phương án thực hiện khác, bộ mã hóa video có thể được tạo cấu hình để trực tiếp nhận khối 203 của ảnh 17, chẳng hạn, một, vài hoặc tất cả các khối tạo thành ảnh 17. Khối ảnh 203 cũng có thể được gọi là khối ảnh hiện tại hoặc khối ảnh cần được tạo mã.

Giống như ảnh 17, khối ảnh 203 lại là hoặc có thể được xem là mảng hai chiều hoặc ma trận mẫu có các giá trị cường độ (các giá trị mẫu), mặc dù có kích thước nhỏ hơn ảnh 17. Nói cách khác, khối 203 có thể bao gồm, chẳng hạn, một mảng mẫu (chẳng hạn, mảng độ sáng trong trường hợp của ảnh đơn sắc 17, hoặc mảng độ sáng hoặc sắc độ trong trường hợp của ảnh màu) hoặc ba mảng mẫu (chẳng hạn, một mảng độ sáng và hai mảng sắc độ trong trường hợp của ảnh màu 17) hoặc số lượng khác bất kỳ và/hoặc loại mảng phụ thuộc vào định dạng màu được áp dụng. Số lượng mẫu theo chiều ngang và dọc (hoặc trục) của khối 203 định nghĩa kích thước của khối 203. Do đó, khối có thể,

chẳng hạn, mảng $M \times N$ mẫu (gồm M cột x N hàng), hoặc mảng $M \times N$ hệ số biến đổi.

Theo các phương án thực hiện, bộ mã hóa video 20 như được thể hiện trên Fig.2 có thể được tạo cấu hình để mã hóa ảnh 17 theo từng khối, chẳng hạn, hoạt động mã hóa và dự báo được thực hiện theo khối 203.

Theo các phương án thực hiện, bộ mã hóa video 20 như được thể hiện trên Fig.2 có thể còn được tạo cấu hình để phân vùng và/hoặc mã hóa ảnh nhờ sử dụng các lát (cũng được gọi là các lát video), trong đó ảnh có thể được phân vùng thành hoặc được mã hóa nhờ sử dụng một hoặc nhiều lát (thường không trùng lặp), và mỗi lát có thể bao gồm một hoặc nhiều khối (chẳng hạn, các CTU).

Theo các phương án thực hiện, bộ mã hóa video 20 như được thể hiện trên Fig.2 có thể còn được tạo cấu hình để phân vùng và/hoặc mã hóa ảnh nhờ sử dụng các nhóm phiên (cũng được gọi là các nhóm phiên video) và/hoặc các phiên (cũng được gọi là các phiên video), trong đó ảnh có thể được phân vùng thành hoặc được mã hóa nhờ sử dụng một hoặc nhiều nhóm phiên (thường không trùng lặp), và mỗi nhóm phiên có thể bao gồm, chẳng hạn, một hoặc nhiều khối (chẳng hạn, các CTU) hoặc một hoặc nhiều phiên, trong đó mỗi phiên, chẳng hạn, có thể có hình chữ nhật và có thể bao gồm một hoặc nhiều khối (chẳng hạn, các CTU), chẳng hạn, các khối hoàn chỉnh hoặc một phần.

Tính toán phần dư

Khối tính toán phần dư 204 có thể được tạo cấu hình để tính toán khối dư 205 (cũng được gọi là phần dư 205) dựa trên khối ảnh 203 và khối dự báo 265 (các chi tiết khác về khối dự báo 265 được nêu sau đây), chẳng hạn, bằng cách lấy các giá trị mẫu của khối ảnh 203 trừ đi các giá trị mẫu của khối dự báo 265, theo từng mẫu (theo từng pixel) để thu được khối dư 205 trong miền mẫu.

Biến đổi

Khối xử lý biến đổi 206 có thể được tạo cấu hình để áp dụng biến đổi, chẳng hạn, biến đổi côsin rời rạc (discrete cosine transform, DCT) hoặc biến

đổi sin rời rạc (discrete sine transform, DST), trên các giá trị mẫu của khối dư 205 để thu được các hệ số biến đổi 207 trong miền biến đổi. Các hệ số biến đổi 207 cũng có thể được gọi là các hệ số dư biến đổi và là khối dư 205 trong miền biến đổi.

Khối xử lý biến đổi 206 có thể được tạo cấu hình để áp dụng các phép xấp xỉ số nguyên của DCT/DST, chẳng hạn các phép biến đổi được xác định cho H.265/HEVC. So với biến đổi DCT trực giao, các phép xấp xỉ số nguyên này thường được chia tỷ lệ bởi hệ số cụ thể. Để bảo toàn chuẩn của khối dư mà được xử lý bằng các phép biến đổi thuận và ngược, các hệ số chia tỷ lệ bổ sung được áp dụng dưới dạng một phần của quá trình biến đổi. Các hệ số chia tỷ lệ thường được chọn dựa trên các ràng buộc cụ thể giống như các hệ số chia tỷ lệ là lũy thừa của 2 đối với các toán tử dịch, độ sâu bit của các hệ số biến đổi, cân bằng giữa độ chính xác và các chi phí triển khai, v.v.. Các hệ số chia tỷ lệ cụ thể chẳng hạn, được xác định đối với phép biến đổi ngược, chẳng hạn, bằng khối xử lý biến đổi ngược 212 (và biến đổi ngược tương ứng, chẳng hạn, bằng khối xử lý biến đổi ngược 312 ở bộ giải mã video 30) và các hệ số chia tỷ lệ tương ứng đối với phép biến đổi thuận, chẳng hạn, bởi khối xử lý biến đổi 206, ở bộ mã hóa 20 có thể được xác định theo đó.

Theo các phương án thực hiện, bộ mã hóa video 20 (khối xử lý biến đổi 206 tương ứng) có thể được tạo cấu hình để xuất ra các tham số biến đổi, chẳng hạn, loại biến đổi hoặc các phép biến đổi, chẳng hạn, trực tiếp hoặc được mã hóa hoặc được nén qua khối mã hóa entropy 270, sao cho, chẳng hạn, bộ giải mã video 30 có thể nhận và sử dụng các tham số biến đổi để giải mã.

Lượng tử hóa

Khối lượng tử hóa 208 có thể được tạo cấu hình để lượng tử hóa các hệ số biến đổi 207 để thu được các hệ số được lượng tử hóa 209, chẳng hạn, bằng cách áp dụng lượng tử hóa vô hướng hoặc lượng tử hóa vectơ. Các hệ số được lượng tử hóa 209 cũng có thể được gọi là các hệ số biến đổi được lượng tử hóa 209 hoặc các hệ số dư được lượng tử hóa 209.

Quá trình lượng tử hóa có thể giảm chiều sâu bit được liên kết với một số hoặc tất cả các hệ số biến đổi 207. Chẳng hạn, hệ số biến đổi n bit có thể được làm tròn xuống hệ số biến đổi m bit trong khi lượng tử hóa, trong đó $n > m$. Mức độ lượng tử hóa có thể được thay đổi bằng cách điều chỉnh tham số lượng tử hóa (quantization parameter, QP). Chẳng hạn đối với lượng tử hóa vô hướng, các hệ số chia tỷ lệ khác nhau có thể được áp dụng để thực hiện lượng tử hóa mịn hơn hoặc thô hơn. Các kích thước bậc lượng tử hóa nhỏ hơn tương ứng với lượng tử hóa mịn hơn, trong đó các kích thước bậc lượng tử hóa lớn hơn tương ứng với lượng tử hóa thô hơn. Kích thước bậc lượng tử hóa áp dụng được có thể được chỉ báo bởi QP. QP có thể chẳng hạn là chỉ số đối với tập hợp định trước của các kích thước bậc lượng tử hóa có thể áp dụng. Chẳng hạn, các QP nhỏ có thể tương ứng với lượng tử hóa mịn (các kích thước bậc lượng tử hóa nhỏ) và các QP lớn có thể tương ứng với lượng tử hóa thô (các kích thước bậc lượng tử hóa lớn) hoặc ngược lại. Hoạt động lượng tử hóa có thể bao gồm phép chia cho kích thước bậc lượng tử hóa và khử lượng tử hóa đảo và/hoặc tương ứng, chẳng hạn, bởi khối lượng tử hóa đảo 210, có thể bao gồm phép nhân với kích thước bậc lượng tử hóa. Các phương án thực hiện theo một số chuẩn, chẳng hạn, HEVC, có thể được tạo cấu hình để sử dụng QP để xác định kích thước bậc lượng tử hóa. Nói chung, kích thước bậc lượng tử hóa có thể được tính toán dựa trên QP nhờ sử dụng phép xấp xỉ dấu phẩy cố định của phương trình bao gồm phép chia. Các hệ số chia tỷ lệ bổ sung có thể được đưa vào để lượng tử hóa và khử lượng tử hóa để phục hồi chuẩn của khối dư, mà có thể được chỉnh sửa do phép chia tỷ lệ được sử dụng trong phép xấp xỉ dấu phẩy cố định của phương trình đối với kích thước bậc lượng tử hóa và QP. Theo một phương án thực hiện ví dụ, việc chia tỷ lệ của phép biến đổi ngược và khử lượng tử hóa có thể được kết hợp. Theo cách khác, các bảng lượng tử hóa được tùy chỉnh có thể được sử dụng và được báo hiệu từ bộ mã hóa đến bộ giải mã, chẳng hạn, trong dòng bit. Lượng tử hóa là toán tử tổn hao, trong đó tổn hao tăng với các kích thước bậc lượng tử hóa tăng.

Theo các phương án thực hiện, bộ mã hóa video 20 (khối lượng tử hóa 208 tương ứng) có thể được tạo cấu hình để xuất ra các QP, chẳng hạn, trực tiếp

hoặc được mã hóa qua khối mã hóa entropy 270, sao cho, chẳng hạn, bộ giải mã video 30 có thể nhận và áp dụng các QP để giải mã.

Lượng tử hóa đảo

Khối lượng tử hóa đảo 210 được tạo cấu hình để áp dụng lượng tử hóa đảo của khối lượng tử hóa 208 trên các hệ số được lượng tử hóa để thu được các hệ số được khử lượng tử hóa 211, chẳng hạn, bằng cách áp dụng phép đảo của sơ đồ lượng tử hóa được áp dụng bằng khối lượng tử hóa 208 dựa trên hoặc sử dụng kích thước bậc lượng tử hóa giống như khối lượng tử hóa 208. Các hệ số được khử lượng tử hóa 211 cũng có thể được gọi là các hệ số dư được khử lượng tử hóa 211 và tương ứng - Mặc dù thường không giống hệt các hệ số biến đổi do tổn hao bằng cách lượng tử hóa – với các hệ số biến đổi 207.

Biến đổi ngược

Khối xử lý biến đổi ngược 212 được tạo cấu hình để áp dụng biến đổi ngược của phép biến đổi được áp dụng bằng khối xử lý biến đổi 206, chẳng hạn, DCT ngược hoặc DST ngược hoặc các phép biến đổi ngược khác, để thu được khối dư được tái tạo 213 (hoặc các hệ số được khử lượng tử hóa tương ứng 213) trong miền mẫu. Khối dư được tái tạo 213 cũng có thể được gọi là khối biến đổi 213.

Tái tạo

Khối tái tạo 214 (chẳng hạn, bộ cộng 214) được tạo cấu hình để bổ sung khối biến đổi 213 (tức là, khối dư được tái tạo 213) vào khối dự báo 265 để thu được khối được tái tạo 215 trong miền mẫu, chẳng hạn, bằng cách bổ sung - theo từng mẫu - các giá trị mẫu của khối dư được tái tạo 213 và các giá trị mẫu của khối dự báo 265.

Lọc

Khối lọc vòng 220 (hoặc viết tắt là “bộ lọc vòng” 220), được tạo cấu hình để lọc khối được tái tạo 215 để thu được khối được lọc 221, hoặc nói chung, để

lọc các mẫu được tái tạo để thu được các mẫu được lọc. Khối lọc vòng, chẳng hạn, được tạo cấu hình để làm mượt các phép dịch chuyển pixel, hoặc ngược lại cải thiện chất lượng video. Khối lọc vòng 220 có thể bao gồm một hoặc nhiều bộ lọc vòng chẳng hạn bộ lọc khử chặn, bộ lọc độ lệch mẫu thích ứng (sample-adaptive offset, SAO) hoặc một hoặc nhiều bộ lọc khác, chẳng hạn, bộ lọc song phương, bộ lọc vòng thích ứng (adaptive loop filter, ALF), bộ lọc làm nét, bộ lọc làm mượt hoặc bộ lọc phối hợp, hoặc tổ hợp bất kỳ của nó. Mặc dù khối lọc vòng 220 được thể hiện trên Fig.2 như là bộ lọc trong vòng, trong các cấu hình khác, khối lọc vòng 220 có thể được triển khai làm bộ lọc vòng hậu. Khối được lọc 221 cũng có thể được gọi là khối được tái tạo được lọc 221.

Theo các phương án thực hiện, bộ mã hóa video 20 (khối lọc vòng 220 tương ứng) có thể được tạo cấu hình để xuất ra các tham số lọc vòng (chẳng hạn thông tin SAO), chẳng hạn, trực tiếp hoặc được mã hóa qua khối mã hóa entropy 270, sao cho, chẳng hạn, bộ giải mã 30 có thể nhận và áp dụng các tham số lọc vòng tương tự hoặc các khối lọc vòng tương ứng để giải mã.

Bộ đệm ảnh được giải mã

DPB 230 có thể là bộ nhớ mà lưu trữ các ảnh tham chiếu, hoặc nói chung dữ liệu ảnh tham chiếu, để mã hóa dữ liệu video bởi bộ mã hóa video 20. DPB 230 có thể được tạo bằng bộ phận bất kỳ trong các bộ nhớ, chẳng hạn bộ nhớ truy nhập ngẫu nhiên (dynamic random access memory, DRAM), bao gồm DRAM đồng bộ (synchronous DRAM, SDRAM), RAM trở kháng (resistive RAM, RRAM), hoặc các loại bộ nhớ khác. DPB 230 có thể được tạo cấu hình để lưu trữ một hoặc nhiều khối được lọc 221. DPB 230 có thể còn được tạo cấu hình để lưu trữ các khối được lọc trước đó khác, chẳng hạn, các khối được lọc và được tái tạo trước đó 221, của cùng ảnh hiện tại hoặc của các ảnh khác nhau, chẳng hạn, các ảnh được tái tạo trước đó, và có thể tạo các ảnh được tái tạo toàn phần trước đó, tức là, được giải mã (và các mẫu và các khối tham chiếu tương ứng) và/hoặc ảnh hiện tại được tái tạo một phần (và các mẫu và các khối tham chiếu tương ứng), chẳng hạn để dự báo ngoài. DPB 230 có thể cũng được tạo cấu hình để lưu trữ một hoặc nhiều khối được tái tạo không được lọc 215,

hoặc nói chung các mẫu được tái tạo không được lọc, chẳng hạn, nếu khối được tái tạo 215 không được lọc bởi khối lọc vòng 220, hoặc phiên bản được xử lý khác bất kỳ của các khối hoặc các mẫu được tái tạo.

Chọn chế độ (phân vùng và dự báo)

Khối chọn chế độ 260 bao gồm khối phân vùng 262, khối dự báo ngoài 244 và khối dự báo trong 254, và được tạo cấu hình để nhận hoặc thu được dữ liệu ảnh ban đầu, chẳng hạn, khối ban đầu 203 (khối hiện tại 203 của ảnh hiện tại 17), và dữ liệu ảnh được tái tạo, chẳng hạn, các mẫu được tái tạo không được lọc và/hoặc được lọc hoặc các khối của cùng ảnh (hiện tại) và/hoặc từ một hoặc nhiều ảnh được giải mã trước đó, chẳng hạn, từ DPB 230 hoặc các bộ đệm khác (chẳng hạn, bộ đệm dòng, không được thể hiện trên hình vẽ). Dữ liệu ảnh được tái tạo được sử dụng làm dữ liệu ảnh tham chiếu để dự báo, chẳng hạn, dự báo ngoài hoặc dự báo trong, để thu được khối dự báo 265 hoặc bộ dự báo 265.

Khối chọn chế độ 260 có thể được tạo cấu hình để xác định hoặc lựa chọn phân vùng cho chế độ dự báo khối hiện tại (bao gồm không phân vùng) và chế độ dự báo (chẳng hạn, chế độ dự báo trong hoặc ngoài) và tạo khối dự báo 265 tương ứng, được sử dụng để tính toán khối dư 205 và để tái tạo khối được tái tạo 215.

Theo các phương án thực hiện, khối chọn chế độ 260 có thể được tạo cấu hình để lựa chọn chế độ phân vùng và dự báo (chẳng hạn, từ các chế độ được hỗ trợ hoặc khả dụng cho khối chọn chế độ 260), tạo so khớp tốt nhất hoặc nói cách khác phần dư nhỏ nhất (phần dư nhỏ nhất nghĩa là nén tốt hơn để truyền hoặc lưu trữ), hoặc phụ tải báo hiệu nhỏ nhất (phụ tải báo hiệu nhỏ nhất nghĩa là nén tốt hơn để truyền hoặc lưu trữ), hoặc xem xét hoặc cân bằng cả hai. Khối chọn chế độ 260 có thể được tạo cấu hình để xác định chế độ phân vùng và dự báo dựa trên tối ưu hóa méo dạng tốc độ (rate distortion optimization, RDO), tức là, lựa chọn chế độ dự báo mà tạo méo dạng tốc độ nhỏ nhất. Các cụm từ giống như “tốt nhất”, “nhỏ nhất”, “tối ưu”, v.v. trong ngữ cảnh này không nhất thiết đề cập đến toàn bộ “tốt nhất”, “nhỏ nhất”, “tối ưu”, v.v. nhưng cũng có thể đề cập đến hoàn thành tiêu chí chấm dứt hoặc lựa chọn giống như giá trị vượt

quá hoặc nằm dưới ngưỡng hoặc các giới hạn khác tiềm năng dẫn đến “lựa chọn tối ưu phụ” nhưng giảm độ phức tạp và thời gian xử lý.

Nói cách khác, khối phân vùng 262 có thể được tạo cấu hình để phân vùng khối 203 thành các phân vùng khối nhỏ hơn hoặc các khối phụ (lại tạo các khối), chẳng hạn, sử dụng lặp lại phân vùng cây tứ phân (quad-tree, QT), phân vùng cây nhị phân (binary tree, BT) hoặc phân vùng cây tam phân (triple-tree, TT) hoặc tổ hợp bất kỳ của nó, và để thực hiện, chẳng hạn, dự báo cho mỗi trong số các phân vùng khối hoặc các khối phụ, trong đó chọn chế độ bao gồm chọn cấu trúc cây của khối được phân vùng 203 và các chế độ dự báo được áp dụng cho mỗi trong số các phân vùng khối hoặc các khối phụ.

Dưới đây quá trình phân vùng (chẳng hạn, bởi khối phân vùng 260) và quá trình dự báo (bởi khối dự báo ngoài 244 và khối dự báo trong 254) được thực hiện bởi bộ mã hóa video 20 lấy làm ví dụ sẽ được giải thích chi tiết hơn.

Phân vùng

Khối phân vùng 262 có thể phân vùng (hoặc tách) khối hiện tại 203 thành các phân vùng nhỏ hơn, chẳng hạn, các khối nhỏ hơn có kích thước vuông hoặc chữ nhật. Các khối nhỏ hơn này (cũng có thể được gọi là các khối phụ) có thể được phân vùng tiếp thành các phân vùng thậm chí nhỏ hơn. Điều này cũng được gọi là phân vùng cây hoặc phân vùng cây phân cấp, trong đó khối gốc, chẳng hạn, ở mức gốc cây 0 (mức phân cấp 0, chiều sâu 0), có thể được phân vùng đệ quy, chẳng hạn, được phân vùng thành hai hoặc nhiều khối của mức cây thấp hơn tiếp theo, chẳng hạn, các nút ở mức cây 1 (mức phân cấp 1, chiều sâu 1), trong đó các khối này có thể lại được phân vùng thành hai hoặc nhiều khối của mức thấp hơn tiếp theo, chẳng hạn, mức cây 2 (mức phân cấp 2, chiều sâu 2), v.v. cho đến khi phân vùng kết thúc, chẳng hạn, do đạt đến tiêu chí kết thúc, chẳng hạn, đạt đến chiều sâu cây lớn nhất hoặc kích thước khối nhỏ nhất. Các khối không được phân vùng tiếp cũng được gọi là các khối lá hoặc các nút lá của cây. Cây sử dụng phân vùng thành hai phân vùng được gọi là cây nhị phân (binary tree, BT), cây sử dụng phân vùng thành ba phân vùng được gọi là

cây tam phân (ternary tree, TT), và cây sử dụng phân vùng thành bốn phân vùng được gọi là cây tứ phân (quad tree, QT).

Như được nêu trước đó, cụm từ “khối” như được sử dụng ở đây có thể là phần, cụ thể là phần hình vuông hoặc chữ nhật, của ảnh. Chẳng hạn, dựa vào HEVC và VVC, khối có thể là hoặc tương ứng với đơn vị cây mã (coding tree unit, CTU), đơn vị mã (coding unit, CU), đơn vị dự báo (prediction unit, PU), và đơn vị biến đổi (transform unit, TU) và/hoặc với các khối tương ứng, chẳng hạn, khối cây mã (coding tree block, CTB), khối mã (coding block, CB), khối biến đổi (transform block, TB) hoặc khối dự báo (prediction block, PB).

Chẳng hạn, CTU có thể là hoặc bao gồm CTB của các mẫu độ sáng, hai CTB tương ứng của các mẫu sắc độ của ảnh có ba mảng mẫu, hoặc CTB của các mẫu của ảnh đơn sắc hoặc ảnh được tạo mã nhờ sử dụng ba mặt phẳng màu riêng rẽ và các cấu trúc cú pháp được sử dụng để tạo mã các mẫu. Một cách tương ứng, CTB có thể là khối $N \times N$ của các mẫu của một số giá trị của N sao cho phép chia của thành phần thành các CTB là phân vùng. CU có thể là hoặc bao gồm CB của các mẫu độ sáng, hai CB của các mẫu sắc độ của ảnh có ba mảng mẫu, hoặc CB của các mẫu của ảnh đơn sắc hoặc ảnh được tạo mã nhờ sử dụng ba mặt phẳng màu riêng rẽ và các cấu trúc cú pháp được sử dụng để tạo mã các mẫu. Một cách tương ứng, CB có thể là khối $M \times N$ của các mẫu đối với một số giá trị của M và N sao cho phép chia của CTB thành các CB là phân vùng.

Theo các phương án thực hiện, chẳng hạn, theo HEVC, CTU có thể được tách thành các CU nhờ sử dụng cấu trúc QT được ký hiệu là cây mã. Quyết định liệu có tạo mã khu vực ảnh hay không nhờ sử dụng dự báo ngoài ảnh (thời gian) hoặc trong ảnh (không gian) được thực hiện ở mức CU. Mỗi CU có thể được tách tiếp thành một, hai hoặc bốn PU theo loại tách PU. Bên trong một PU, quá trình dự báo tương tự được áp dụng và thông tin liên quan được truyền đến bộ giải mã dựa trên PU. Sau khi nhận được khối dư bằng cách áp dụng quy trình dự báo dựa trên loại tách PU, CU có thể được phân vùng thành các TU theo cấu trúc QT khác giống như cây mã cho CU.

Theo các phương án thực hiện, chẳng hạn, theo chuẩn tạo mã video mới nhất đang được phát triển, được gọi là VVC, phân vùng cây tứ phân và cây nhị phân (Quad-tree and binary tree, QTBT) kết hợp chẳng hạn được sử dụng để phân vùng CB. Trong cấu trúc khối QTBT, CU có thể có hình vuông hoặc hình chữ nhật. Chẳng hạn, CTU được phân vùng trước bởi cấu trúc QT. Các nút lá QT được phân vùng tiếp bởi cấu trúc BT hoặc TT. Các nút lá cây phân vùng được gọi là các CU, và phân đoạn đó được sử dụng để xử lý dự báo và phân vùng mà không phân vùng thêm. Điều này nghĩa là CU, PU và TU có cùng kích thước khối trong cấu trúc CU QTBT. Một cách song song, nhiều phân vùng, chẳng hạn, phân vùng cây tam phân có thể được sử dụng cùng với cấu trúc khối QTBT.

Trong một ví dụ, khối chọn chế độ 260 của bộ mã hóa video 20 có thể được tạo cấu hình để thực hiện tổ hợp bất kỳ của các kỹ thuật phân vùng được nêu ở đây.

Như được nêu trên, bộ mã hóa video 20 được tạo cấu hình để xác định hoặc lựa chọn chế độ dự báo tốt nhất hoặc tối ưu từ tập hợp của các chế độ dự báo (chẳng hạn, được định trước). Tập hợp chế độ dự báo có thể bao gồm, chẳng hạn, các chế độ dự báo trong và/hoặc các chế độ dự báo ngoài.

Dự báo trong

Tập hợp các chế độ dự báo trong có thể bao gồm 35 chế độ dự báo trong khác nhau, chẳng hạn, các chế độ vô hướng như chế độ DC (hoặc trung bình) và chế độ phẳng, hoặc chế độ định hướng, chẳng hạn, như được định nghĩa trong HEVC, hoặc có thể bao gồm 67 chế độ dự báo trong khác nhau, chẳng hạn, các chế độ vô hướng như chế độ DC (hoặc trung bình) và chế độ phẳng, hoặc chế độ định hướng, chẳng hạn, như được định nghĩa đối với VVC.

Khối dự báo trong 254 được tạo cấu hình để sử dụng các mẫu được tái tạo của các khối lân cận của cùng ảnh để tạo khối dự báo trong 265 theo chế độ dự báo trong của tập hợp các chế độ dự báo trong.

Khối dự báo trong 254 (hoặc nói chung khối chọn chế độ 260) còn được tạo cấu hình để xuất các tham số dự báo trong (hoặc nói chung thông tin chỉ báo

chế độ dự báo trong được chọn cho khối) ra khối mã hóa entropy 270 ở dạng phần tử cú pháp 266 để bao gồm vào dữ liệu ảnh được mã hóa 21, sao cho, chẳng hạn, bộ giải mã video 30 có thể nhận và sử dụng các tham số dự báo để giải mã.

Dự báo ngoài

Tập hợp các chế độ dự báo ngoài (hoặc khả thi) tùy thuộc vào các ảnh tham chiếu khả dụng (tức là, ít nhất các ảnh được giải mã một phần trước đó, chẳng hạn, được lưu trữ trong DBP 230) và các tham số dự báo ngoài khác, chẳng hạn, liệu toàn bộ ảnh tham chiếu hoặc chỉ một phần, chẳng hạn, khu vực cửa sổ tìm kiếm xung quanh khu vực của khối hiện tại, của ảnh tham chiếu được sử dụng để tìm kiếm khối tham chiếu so khớp tốt nhất, và/hoặc chẳng hạn, liệu có áp dụng nội suy pixel hay không, chẳng hạn, nội suy bán/nửa pel và/hoặc nội suy 1/4 pel.

Bên cạnh các chế độ dự báo nêu trên, chế độ nhảy và/hoặc chế độ trực tiếp có thể được áp dụng.

Khối dự báo ngoài 244 có thể bao gồm khối ước tính chuyển động (motion estimation, ME) và khối bù chuyển động (motion compensation, MC) (đều không được thể hiện trên Fig.2). Khối ME có thể được tạo cấu hình để nhận hoặc thu được khối ảnh 203 (khối ảnh hiện tại 203 của ảnh hiện tại 17) và ảnh được giải mã 231, hoặc ít nhất một hoặc nhiều khối được tái tạo trước đó, chẳng hạn, các khối được tái tạo của một hoặc nhiều ảnh khác nhau được giải mã trước đó 231, để ước tính chuyển động. Chẳng hạn, chuỗi video có thể bao gồm ảnh hiện tại và ảnh được giải mã trước đó 231, hoặc nói cách khác, ảnh hiện tại và ảnh được giải mã trước đó 231 có thể là một phần hoặc tạo chuỗi ảnh tạo chuỗi video.

Bộ mã hóa 20 có thể, chẳng hạn, được tạo cấu hình để lựa chọn khối tham chiếu từ các khối tham chiếu của các ảnh giống nhau hoặc khác nhau của các ảnh khác nhau và tạo ảnh tham chiếu (hoặc chỉ số ảnh tham chiếu) và/hoặc độ lệch (độ lệch không gian) giữa vị trí (các tọa độ x, y) của khối tham chiếu và vị

trí của khối hiện tại dưới dạng các tham số dự báo ngoài với khối ME. Độ lệch này cũng được gọi là vector chuyển động (motion vector, MV).

Khối MC được tạo cấu hình để thu được, chẳng hạn, nhận, tham số dự báo ngoài và để thực hiện dự báo ngoài dựa trên hoặc sử dụng tham số dự báo ngoài để thu được khối dự báo ngoài 265. Việc bù chuyển động, được thực hiện bởi khối MC, có thể bao gồm việc nạp hoặc tạo khối dự báo dựa trên MV được xác định bởi ước tính chuyển động, có thể thực hiện các phép nội suy ở độ chính xác pixel phụ. Việc lọc nội suy có thể tạo các mẫu pixel bổ sung từ các mẫu pixel đã biết, do vậy một cách tiềm năng tăng số lượng khối dự báo ứng viên mà có thể được sử dụng để tạo mã khối ảnh. Khi nhận MV đối với PU của khối ảnh hiện tại, khối MC có thể định vị khối dự báo mà MV chỉ đến trong một trong các danh sách ảnh tham chiếu.

Khối MC có thể cũng tạo các phần tử cú pháp được liên kết với các khối và các lát video để sử dụng bởi bộ giải mã video 30 khi giải mã các khối ảnh của lát video. Bên cạnh các lát và các phần tử cú pháp tương ứng, các nhóm phiên và/hoặc các phiên và các phần tử cú pháp tương ứng có thể được tạo hoặc được sử dụng.

Tạo mã entropy

Khối mã hóa entropy 270 được tạo cấu hình để áp dụng, chẳng hạn, thuật toán hoặc sơ đồ mã hóa entropy (chẳng hạn, sơ đồ tạo mã có độ dài biến thiên (variable length coding, VLC), sơ đồ VLC có ngữ cảnh thích ứng (context adaptive VLC scheme, CAVLC), sơ đồ tạo mã số học, nhị phân hóa, tạo mã số học nhị phân ngữ cảnh thích ứng (context adaptive binary arithmetic coding, CABAC), CABAC dựa trên cú pháp (syntax-based context-adaptive binary arithmetic coding, SBAC), tạo mã entropy phân vùng khoảng xác suất (probability interval partitioning entropy, PIPE) hoặc phương pháp hoặc kỹ thuật mã hóa entropy khác) hoặc đi vòng (không nén) trên các hệ số được lượng tử hóa 209, các tham số dự báo ngoài, các tham số dự báo trong, các tham số lọc vòng và/hoặc các phần tử cú pháp khác để thu được dữ liệu ảnh được mã hóa 21 có thể được xuất ra qua đầu ra 272, chẳng hạn, ở dạng dòng bit

được mã hóa 21, sao cho, chẳng hạn, bộ giải mã video 30 có thể nhận và sử dụng các tham số để giải mã. Dòng bit được mã hóa 21 có thể được truyền đến bộ giải mã video 30, hoặc được lưu trữ trong bộ nhớ để truyền hoặc truy xuất sau này bởi bộ giải mã video 30.

Các biến thể cấu trúc khác của bộ mã hóa video 20 có thể được sử dụng để mã hóa dòng video. Chẳng hạn, bộ mã hóa không dựa trên biến đổi 20 có thể lượng tử hóa tín hiệu dư trực tiếp mà không khối xử lý biến đổi 206 đối với các khối hoặc các khung cụ thể. Theo triển khai khác, bộ mã hóa 20 có thể có khối lượng tử hóa 208 và khối lượng tử hóa đảo 210 được kết hợp thành một khối.

Bộ giải mã và phương pháp giải mã

Fig.3 thể hiện ví dụ của bộ giải mã video 30 được tạo cấu hình để thực hiện các kỹ thuật của sáng chế. Bộ giải mã video 30 được tạo cấu hình để nhận dữ liệu ảnh được mã hóa 21 (chẳng hạn, dòng bit được mã hóa 21), chẳng hạn, được mã hóa bởi bộ mã hóa 20, để thu được ảnh được giải mã 331. Dữ liệu ảnh được mã hóa hoặc dòng bit bao gồm thông tin để giải mã dữ liệu ảnh được mã hóa, chẳng hạn, dữ liệu là các khối ảnh của lát video được mã hóa (và/hoặc các nhóm phiên hoặc các phiên) và các phần tử cú pháp được liên kết.

Ở ví dụ trên Fig.3, bộ giải mã 30 bao gồm khối giải mã entropy 304, khối lượng tử hóa ngược 310, khối xử lý biến đổi ngược 312, khối tái tạo 314 (chẳng hạn, bộ cộng 314), khối lọc vòng 320, DBP 330, khối áp dụng chế độ 360, khối dự báo ngoài 344 và khối dự báo trong 354. Khối dự báo ngoài 344 có thể là hoặc bao gồm khối MC. Bộ giải mã video 30 có thể, trong một số ví dụ, thực hiện bước giải mã nói chung nghịch đảo bước mã hóa được mô tả dựa vào bộ mã hóa video 100 từ Fig.2.

Như được giải thích dựa vào bộ mã hóa 20, khối lượng tử hóa đảo 210, khối xử lý biến đổi ngược 212, khối tái tạo 214 khối lọc vòng 220, DPB 230, khối dự báo ngoài 344 và khối dự báo trong 354 cũng được gọi là tạo “bộ giải mã cài sẵn” của bộ mã hóa video 20. Do đó, khối lượng tử hóa đảo 310 có thể có chức năng giống như khối lượng tử hóa đảo 110, khối xử lý biến đổi ngược 312 có thể có chức năng giống như khối xử lý biến đổi ngược 212, khối tái tạo 314

có thể có chức năng giống như khối tái tạo 214, khối lọc vòng 320 có thể có chức năng giống như khối lọc vòng 220, và DPB 330 có thể có chức năng giống như DPB 230. Do vậy, các giải thích dành cho các khối riêng rẽ và các chức năng của bộ mã hóa video 20 áp dụng tương ứng cho các khối và các chức năng tương ứng của bộ giải mã video 30.

Giải mã entropy

Khối giải mã entropy 304 được tạo cấu hình để phân tách dòng bit 21 (hoặc nói chung dữ liệu ảnh được mã hóa 21) và thực hiện, chẳng hạn, giải mã entropy cho dữ liệu ảnh được mã hóa 21 để thu được, chẳng hạn, các hệ số được lượng tử hóa 309 và/hoặc các tham số mã được giải mã (không được thể hiện trên Fig.3), chẳng hạn, bất kỳ hoặc tất cả các tham số dự báo ngoài (chẳng hạn, chỉ số ảnh tham chiếu và MV), tham số dự báo trong (chẳng hạn, chế độ hoặc chỉ số dự báo trong), các tham số biến đổi, các QP, các tham số lọc vòng, và/hoặc các phần tử cú pháp khác. Khối giải mã entropy 304 có thể được tạo cấu hình để áp dụng các thuật toán hoặc các sơ đồ giải mã tương ứng với các sơ đồ mã hóa như được mô tả dựa vào khối mã hóa entropy 270 của bộ mã hóa 20. Khối giải mã entropy 304 có thể còn được tạo cấu hình để cấp các tham số dự báo ngoài, tham số dự báo trong và/hoặc các phần tử cú pháp khác cho khối áp dụng chế độ 360 và các tham số khác cho các khối khác của bộ giải mã 30. Bộ giải mã video 30 có thể nhận các phần tử cú pháp ở mức lát video và/hoặc mức khối video. Bên cạnh các lát và các phần tử cú pháp tương ứng, các nhóm phiên và/hoặc các phiên và các phần tử cú pháp tương ứng có thể được nhận và/hoặc được sử dụng.

Lượng tử hóa đảo

Khối lượng tử hóa đảo 310 có thể được tạo cấu hình để nhận các QP (hoặc nói chung thông tin liên quan đến lượng tử hóa đảo) và các hệ số được lượng tử hóa từ dữ liệu ảnh được mã hóa 21 (chẳng hạn, bằng cách phân tách và/hoặc giải mã, chẳng hạn, bởi khối giải mã entropy 304) và để áp dụng dựa trên Các QP lượng tử hóa đảo trên các hệ số được lượng tử hóa được giải mã 309 để thu

được các hệ số được khử lượng tử hóa 311, cũng có thể được gọi là các hệ số biến đổi 311. Quá trình lượng tử hóa đảo có thể bao gồm sử dụng QP được xác định bởi bộ mã hóa video 20 cho mỗi khối video trong lát video (hoặc phiên hoặc nhóm phiên) để xác định mức độ lượng tử hóa và, tương tự, mức độ lượng tử hóa đảo cần được áp dụng.

Biến đổi ngược

Khối xử lý biến đổi ngược 312 có thể được tạo cấu hình để nhận các hệ số được khử lượng tử hóa 311, cũng được gọi là các hệ số biến đổi 311, và để áp dụng biến đổi cho các hệ số được khử lượng tử hóa 311 để thu được các khối dư được tái tạo 213 trong miền mẫu. Các khối dư được tái tạo 213 cũng có thể được gọi là các khối biến đổi 313. Phép biến đổi có thể là biến đổi ngược, chẳng hạn, DCT ngược, DST ngược, biến đổi số nguyên ngược, hoặc quá trình biến đổi ngược tương tự về mặt khái niệm. Khối xử lý biến đổi ngược 312 có thể còn được tạo cấu hình để nhận các tham số biến đổi hoặc thông tin tương ứng từ dữ liệu ảnh được mã hóa 21 (chẳng hạn, bằng cách phân tách và/hoặc giải mã, chẳng hạn, bởi khối giải mã entropy 304) để xác định phép biến đổi được áp dụng cho các hệ số được khử lượng tử hóa 311.

Tái tạo

Khối tái tạo 314 (chẳng hạn, bộ cộng 314) có thể được tạo cấu hình để bổ sung khối dư được tái tạo 313, vào khối dự báo 365 để thu được khối được tái tạo 315 trong miền mẫu, chẳng hạn, bằng cách bổ sung các giá trị mẫu của khối dư được tái tạo 313 và các giá trị mẫu của khối dự báo 365.

Lọc

Khối lọc vòng 320 (trong vòng mã hoặc sau vòng mã) được tạo cấu hình để lọc khối được tái tạo 315 để thu được khối được lọc 321, chẳng hạn, để làm mượt các phép dịch chuyển pixel, hoặc ngược lại cải thiện chất lượng video. Khối lọc vòng 320 có thể bao gồm một hoặc nhiều bộ lọc vòng chẳng hạn bộ lọc khử chặn, bộ lọc độ lệch mẫu thích ứng (sample-adaptive offset, SAO) hoặc

một hoặc nhiều bộ lọc khác, chẳng hạn, bộ lọc song phương, ALF, bộ lọc làm nét, bộ lọc làm mượt hoặc bộ lọc phối hợp, hoặc tổ hợp bất kỳ của nó. Mặc dù khối lọc vòng 320 được thể hiện trên Fig.3 như là bộ lọc trong vòng, trong các cấu hình khác, khối lọc vòng 320 có thể được triển khai làm bộ lọc vòng hậu.

Bộ đệm ảnh được giải mã

Sau đó, các khối video được giải mã 321 của ảnh được lưu trữ trong DPB 330, mà lưu trữ các ảnh được giải mã 331 làm các ảnh tham chiếu để bù chuyển động tiếp theo cho các ảnh khác nhau và/hoặc cho hiển thị đầu ra tương ứng.

Bộ giải mã 30 được tạo cấu hình để xuất ra ảnh được giải mã 311, chẳng hạn, qua đầu ra 312, để hiển thị đến người dùng.

Dự báo

Khối dự báo ngoài 344 có thể có chức năng giống như khối dự báo ngoài 244 (cụ thể với khối MC) và khối dự báo trong 354 có thể có chức năng giống như khối dự báo ngoài 254, và thực hiện các quyết định tách hoặc phân vùng và dự báo dựa trên các tham số dự báo và/hoặc phân vùng hoặc thông tin tương ứng được nhận từ dữ liệu ảnh được mã hóa 21 (chẳng hạn, bằng cách phân tách và/hoặc giải mã, chẳng hạn, bởi khối giải mã entropy 304). Khối áp dụng chế độ 360 có thể được tạo cấu hình để thực hiện dự báo (dự báo trong hoặc ngoài) theo khối dựa trên các ảnh được tái tạo, các khối hoặc các mẫu tương ứng (được lọc hoặc không được lọc) để thu được khối dự báo 365.

Khi lát video được tạo mã dưới dạng lát được mã hóa trong (I), khối dự báo trong 354 của khối áp dụng chế độ 360 được tạo cấu hình để tạo khối dự báo 365 đối với khối ảnh của lát video hiện tại dựa trên chế độ dự báo trong và dữ liệu được báo hiệu từ các khối được giải mã trước đó của ảnh hiện tại. Khi ảnh video được tạo mã dưới dạng lát được mã hóa ngoài (tức là, B, hoặc P), khối dự báo ngoài 344 (chẳng hạn, khối MC) của khối áp dụng chế độ 360 được tạo cấu hình để tạo các khối dự báo 365 đối với khối video của lát video hiện tại dựa trên các MV và các phần tử cú pháp khác được nhận từ khối giải mã

entropy 304. Để dự báo ngoài, các khối dự báo có thể được tạo từ một trong các ảnh tham chiếu trong một trong các danh sách ảnh tham chiếu. Bộ giải mã video 30 có thể tạo các danh sách khung tham chiếu, danh sách 0 và danh sách 1, nhờ sử dụng các kỹ thuật tạo mặc định dựa trên các ảnh tham chiếu được lưu trữ trong DPB 330. Kỹ thuật giống hoặc tương tự có thể được áp dụng cho hoặc theo các phương án thực hiện nhờ sử dụng các nhóm phiên (chẳng hạn, các nhóm phiên video) và/hoặc các phiên (chẳng hạn, các phiên video) bên cạnh hoặc theo cách khác với các lát (chẳng hạn, các lát video), chẳng hạn, video có thể được tạo mã nhờ sử dụng các nhóm phiên và/hoặc các phiên I, P hoặc B.

Khối áp dụng chế độ 360 được tạo cấu hình để xác định thông tin dự báo đối với khối video của lát video hiện tại bằng cách phân tách các MV hoặc thông tin liên quan và các phần tử cú pháp khác, và sử dụng thông tin dự báo để tạo các khối dự báo cho khối video hiện tại được giải mã. Chẳng hạn, khối áp dụng chế độ 360 sử dụng một số phần tử cú pháp được nhận để xác định chế độ dự báo (chẳng hạn, dự báo trong hoặc ngoài) được sử dụng để tạo mã các khối video của lát video, loại lát dự báo ngoài (chẳng hạn, Lát B, lát P, hoặc lát GPB), thông tin tạo cho một hoặc nhiều danh sách ảnh tham chiếu cho lát, các MV cho mỗi khối video được mã hóa ngoài của lát, trạng thái dự báo ngoài cho mỗi khối video được mã hóa ngoài của lát, và thông tin khác để giải mã các khối video trong lát video hiện tại. Kỹ thuật tương tự hoặc giống có thể được áp dụng cho hoặc theo các phương án thực hiện nhờ sử dụng các nhóm phiên (chẳng hạn, các nhóm phiên video) và/hoặc các phiên (chẳng hạn, các phiên video) bên cạnh hoặc theo cách khác với các lát (chẳng hạn, các lát video), chẳng hạn, video có thể được tạo mã nhờ sử dụng các nhóm phiên I, P hoặc B và/hoặc các phiên.

Theo các phương án thực hiện, bộ giải mã video 30 như được thể hiện trên Fig.3 có thể được tạo cấu hình để phân vùng và/hoặc giải mã ảnh nhờ sử dụng các lát (cũng được gọi là các lát video), trong đó ảnh có thể được phân vùng thành hoặc được giải mã nhờ sử dụng một hoặc nhiều lát (thường không trùng lặp), và mỗi lát có thể bao gồm một hoặc nhiều khối (chẳng hạn, các CTU).

Các phương án thực hiện bộ giải mã video 30 như được thể hiện trên Fig.3 có thể được tạo cấu hình để phân vùng và/hoặc giải mã ảnh nhờ sử dụng các nhóm phiên (cũng được gọi là các nhóm phiên video) và/hoặc các phiên (cũng được gọi là các phiên video), trong đó ảnh có thể được phân vùng thành hoặc được giải mã nhờ sử dụng một hoặc nhiều nhóm phiên (thường không trùng lặp), và mỗi nhóm phiên có thể bao gồm, chẳng hạn, một hoặc nhiều khối (chẳng hạn, các CTU) hoặc một hoặc nhiều phiên, trong đó mỗi phiên, chẳng hạn, có thể có hình chữ nhật và có thể bao gồm một hoặc nhiều khối (chẳng hạn, các CTU), chẳng hạn, các khối hoàn chỉnh hoặc một phần.

Các biến thể khác của bộ giải mã video 30 có thể được sử dụng để giải mã dữ liệu ảnh được mã hóa 21. Chẳng hạn, bộ giải mã 30 có thể tạo dòng video đầu ra mà không có khối lọc vòng 320. Chẳng hạn, bộ giải mã không dựa trên biến đổi 30 có thể lượng tử hóa ngược tín hiệu dư trực tiếp mà không khối xử lý biến đổi ngược 312 đối với các khối hoặc các khung cụ thể. Theo triển khai khác, bộ giải mã video 30 có thể có khối lượng tử hóa đảo 310 và khối xử lý biến đổi ngược 312 được kết hợp thành một khối.

Cần hiểu rằng, trong bộ mã hóa 20 và bộ giải mã 30, kết quả xử lý của bước hiện tại có thể được xử lý tiếp và sau đó được xuất ra bước tiếp theo. Chẳng hạn, sau khi lọc nội suy, dẫn xuất MV hoặc lọc vòng, hoạt động khác, chẳng hạn xén tia hoặc dịch chuyển, có thể được thực hiện trên kết quả xử lý lọc nội suy, dẫn xuất MV hoặc lọc vòng.

Cần lưu ý rằng các hoạt động khác có thể được áp dụng cho các MV được dẫn xuất của khối hiện tại (bao gồm nhưng không bị giới hạn ở các MV điểm điều khiển của chế độ afin, các MV khối phụ trong afin, các chế độ phẳng, các chế độ ATMVP, các MV thời gian, và v.v.). Chẳng hạn, giá trị của MV bị giới hạn ở khoảng định trước theo bit biểu diễn của nó. Nếu bit biểu diễn của MV là bitDepth, thì khoảng là $-2^{(\text{bitDepth}-1)} \sim 2^{(\text{bitDepth}-1)}-1$, trong đó “^” nghĩa là nâng lũy thừa. Chẳng hạn, nếu bitDepth được gán bằng 16, khoảng này là từ -32768 đến 32767; nếu bitDepth được gán bằng 18, khoảng này là từ -131072 đến 131071. Chẳng hạn, giá trị của MV được dẫn xuất (chẳng hạn, các MV của bốn khối phụ 4x4 trong một khối 8x8) bị giới hạn sao cho hiệu số lớn nhất giữa

các phần số nguyên của bốn MV khối phụ 4x4 không lớn hơn N pixel, chẳng hạn không lớn hơn 1 pixel. Ở đây đề xuất hai phương pháp giới hạn MV theo bitDepth.

Phương pháp 1: loại bỏ bit có trọng số lớn nhất (most significant bit, MSB) tràn bằng các phép toán sau

$$ux = (mvx + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (1)$$

$$mvx = (ux \geq 2^{\text{bitDepth}-1}) ? (ux - 2^{\text{bitDepth}}) : ux \quad (2)$$

$$uy = (mvy + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (3)$$

$$mvy = (uy \geq 2^{\text{bitDepth}-1}) ? (uy - 2^{\text{bitDepth}}) : uy \quad (4)$$

trong đó mvx là thành phần ngang của MV của khối hoặc khối phụ ảnh, mvy là thành phần dọc của MV của khối hoặc khối phụ ảnh, và ux và uy chỉ báo giá trị trung gian;

Chẳng hạn, nếu giá trị của mvx = -32769, sau khi áp dụng công thức (1) và (2), giá trị thu được = 32767. Trong hệ thống máy tính, các số thập phân được lưu trữ dưới dạng bù 2. Bù 2 của -32769 bằng 1,0111,1111,1111,1111 (17 bit), sau đó MSB bị loại bỏ, do vậy bù 2 thu được bằng 0111,1111,1111,1111 (số thập phân = 32767), điều này tương tự như đầu ra bằng cách áp dụng công thức (1) và (2).

$$ux = (mvp_x + mvd_x + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (5)$$

$$mvx = (ux \geq 2^{\text{bitDepth}-1}) ? (ux - 2^{\text{bitDepth}}) : ux \quad (6)$$

$$uy = (mvp_y + mvd_y + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (7)$$

$$mvy = (uy \geq 2^{\text{bitDepth}-1}) ? (uy - 2^{\text{bitDepth}}) : uy \quad (8)$$

Các phép toán có thể được áp dụng trong khi cộng.mvp và mvd, như được thể hiện trên công thức từ (5) đến (8).

Phương pháp 2: loại bỏ MSB tràn bằng cách xén tỉa giá trị

$$vx = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1} - 1, vx)$$

$$vy = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1} - 1, vy)$$

trong đó vx là thành phần ngang của MV của khối hoặc khối phụ ảnh, vy là thành phần dọc của MV của khối hoặc khối phụ ảnh; x, y và z lần lượt tương ứng với ba giá trị đầu vào của quá trình xén MV, và định nghĩa về hàm Clip3 như sau:

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{ngược lại} \end{cases}$$

Fig.4 là sơ đồ của thiết bị tạo mã video 400 theo phương án thực hiện sáng chế. Thiết bị tạo mã video 400 thích hợp để thực hiện các phương án thực hiện được bộc lộ như được nêu ở đây. Theo phương án thực hiện, thiết bị tạo mã video 400 có thể là bộ giải mã chẳng hạn bộ giải mã video 30 trên Fig.1A hoặc bộ mã hóa chẳng hạn bộ mã hóa video 20 trên Fig.1A.

Thiết bị tạo mã video 400 bao gồm các cổng vào 410 và các bộ nhận (Rx) 420 để nhận dữ liệu; bộ xử lý, đơn vị logic, hoặc khối xử lý trung tâm (central processing unit, CPU) 430 để xử lý dữ liệu; các bộ truyền (Tx) 440 và các cổng ra 450 để truyền dữ liệu; và bộ nhớ 460 để lưu trữ dữ liệu. Thiết bị tạo mã video 400 có thể cũng bao gồm các thành phần quang sang điện (optical-to-electrical, OE) và các thành phần điện sang quang (electrical-to-optical, EO) được ghép nối với các cổng vào 410, các bộ nhận 420, các bộ truyền 440, và các cổng ra 450 để xuất hoặc nhập các tín hiệu quang hoặc điện.

Bộ xử lý 430 được triển khai bằng phần cứng và phần mềm. Bộ xử lý 430 có thể được triển khai dưới dạng một hoặc nhiều chip, lõi CPU (chẳng hạn, dưới dạng bộ xử lý nhiều lõi), FPGA, ASIC, và DSP. Bộ xử lý 430 kết nối với các cổng vào 410, các bộ nhận 420, các bộ truyền 440, các cổng ra 450, và bộ nhớ 460. Bộ xử lý 430 bao gồm môđun tạo mã 470. Môđun tạo mã 470 thực hiện các phương án thực hiện được bộc lộ nêu trên. Chẳng hạn, môđun tạo mã 470 thực hiện, xử lý, chuẩn bị, hoặc tạo các hoạt động tạo mã khác nhau. Do vậy, việc bao gồm môđun tạo mã 470 cải thiện đáng kể chức năng của thiết bị tạo mã video 400 và thực hiện biến đổi thiết bị tạo mã video 400 sang trạng thái khác. Theo cách khác, môđun tạo mã 470 được triển khai dưới dạng các lệnh được lưu trữ trong bộ nhớ 460 và được thực thi bởi bộ xử lý 430.

Bộ nhớ 460 có thể bao gồm một hoặc nhiều đĩa, ổ băng, và ổ trạng thái rắn và có thể được sử dụng làm bộ lưu trữ dữ liệu tràn, để lưu trữ các chương trình khi các chương trình này được chọn để thực thi, và để lưu trữ các lệnh và dữ liệu được đọc khi thực thi chương trình. Bộ nhớ 460 có thể là, chẳng hạn, khả

biến và/hoặc bất biến và có thể là bộ nhớ chỉ đọc (read-only memory, ROM), bộ nhớ truy nhập ngẫu nhiên (random access memory, RAM), bộ nhớ có nội dung đánh địa chỉ được tam phân (ternary content-addressable memory, TCAM), và/hoặc RAM tĩnh (static random-access memory, SRAM).

Fig.5 là sơ đồ khối được đơn giản hóa của thiết bị 500 có thể được sử dụng dưới dạng một trong hoặc cả hai thiết bị nguồn 12 và thiết bị đích 14 từ Fig.1 theo phương án thực hiện lấy làm ví dụ.

Bộ xử lý 502 trong thiết bị 500 có thể là CPU. Theo cách khác, bộ xử lý 502 có thể là loại thiết bị khác bất kỳ, hoặc nhiều thiết bị, có thể thao tác hoặc xử lý thông tin hiện có hoặc được phát triển sau này. Mặc dù các triển khai được bộc lộ có thể được thực hiện với một bộ xử lý như được thể hiện trên hình vẽ, chẳng hạn, bộ xử lý 502, các ưu điểm về tốc độ và hiệu suất có thể đạt được nhờ sử dụng nhiều hơn một bộ xử lý.

Bộ nhớ 504 trong thiết bị 500 có thể là ROM hoặc RAM theo triển khai. Loại bộ nhớ thích hợp khác bất kỳ có thể được sử dụng làm bộ nhớ 504. Bộ nhớ 504 có thể bao gồm mã và dữ liệu 506 được truy nhập bởi bộ xử lý 502 nhờ sử dụng buýt 512. Bộ nhớ 504 có thể còn bao gồm hệ điều hành 508 và các chương trình ứng dụng 510, các chương trình ứng dụng 510 bao gồm ít nhất một chương trình cho phép bộ xử lý 502 để thực hiện các phương pháp được mô tả ở đây. Chẳng hạn, các chương trình ứng dụng 510 có thể bao gồm các ứng dụng từ 1 đến N, mà còn bao gồm ứng dụng tạo mã video mà thực hiện các phương pháp được mô tả ở đây.

Thiết bị 500 cũng có thể bao gồm một hoặc nhiều bộ phận đầu ra, chẳng hạn màn hiển thị 518. Màn hiển thị 518 có thể là, trong một ví dụ, màn hiển thị cảm ứng chạm mà kết hợp màn hiển thị với phần tử cảm ứng chạm mà có thể vận hành để cảm nhận các lần chạm vào. Màn hiển thị 518 có thể được ghép nối với bộ xử lý 502 qua buýt 512.

Mặc dù được mô tả ở đây dưới dạng một buýt, song buýt 512 của thiết bị 500 có thể gồm nhiều buýt. Ngoài ra, bộ lưu trữ thứ cấp 514 có thể được ghép nối trực tiếp với các thành phần khác của thiết bị 500 hoặc có thể được truy nhập qua mạng và có thể bao gồm một khối tích hợp chẳng hạn thẻ nhớ hoặc

nhiều khối chẳng hạn nhiều thể nhớ. Do vậy, thiết bị 500 có thể được triển khai trong các cấu hình.

Khối dự báo ngoài 244 có thể bao gồm khối ME và khối MC (không được thể hiện trên Fig.2). Khối ME được tạo cấu hình để nhận hoặc thu được khối ảnh 203 (khối ảnh hiện tại 203 của ảnh hiện tại 201) và ảnh được giải mã 331, hoặc ít nhất một hoặc nhiều khối được tái tạo trước đó, chẳng hạn, các khối được tái tạo của một hoặc nhiều ảnh khác nhau được giải mã trước đó 331, để ước tính chuyển động. Chẳng hạn, chuỗi video có thể bao gồm ảnh hiện tại và ảnh được giải mã trước đó 331, hoặc nói cách khác, ảnh hiện tại và ảnh được giải mã trước đó 331 có thể là một phần hoặc tạo chuỗi ảnh tạo chuỗi video. Bộ mã hóa 200 có thể, chẳng hạn,, được tạo cấu hình để lựa chọn khối tham chiếu từ các khối tham chiếu của các ảnh giống nhau hoặc khác nhau của các ảnh khác nhau và tạo ảnh tham chiếu (hoặc chỉ số ảnh tham chiếu, ...) và/hoặc độ lệch (độ lệch không gian) giữa vị trí (các tọa độ x, y) của khối tham chiếu và vị trí của khối hiện tại dưới dạng các tham số dự báo ngoài với khối ME (không được thể hiện trên Fig.2). Độ lệch này cũng được gọi là MV. Hợp nhất là công cụ ước tính chuyển động quan trọng được sử dụng trong HEVC và kế thừa cho VVC.

Để thực hiện ước tính hợp nhất, danh sách ứng viên hợp nhất được tạo, mỗi ứng viên trong danh sách hợp nhất bao gồm dữ liệu chuyển động, các dữ liệu chuyển động này bao gồm thông tin mà chỉ báo liệu một hoặc hai danh sách ảnh tham chiếu được sử dụng, và cũng bao gồm thông tin khác chẳng hạn chỉ số tham chiếu và MV cho mỗi danh sách. Trong ví dụ, danh sách ứng viên hợp nhất được tạo dựa trên các ứng viên sau:

- a. lên đến bốn ứng viên hợp nhất không gian mà được suy ra từ năm khối lân cận không gian;
- b. một ứng viên hợp nhất thời gian được suy ra từ hai khối cùng vị trí thời gian;
- c. các ứng viên hợp nhất bổ sung bao gồm các ứng viên dự báo kép kết hợp và các ứng viên MV số 0.

Các ứng viên thứ nhất trong danh sách ứng viên hợp nhất là các lân cận không gian. Lên tới bốn ứng viên được đưa vào danh sách hợp nhất bằng cách

tuần tự kiểm tra A1, B1, B0, A0 và B2, theo thứ tự đó, theo phần bên phải của Fig.6.

Bên cạnh kiểm tra liệu CB có khả dụng và chứa thông tin chuyển động, một số phép kiểm tra dư bổ sung được thực hiện trước khi xem xét tất cả dữ liệu chuyển động của CB làm ứng viên hợp nhất. Các phép kiểm tra dư này có thể được phân thành hai phân loại: a. tránh dữ liệu chuyển động dư trong danh sách; b. ngăn ngừa hợp nhất hai phân vùng có thể được biểu diễn bằng các phương tiện khác mà tạo cú pháp dư.

Trong ví dụ, N là số lượng ứng viên hợp nhất không gian, kiểm tra dư hoàn chỉnh sử dụng bao gồm $(N \cdot (N-1))/2$ phép so sánh dữ liệu chuyển động. Trong trường hợp của năm ứng viên hợp nhất không gian tiềm năng, mười phép so sánh dữ liệu chuyển động sẽ cần để đảm bảo rằng tất cả các ứng viên trong danh sách hợp nhất có dữ liệu chuyển động khác nhau. Trong khi phát triển HEVC, các phép kiểm tra cho dữ liệu chuyển động dư đã được giảm xuống tập con theo cách hiệu suất mã được duy trì, trong khi logic so sánh được giảm đáng kể. Cuối cùng, không nhiều hơn hai phép so sánh được thực hiện trên ứng viên dẫn đến tổng cộng năm phép so sánh. Với thứ tự {A1, B1, B0, A0, B2}, B0 chỉ kiểm tra B1, A0 chỉ kiểm tra A1, và B2 chỉ kiểm tra A1 và B1. Trong ví dụ kiểm tra dư phân vùng, PU dưới của phân vùng $2N \times N$ được hợp nhất với PU trên bằng cách chọn ứng viên B1. Điều này sẽ dẫn đến một CU với hai PU có cùng dữ liệu chuyển động, mà có thể được báo hiệu bằng nhau dưới dạng CU $2N \times 2N$. Nói chung, phép kiểm tra này áp dụng cho tất cả PU thứ hai của các phân vùng bất đối xứng và chữ nhật $2N \times N$, $2N \times nU$, $2N \times nD$, $N \times 2N$, $nR \times 2N$ và $nL \times 2N$. Lưu ý rằng đối với các ứng viên hợp nhất không gian, chỉ các phép kiểm tra dư được thực hiện và dữ liệu chuyển động được sao chép từ các khối ứng viên. Do vậy, ở đây không cần đo tỷ lệ MV.

Việc dẫn xuất các MV đối với ứng viên hợp nhất thời gian giống như đối với dự báo MV thời gian (Temporal Motion Vector Prediction, TMVP). Do ứng viên hợp nhất bao gồm tất cả dữ liệu chuyển động và TMVP chỉ là một MV (nghĩa là, trong hợp nhất thông thường, chỉ một ứng viên TMVP lớn nhất được phép), việc dẫn xuất dữ liệu chuyển động tùy thuộc vào loại lát. Đối với

các lát dự báo kép, TMVP được dẫn xuất cho mỗi danh sách ảnh tham chiếu. Tùy thuộc vào tính khả dụng của TMVP đối với mỗi danh sách, loại dự báo được gán bằng dự báo kép hoặc với danh sách mà TMVP khả dụng cho nó. Tất cả các chỉ số ảnh tham chiếu liên kết được gán bằng 0. Đối với các lát dự báo đơn, TMVP cho danh sách 0 được dẫn xuất cùng với chỉ số ảnh tham chiếu = 0.

Khi ít nhất một TMVP khả dụng và ứng viên hợp nhất thời gian được thêm vào danh sách, kiểm tra dư không được thực hiện. Điều này khiến việc tạo danh sách hợp nhất độc lập với ảnh cùng vị trí mà cải thiện độ bền lỗi. Xem xét trường hợp trong đó ứng viên hợp nhất thời gian sẽ dư và do vậy không được bao gồm trong danh sách ứng viên hợp nhất. Trong trường hợp ảnh cùng vị trí thất lạc, bộ giải mã không thể suy ra các ứng viên thời gian và do vậy không thể kiểm tra liệu nó sẽ dư thừa hay không. Việc đánh chỉ số tất cả các ứng viên tiếp sẽ bị ảnh hưởng bởi điều này.

Để phân tách các nguyên nhân độ bền, số lượng ứng viên trong danh sách ứng viên hợp nhất được cố định. Sau khi các ứng viên hợp nhất không gian và thời gian đã được thêm vào, danh sách có thể không đầy (số lượng ứng viên trong danh sách ứng viên hợp nhất nhỏ hơn số cố định). Để bù lại tổn hao hiệu suất mã đi cùng với báo hiệu chỉ số danh sách độ dài không thích ứng, các ứng viên bổ sung được tạo. Tùy thuộc vào loại lát, lên đến hai loại ứng viên được sử dụng để điền đầy đủ danh sách: a. Các ứng viên dự báo kép kết hợp; b. Các ứng viên MV số 0.

Trong các lát dự báo kép, các ứng viên bổ sung có thể được tạo dựa trên các ứng viên hiện tại, bằng cách kết hợp danh sách ảnh tham chiếu 0 dữ liệu chuyển động của một ứng viên với và danh sách 1 dữ liệu chuyển động của ứng viên khác. Điều này được thực hiện bằng cách sao chép Δx_0 , Δy_0 , Δt_0 từ một ứng viên, chẳng hạn, ứng viên thứ nhất, và Δx_1 , Δy_1 , Δt_1 từ ứng viên khác, chẳng hạn, ứng viên thứ hai. Các tổ hợp khác nhau được định trước và được nêu trong Bảng 1.

Bảng 1

Thứ tự kết hợp	0	1	2	3	4	5	6	7	8	9	10	11
$\Delta x_0, \Delta y_0, \Delta t_0$, từ Cand	0	1	0	2	1	2	0	3	1	3	2	3
$\Delta x_1, \Delta y_1, \Delta t_1$, từ Cand	1	0	2	0	2	1	3	0	3	1	3	2

Khi danh sách vẫn không đầy sau khi bổ sung các ứng viên dự báo kép kết hợp, hoặc đối với các lát dự báo đơn, các ứng viên MV số 0 được bổ sung. Các ứng viên MV số 0 có một MV dịch chuyển số 0 cho các lát dự báo đơn và hai MV dịch chuyển số 0 cho các lát dự báo kép. Các chỉ số tham chiếu được gán bằng 0, và được tăng lên 1 cho mỗi ứng viên bổ sung, cho đến khi số lượng lớn nhất các chỉ số tham chiếu được đạt đến. Nếu các ứng viên bổ sung vẫn cần được bổ sung, chỉ số tham chiếu = 0 được sử dụng để tạo các ứng viên bổ sung. Đối với các ứng viên bổ sung, các phép kiểm tra dư không được thực hiện do việc bỏ qua các phép kiểm tra này không gây tổn hao hiệu suất tạo mã.

Đối với mỗi PU được tạo mã trong chế độ dự báo ngoài ảnh, `merge_flag` được sử dụng để chỉ báo rằng việc hợp nhất khối được sử dụng để suy ra dữ liệu chuyển động hay không. `merge_idx` được sử dụng để xác định ứng viên trong danh sách hợp nhất tạo dữ liệu chuyển động cần để dự báo bù chuyển động (motion compensated prediction, MCP). Bên cạnh báo hiệu mức PU, số lượng ứng viên trong danh sách hợp nhất được báo hiệu trong tiêu đề lát. Trong ví dụ, giá trị mặc định = 5, được biểu diễn dưới dạng hiệu số với 5 (`five_minus_max_num_merge_cand`). Liên quan quá trình tạo danh sách ứng viên hợp nhất, toàn bộ quá trình vẫn như cũ mặc dù nó kết thúc sau khi danh sách chứa số lượng lớn nhất ứng viên hợp nhất. Trong thiết kế ban đầu, giá trị lớn nhất để tạo mã chỉ số hợp nhất được tạo bởi số lượng ứng viên thời gian và không gian khả dụng trong danh sách. Khi chỉ hai ứng viên khả dụng, chỉ số có thể được tạo mã hiệu quả dưới dạng cờ. Để phân tách chỉ số hợp nhất, toàn bộ danh sách ứng viên hợp nhất phải được tạo để biết số lượng ứng viên thực.

Việc áp dụng hợp nhất khối trong HEVC được kết hợp với chế độ nhảy. Chế độ nhảy được sử dụng cho khối để chỉ báo rằng dữ liệu chuyển động cho khối được suy luận, thay cho báo hiệu tường minh trong dòng bit và rằng phần dư

cho khối = 0, tức là, không hệ số biến đổi nào được truyền. Trong HEVC, bắt đầu của mỗi CU trong lát dự báo ngoài ảnh, skip_flag được báo hiệu ngụ ý như sau: a. CU chỉ chứa một PU (loại phân vùng $2N \times 2N$); b. chế độ hợp nhất được sử dụng để suy ra dữ liệu chuyển động (merge_flag = 1); c. dữ liệu dư không xuất hiện trong dòng bit.

Công cụ ước tính khác được đưa vào HEVC và kế thừa trong VVC được gọi là dự báo MV cải tiến (Advanced Motion Vector Prediction, AMVP). Trong chế độ AMVP, các MV được tạo mã liên quan đến các thành phần ngang (x) và dọc (y) làm hiệu số cho bộ dự báo MV (motion vector predictor, MVP). Việc tính toán thành phần hiệu số MV (motion vector difference, MVD) được thể hiện dưới dạng $MVD_x = MV_x - MVP_x$, $MVD_y = MV_y - MVP_y$.

Các MV của khối hiện tại thường được tương quan với các MV của các khối lân cận trong ảnh hiện tại hoặc trong các ảnh được tạo mã sớm hơn. Do các khối lân cận có thể tương ứng với cùng đối tượng chuyển động có chuyển động tương tự, và chuyển động của đối tượng không thể bị thay đổi đột ngột theo thời gian. Kết quả là, việc sử dụng các MV trong các khối lân cận làm các bộ dự báo sẽ giảm kích thước MVD được báo hiệu. Các MVP thường được suy ra từ các MV được giải mã từ các khối lân cận không gian hoặc từ các khối lân cận theo thời gian trong ảnh cùng vị trí. Trong HEVC, cách thức suy diễn ngầm MVP được thay bằng công cụ được biết đến như là cạnh tranh MV, báo hiệu tường minh MVP từ danh sách các MVP, được sử dụng để suy ra MV. Cấu trúc khối QT tạo mã biến thiên trong HEVC có thể dẫn đến một khối có vài khối lân cận với các MV làm các ứng viên MVP tiềm năng. Lấy lân cận bên trái làm ví dụ, trong trường hợp mà khối dự báo độ sáng 64×64 có thể có 16 khối dự báo độ sáng 8×4 về bên trái, khi khối cây mã độ sáng 64×64 không được phân tách thêm và khối bên trái được tách đến độ sâu lớn nhất. AMVP được đưa vào để thay đổi cạnh tranh MV để xem xét cấu trúc khối linh hoạt này. Trong khi phát triển HEVC, thiết kế AMVP ban đầu được đơn giản hóa đáng kể để cân bằng tốt giữa hiệu suất tạo mã và thiết kế thân thiện với triển khai.

Thiết kế ban đầu của AMVP bao gồm năm MVP từ ba lớp khác nhau của các bộ dự báo: ba MV từ các lân cận không gian, median của ba bộ dự báo

không gian và MV được chia tỷ lệ từ khối lân cận về thời gian được đặt cùng vị trí. Ngoài ra, danh sách bộ dự báo được thay đổi bằng cách sắp lại thứ tự để đặt bộ dự báo chuyển động có xác suất lớn nhất trong vị trí thứ nhất, và bằng cách loại bỏ các ứng viên dư thừa để đảm bảo phụ tải báo hiệu nhỏ nhất. Các thí nghiệm vết cặn trong suốt quá trình chuẩn hóa đã nghiên cứu cách thức có thể giảm độ phức tạp của dự báo MV này, và sơ đồ báo hiệu mà không hi sinh quá nhiều hiệu suất tạo mã. Điều này dẫn đến việc đơn giản hóa đáng kể thiết kế AMVP chẳng hạn loại bỏ bộ dự báo median, giảm số lượng ứng viên trong danh sách từ 5 xuống 2, cố định thứ tự ứng viên trong danh sách và giảm số lượng phép kiểm tra dư. Thiết kế tạo danh sách ứng viên AMVP bao gồm hai ứng viên MVP sau:

- lên đến hai MVP ứng viên không gian mà được suy ra từ năm khối lân cận không gian;
- một MVP ứng viên thời gian được suy ra từ hai khối cùng vị trí thời gian khi cả hai MVP ứng viên không gian không có sẵn hoặc chúng giống hệt nhau;
- các MV số 0 khi các ứng viên không gian, thời gian hoặc cả hai không có sẵn.

Trong phần mô tả ứng viên, luồng tiến trình dẫn xuất đối với hai ứng viên không gian A và B được mô tả trên Fig 13. Đối với ứng viên A, dữ liệu chuyển động từ hai khối A0 và A1 ở góc trái bên dưới được xem xét trong cách thức hai bước. Ở bước thứ hai, kiểm tra liệu khối bất kỳ trong các khối ứng viên có chứa chỉ số tham chiếu bằng chỉ số tham chiếu của khối hiện tại. MV thứ nhất sẽ được lấy làm ứng viên A. Khi tất cả chỉ số tham chiếu từ A0 và A1 đang chỉ đến ảnh tham chiếu khác với chỉ số tham chiếu của khối hiện tại, MV liên kết không thể được sử dụng như hiện tại. Do vậy, trong bước thứ hai, các MV cần được chia tỷ lệ theo các khoảng thời gian giữa ảnh tham chiếu ứng viên và ảnh tham chiếu hiện tại. Khoảng cách thời gian được biểu diễn liên quan đến hiệu số giữa các giá trị đếm thứ tự ảnh (picture order count, POC), các giá trị POC được sử dụng để định nghĩa thứ tự hiển thị của các ảnh.

Đối với ứng viên B, các ứng viên từ B0 đến B2 được kiểm tra tuần tự theo cách giống như A0 và A1. Tuy nhiên, bước thứ hai được thực hiện khi các khối A0 và A1 không chứa thông tin chuyển động bất kỳ, tức là, không khả dụng

hoặc được tạo mã nhờ sử dụng dự báo trong ảnh. Sau đó, ứng viên A được thiết lập bằng ứng viên không được chia tỷ lệ B, nếu được tìm thấy, và ứng viên B được thiết lập bằng biến thể thứ hai có chia tỷ lệ hoặc không chia tỷ lệ của ứng viên B. Bước thứ hai có thể được kết thúc khi vẫn có thể có các ứng viên không được chia tỷ lệ tiềm năng, bước thứ hai tìm kiếm các MV không được chia tỷ lệ cũng như có chia tỷ lệ được suy ra từ các ứng viên từ B0 đến B2. Nói chung, thiết kế này cho phép xử lý A0 và A1 độc lập từ B0, B1, và B2. Việc dẫn xuất B nên nhận thức tính khả dụng của cả A0 lẫn A1, để tìm kiếm MV được chia tỷ lệ hoặc không được chia tỷ lệ bổ sung được suy ra từ B0 đến B2. Sự lệ thuộc này giảm đáng kể các hoạt động chia tỷ lệ MV phức tạp đối với ứng viên B. Việc giảm số lượng các phép chia tỷ lệ MV nghĩa là giảm độ phức tạp đáng kể trong trong trình dẫn xuất MV.

Trong quá trình chọn ứng viên thời gian. Có thể thấy từ Fig.6 rằng các MV từ các khối lân cận không gian sang bên trái và bên trên khối hiện tại được xem là các ứng viên MVP không gian. Điều này có thể được giải thích như là việc các khối sang bên phải và bên dưới khối hiện tại vẫn chưa được giải mã và do vậy, dữ liệu chuyển động của chúng là không khả dụng. Do khối nằm cùng vị trí là ảnh tham chiếu đã được giải mã, cũng có thể xem xét dữ liệu chuyển động từ khối ở cùng vị trí, từ các khối sang bên phải của khối cùng vị trí hoặc từ các khối bên dưới. Trên HEVC, khối sang dưới bên phải và ở tâm của khối hiện tại đã được xác định là thích hợp nhất để có bộ dự báo MV thời gian tốt (temporal motion vector predictor, TMVP). Các ứng viên này được minh họa trên Fig.6, trong đó C0 là lân cận dưới bên phải và C1 là khối trung tâm. Dữ liệu chuyển động của C0 được xem xét trước, nếu dữ liệu chuyển động của C0 không khả dụng, dữ liệu chuyển động từ khối ứng viên cùng vị trí ở tâm được sử dụng để suy ra ứng viên MVP thời gian C. Dữ liệu chuyển động của C0 cũng được xem là không khả dụng khi PU liên kết thuộc CTU ngoài hàng CTU hiện tại. Điều này giảm tối đa các yêu cầu băng thông bộ nhớ để lưu trữ dữ liệu chuyển động cùng vị trí. Ngược lại các ứng viên MVP không gian, trong đó các MV có thể đề cập đến cùng ảnh tham chiếu, phép chia tỷ lệ MV là bắt buộc đối với TMVP.

Cả quá trình tạo danh sách hợp nhất lẫn quá trình tạo AMVP, HMVP đến danh sách hợp nhất hoặc AMVP. Các ứng viên hợp nhất HMVP được bổ sung vào danh sách hợp nhất/AMVP sau MVP không gian và TMVP. Trong phương pháp này, thông tin chuyển động của khối được tạo mã trước đó được lưu trữ trong bảng và được sử dụng làm MVP đối với CU hiện tại. Bảng có nhiều ứng viên HMVP được duy trì trong suốt quá trình mã hóa/giải mã. Bảng được thiết lập lại (được làm trống) khi gặp hàng CTU mới. Bất kể khi nào có CU được tạo mã ngoài không phải khối phụ, thông tin chuyển động được liên kết được bổ sung vào mục cuối cùng của bảng như là ứng viên HMVP mới.

Trong VTM4, kích thước bảng HMVP được gán bằng 6, chỉ báo lên đến 6 ứng viên HMVP có thể được thêm vào bảng. Khi chèn ứng viên chuyển động mới vào bảng, quy tắc vào trước ra trước (first-in-first-out, FIFO) có giới hạn được sử dụng trong đó phép kiểm tra dư được áp dụng trước để tìm ra liệu có HMVP giống hệt trong bảng hay không. Nếu có HMVP giống hệt trong bảng, HMVP giống hệt được loại bỏ khỏi bảng và tất cả các ứng viên HMVP về sau được di chuyển lên trước.

Các ứng viên HMVP có thể được sử dụng trong các tiến trình xây dựng danh sách ứng viên hợp nhất/danh sách AMVP. Vài ứng viên HMVP mới nhất trong bảng được kiểm tra theo thứ tự và được đưa vào danh sách ứng viên list sau ứng viên TMVP. Phép kiểm tra dư được áp dụng trên các ứng viên HMVP vào ứng viên hợp nhất không gian hoặc thời gian.

Để giảm số lượng hoạt động kiểm tra dư, các phép đơn giản hóa sau được đưa vào:

Số lượng ứng viên HMPV được sử dụng để tạo danh sách hợp nhất được thiết lập bằng $(N \leq 4) ? M : (8 - N)$, trong đó N chỉ báo số lượng ứng viên hiện có trong danh sách hợp nhất và M chỉ báo số lượng ứng viên HMVP khả dụng trong bảng.

Khi tổng số lượng ứng viên hợp nhất khả dụng bằng các ứng viên hợp nhất được phép lớn nhất - 1, quá trình tạo danh sách ứng viên hợp nhất từ HMVP kết thúc.

Trong phác thảo VVC song song với chế độ ngoài, chế độ IBC được giới thiệu.

IBC là công cụ được đưa vào trong các phần mở rộng HEVC trên SCC. Nó cải thiện đáng kể hiệu suất tạo mã của các tài liệu nội dung màn hình. Do chế độ IBC được triển khai dưới dạng chế độ tạo mã mức khối, quá trình so khớp khối (block matching, BM) được thực hiện ở bộ mã hóa để tìm ra vector khối tối ưu (hoặc MV) cho mỗi CU. Ở đây, MV được sử dụng để chỉ báo dịch chuyển từ khối hiện tại đến khối tham chiếu, đã được tái tạo bên trong ảnh hiện tại. MV độ sáng của CU được tạo mã IBC là có độ chính xác nguyên. MV sắc độ được xén tỉa đến độ chính xác nguyên. Khi được kết hợp với AMVR, chế độ IBC có thể chuyển đổi giữa các độ chính xác MV 1-pel và 4-pel. CU được tạo mã IBC được xử lý như là chế độ dự báo thứ ba khác ngoài chế độ dự báo trong hoặc ngoài.

Để giảm độ phức tạp tiêu thụ bộ nhớ lẫn bộ giải mã, IBC trong VTM4 cho phép phần được tái tạo của khu vực được định trước bao gồm CTU hiện tại cần được sử dụng. Giới hạn này cho phép chế độ IBC được triển khai nhờ sử dụng bộ nhớ trên chip cục bộ để triển khai phần cứng.

Ở phía bộ mã hóa, ước tính chuyển động dựa trên băm được thực hiện cho IBC. Bộ mã hóa thực hiện kiểm tra RD cho các khối có chiều rộng hoặc chiều cao không lớn hơn 16 mẫu độ sáng. Đối với chế độ không hợp nhất, phép tìm kiếm vector khối được thực hiện nhờ sử dụng phép tìm kiếm dựa trên băm trước. Nếu phép tìm kiếm băm không trả về ứng viên hợp lệ, phép tìm kiếm cục bộ dựa trên so khớp khối sẽ được thực hiện.

Trong tìm kiếm dựa trên băm, so khớp khóa băm (CRC 32 bit) giữa khối hiện tại và khối tham chiếu được mở rộng sang tất cả các kích thước khối được phép. Việc tính toán khóa băm cho mỗi vị trí trong ảnh hiện tại dựa trên các khối phụ 4x4. Đối với khối hiện tại có kích thước lớn hơn, khóa băm được xác định để khớp với khóa của khối tham chiếu khi tất cả các khóa băm của tất cả khối phụ 4x4 so khớp các khóa băm trong các vị trí tham chiếu tương ứng. Nếu các khóa băm của nhiều khối tham chiếu được tìm thấy để so khớp với khóa

của khối hiện tại, các chi phí vector khối của mỗi tham chiếu được so khớp được tính toán và vector có chi phí nhỏ nhất được chọn.

Trong tìm kiếm so khớp khối, khoảng tìm kiếm được gán bằng N mẫu sang bên trái và bên trên của khối hiện tại trong CTU hiện tại. Ở phần đầu của CTU, giá trị N được khởi tạo bằng 128 nếu không có ảnh tham chiếu thời gian, hặc được khởi tạo bằng 64 nếu có ít nhất một ảnh tham chiếu thời gian. Tỷ lệ bit băm được định nghĩa là phần trăm mẫu trong CTU đã tìm thấy so khớp nhờ sử dụng tìm kiếm dựa trên hàm băm. Khi mã hóa CTU hiện tại, nếu tỷ lệ bit băm nhỏ hơn 5%, N được giảm đi một nửa.

Ở mức CU, chế độ IBC được báo hiệu bằng cờ và có thể được báo hiệu dưới dạng chế độ AMVP IBC hoặc chế độ nhảy/hợp nhất IBC như sau:

Chế độ nhảy/hợp nhất IBC: chỉ số ứng viên hợp nhất được sử dụng để chỉ báo các vector khối nào trong danh sách từ các khối được tạo mã IBC ứng viên lân cận được sử dụng để dự báo khối hiện tại. Danh sách hợp nhất bao gồm các ứng viên theo cặp, không gian và HMVP.

Chế độ AMVP IBC: hiệu số vector khối được tạo mã theo cách giống như MVD. Phương pháp dự báo vector khối sử dụng hai ứng viên làm các bộ dự báo, một từ lân cận bên trái và một từ lân cận bên trên (nếu được tạo mã IBC). Khi lân cận bất kỳ không khả dụng, vector khối mặc định sẽ được sử dụng làm bộ dự báo. Cờ được báo hiệu để chỉ báo chỉ số bộ dự báo vector khối.

Do IBC đã giới thiệu chế độ nhảy/hợp nhất IBC và chế độ AMVP IBC, có danh sách hợp nhất IBC và danh sách AMVP bổ sung cần được tạo trong VVC Draft 4.0.

Do chỉ các ứng viên không gian (khối lân cận bên trái A1, khối lân cận bên trên B1, khối lân cận dưới bên trái A0, khối lân cận phải bên trên B0 và khối lân cận trên bên trái B2 như được thể hiện trên Fig 6), Các ứng viên HMVP ($H_1 \dots H_k$, k bằng kích thước danh sách HMVP lớn nhất) và các ứng viên theo cặp được sử dụng để tạo danh sách hợp nhất IBC theo thứ tự trong VVC Draft 4.0. Các phép tĩa sau có thể được thực hiện trong khi tạo danh sách hợp nhất IBC:

- Tĩa giữa A1 và B1
- Tĩa giữa A0 và A1

- Tỉa giữa B0 và B1
- Tỉa giữa B2 và A1
- Tỉa giữa B2 và B1
- Tỉa giữa ứng viên HMVP cuối cùng H_k và A1
- Tỉa giữa ứng viên HMVP cuối cùng H_k và B1
- Tỉa giữa ứng viên HMVP cuối cùng thứ hai H_{k-1} và A1
- Tỉa giữa ứng viên HMVP cuối cùng thứ hai H_{k-1} và B1

Quá trình tỉa nghĩa là so sánh liệu hai ứng viên hợp nhất IBC có giống nhau không. Cụ thể hơn, quá trình tỉa sẽ so sánh liệu các vector khối giữa hai ứng viên hợp nhất IBC có giống nhau không.

Tóm lại, số lượng 9 phép tỉa lớn nhất cần thiết để tạo danh sách hợp nhất IBC của khối hiện tại.

Trong bộ mã hóa lẫn bộ giải mã, phép tỉa là làm trễ quá trình tạo danh sách hợp nhất. Do trong mỗi giai đoạn tỉa, điều kiện “nếu” được kiểm tra, quá trình tạo danh sách hợp nhất khác được thực hiện tùy thuộc vào các phép kiểm tra điều kiện “nếu” này. Càng nhiều phép tỉa trong quá trình tạo danh sách ứng viên hợp nhất, quá trình bộ mã hóa lẫn bộ giải mã càng phức tạp. Để giảm độ phức tạp xây dựng danh sách hợp nhất, các giải pháp sau được đưa vào.

Giải pháp 1

Do các khối lân cận không gian bên trái (A1) và bên trên (B1) là quan trọng để dự báo khối hiện tại nhờ sử dụng chế độ IBC. Trong giải pháp 1, phép tỉa khối lân cận không gian giữa A1 và B1 được duy trì, phép tỉa khối lân cận không gian còn lại được loại bỏ. Ứng viên tỉa HMVP được giữ lại. Theo phương án thực hiện, phép tỉa sau có thể được thực hiện trong quá trình tạo danh sách hợp nhất IBC:

- Tỉa giữa A1 và B1
- Tỉa giữa ứng viên HMVP cuối cùng H_k và A1
- Tỉa giữa ứng viên HMVP cuối cùng H_k và B1
- Tỉa giữa ứng viên HMVP cuối cùng thứ hai H_{k-1} và A1

- Tia giữa ứng viên HMVP cuối cùng thứ hai H_{k-1} và B1.

Trong trường hợp này, số lượng lớn nhất phép xén tia đối với việc tạo danh sách ứng viên hợp nhất IBC của khối hiện tại được giảm từ 9 xuống 5. Giải pháp này giảm đáng kể độ phức tạp tạo danh sách hợp nhất IBC cho cả bộ mã hóa lẫn bộ giải mã.

Trong ví dụ, danh sách ứng viên hợp nhất IBC cụ thể được tạo như sau:

Nếu khối lân cận A1 khả dụng và đang sử dụng chế độ IBC, vector khối của khối A1 được chèn vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (A1 không khả dụng hoặc A1 không sử dụng chế độ IBC), vector khối của khối A1 không được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại (ứng viên thứ nhất, không tia).

Nếu khối lân cận B1 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B1 giống như A1. Nếu vector khối của B1 không giống như vector khối của A1, vector khối của khối B1 được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B1 giống như vector khối A1, hoặc B1 không khả dụng hoặc B1 không sử dụng chế độ IBC), vector khối của khối B1 không được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia A1 và B1).

Nếu khối lân cận B0 khả dụng và đang sử dụng chế độ IBC, vector khối của khối B0 được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), vector khối của khối B0 không được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại (không tia).

Nếu khối lân cận A0 khả dụng và đang sử dụng chế độ IBC, vector khối của khối A0 được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (A0 không khả dụng hoặc A0 không sử dụng chế độ IBC), vector khối của khối A0 không được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại (không tia).

Nếu khối lân cận B2 khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn 4, vector khối của khối B2 được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại

(B2 không khả dụng hoặc B2 không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn 4), vector khối của khối B2 không được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại (không tĩa).

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1; và vector khối của H_k không giống như A1 và B1, chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng or H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_k giống như vector khối của A1, hoặc vector khối của H_k giống như vector khối của B1) không chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_k và A1, H_k và B1).

Nếu ứng viên HMVP cuối cùng thứ hai H_{k-1} khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_{k-1} không giống như A1 và B1, chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_{k-1} không khả dụng hoặc H_{k-1} không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_{k-1} giống như vector khối của A1, hoặc vector khối của H_{k-1} giống như vector khối của B1) không chèn vector khối của khối H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_{k-1} và A1, H_{k-1} và B1).

Chèn các ứng viên HMVP còn lại lần lượt, nếu ứng viên khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1 (không tĩa).

Chèn ứng viên theo cặp, kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất (không tĩa).

Giải pháp 2

Do, các khối lân cận không gian bên trái (A1) và bên trên (B1) là quan trọng để dự báo khối hiện tại nhờ sử dụng chế độ IBC. Trong giải pháp 2, khối lân cận không gian A1 và B1 được duy trì để chèn vào ứng viên hợp nhất IBC, các khối lân cận không gian ứng viên còn lại được loại bỏ. Ứng viên tia HMVP được giữ nguyên. Theo phương án thực hiện, các phép tia sau có thể thực hiện trong khi tạo danh sách hợp nhất IBC:

- Tia giữa A1 và B1
- Tia giữa ứng viên HMVP cuối cùng H_k và A1
- Tia giữa ứng viên HMVP cuối cùng H_k và B1
- Tia giữa ứng viên HMVP cuối cùng thứ hai H_{k-1} và A1
- Tia giữa ứng viên HMVP cuối cùng thứ hai H_{k-1} và B1

Trong trường hợp này, số lượng lớn nhất phép xén tia đối với việc tạo danh sách IBC ứng viên hợp nhất của khối hiện tại được giảm từ 9 xuống 5. Giải pháp này giảm đáng kể độ phức tạp tạo IBC danh sách hợp nhất cho cả bộ mã hóa lẫn bộ giải mã.

Trong ví dụ, danh sách IBC ứng viên hợp nhất cụ thể được tạo như sau:

Nếu khối lân cận A1 khả dụng và đang sử dụng chế độ IBC, chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (A1 không khả dụng hoặc A1 không sử dụng chế độ IBC), không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (ứng viên thứ nhất, không tia).

Nếu khối lân cận B1 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B1 giống như A1. Nếu vector khối của B1 không giống như vector khối của A1, chèn vector khối của khối B1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B1 giống như vector khối A1, hoặc B1 không khả dụng hoặc B1 không sử dụng chế độ IBC), không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia A1 và B1).

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_k

không giống như $A1$ và $B1$, chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_k giống như vector khối của $A1$, hoặc vector khối của H_k giống như vector khối của $B1$), không chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_k và $A1$, H_k và $B1$).

Nếu ứng viên HMVP cuối cùng thứ hai H_{k-1} khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_{k-1} không giống như $A1$ và $B1$, chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_{k-1} không khả dụng hoặc H_{k-1} không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_{k-1} giống như vector khối của $A1$, hoặc vector khối của H_{k-1} giống như vector khối của $B1$) không chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_{k-1} và $A1$, H_{k-1} và $B1$).

Chèn các ứng viên HMVP còn lại lần lượt, nếu ứng viên khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1. (không tĩa)

Chèn ứng viên theo cặp, kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất. (không tĩa)

Giải pháp 3

Theo giải pháp 3, tĩa vector khối của khối lân cận không gian được duy trì. Đối với tĩa ứng viên HMVP, vector khối của ứng viên HMVP cuối cùng H_k được tĩa bằng vector khối khối lân cận không gian $A1$ và $B1$. Các phép tĩa ứng viên HMVP còn lại được loại bỏ. Theo phương án thực hiện, phép tĩa sau có thể được thực hiện trong khi tạo danh sách hợp nhất IBC:

- Tĩa giữa $A1$ và $B1$
- Tĩa giữa $A0$ và $A1$

- Tia giữa B0 và B1
- Tia giữa B2 và A1
- Tia giữa B2 và B1
- Tia giữa ứng viên HMVP cuối cùng H_k và A1
- Tia giữa ứng viên HMVP cuối cùng H_k và B1.

Trong trường hợp này, số lượng lớn nhất phép xén tia đối với việc tạo danh sách IBC ứng viên hợp nhất của khối hiện tại được giảm từ 9 xuống 7. Giải pháp này giảm đáng kể độ phức tạp tạo IBC danh sách hợp nhất cho cả bộ mã hóa lẫn bộ giải mã.

Trong ví dụ, danh sách IBC ứng viên hợp nhất cụ thể được tạo như sau:

Nếu khối lân cận A1 khả dụng và đang sử dụng chế độ IBC, chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (A1 không khả dụng hoặc A1 không sử dụng chế độ IBC), không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (ứng viên thứ nhất, không tia).

Nếu khối lân cận B1 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B1 giống như A1. Nếu vector khối của B1 không giống như vector khối của A1, chèn vector khối của khối B1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B1 giống như vector khối A1, hoặc B1 không khả dụng hoặc B1 không sử dụng chế độ IBC), không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia A1 và B1).

Nếu khối lân cận B0 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B0 giống như B1. Nếu vector khối của B0 không giống như vector khối của B1, chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B0 giống như vector khối B1, hoặc B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), không chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia B0 và B1).

Nếu khối lân cận A0 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của A0 giống như A1. Nếu vector khối của A0 không

giống như vector khối của A1, chèn vector khối của khối A0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của A0 giống như vector khối A1, hoặc A0 không khả dụng hoặc A0 không sử dụng chế độ IBC), không chèn vector khối của khối A0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (A0 và A1 tĩa).

Nếu khối lân cận B2 khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách hợp nhất IBC hiện tại nhỏ hơn 4, tĩa (hoặc xác định) liệu vector khối của B2 giống như A1 và liệu vector khối của B2 giống như B1. Nếu vector khối của B2 không giống như vector khối của A1, và vector khối của B2 không giống như B1, chèn vector khối của khối B2 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B2 giống như vector khối A1 hoặc B1, hoặc B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), không chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa B2 và A1 và tĩa B2 và B1).

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_k không giống như A1 và B1, chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_k giống như vector khối của A1, hoặc vector khối của H_k giống như vector khối của B1) không chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_k và A1, H_k và B1).

Chèn các ứng viên HMVP còn lại lần lượt, nếu ứng viên khả dụng và đang sử dụng chế độ IBC và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1 (không tĩa).

Chèn ứng viên theo cặp, kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất (không tĩa).

Giải pháp 4

Theo giải pháp 4, tia vector khối của khối lân cận không gian được duy trì. Đối với phép tia ứng viên HMVP, vector khối của ứng viên HMVP cuối cùng H_k và ứng viên HMVP cuối cùng thứ hai H_{k-1} được tia bằng vector khối khối lân cận không gian A1. Phép tia ứng viên HMVP còn lại được loại bỏ. Theo phương án thực hiện, phép tia sau có thể được thực hiện trong khi tạo danh sách hợp nhất IBC:

- Tia giữa A1 và B1
- Tia giữa A0 và A1
- Tia giữa B0 và B1
- Tia giữa B2 và A1
- Tia giữa B2 và B1
- Tia giữa ứng viên HMVP cuối cùng H_k và A1
- Tia giữa ứng viên HMVP cuối cùng thứ hai H_{k-1} và A1.

Trong trường hợp này, số lượng lớn nhất phép xén tia đối với việc tạo danh sách IBC ứng viên hợp nhất của khối hiện tại được giảm từ 9 xuống 7. Giải pháp này giảm đáng kể độ phức tạp tạo IBC danh sách hợp nhất cho cả bộ mã hóa lẫn bộ giải mã.

Trong ví dụ, danh sách IBC ứng viên hợp nhất cụ thể được tạo như sau:

Nếu khối lân cận A1 khả dụng và đang sử dụng chế độ IBC, chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (A1 không khả dụng hoặc A1 không sử dụng chế độ IBC), không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (ứng viên thứ nhất, không tia).

Nếu khối lân cận B1 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B1 giống như A1. Nếu vector khối của B1 không giống như vector khối của A1, chèn vector khối của khối B1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B1 giống như vector khối A1, hoặc B1 không khả dụng hoặc B1 không sử dụng chế độ IBC), không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia A1 và B1).

Nếu khối lân cận B0 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B0 giống như B1. Nếu vector khối của B0 không giống như vector khối của B1, chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B0 giống như vector khối B1, hoặc B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), không chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia B0 và B1).

Nếu khối lân cận A0 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của A0 giống như A1. Nếu vector khối của A0 không giống như vector khối của A1, chèn vector khối của khối A0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của A0 giống như vector khối A1, hoặc A0 không khả dụng hoặc A0 không sử dụng chế độ IBC), không chèn vector khối của khối A0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia A0 và A1).

Nếu khối lân cận B2 khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách hợp nhất IBC hiện tại nhỏ hơn 4, tia (hoặc xác định) liệu vector khối của B2 giống như A1 và liệu vector khối của B2 giống như B1. Nếu vector khối của B2 không giống như vector khối của A1 và vector khối của B2 không giống như B1, chèn vector khối của khối B2 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B2 giống như vector khối A1 hoặc B1, hoặc B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), không chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia B2 và A1 và tia B2 và B1).

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_k không giống như A1, chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_k giống như vector khối của A1), không chèn vector khối của H_k vào danh sách

ứng viên hợp nhất IBC của khối hiện tại (tĩa H_k và $A1$).

Nếu ứng viên HMVP cuối cùng thứ hai H_{k-1} khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_{k-1} không giống như $A1$, chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_{k-1} không khả dụng hoặc H_{k-1} không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_{k-1} giống như vector khối của $A1$), không chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_{k-1} và $B1$).

Chèn các ứng viên HMVP còn lại lần lượt, nếu ứng viên khả dụng và đang sử dụng chế độ IBC và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1 (không tĩa).

Chèn ứng viên theo cặp, kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất (không tĩa).

Giải pháp 5

Theo giải pháp 5, tĩa vector khối của khối lân cận không gian được duy trì. Đối với phép tĩa ứng viên HMVP, vector khối của ứng viên HMVP cuối cùng H_k và ứng viên HMVP cuối cùng thứ hai H_{k-1} được tĩa bằng vector khối khối lân cận không gian $B1$. Phép tĩa ứng viên HMVP còn lại được loại bỏ. Theo phương án thực hiện, phép tĩa sau có thể được thực hiện trong khi tạo danh sách hợp nhất IBC:

- Tĩa giữa $A1$ và $B1$
- Tĩa giữa $A0$ và $A1$
- Tĩa giữa $B0$ và $B1$
- Tĩa giữa $B2$ và $A1$
- Tĩa giữa $B2$ và $B1$
- Tĩa giữa ứng viên HMVP cuối cùng H_k và $B1$
- Tĩa giữa ứng viên HMVP cuối cùng thứ hai H_{k-1} và $B1$

Trong trường hợp này, số lượng lớn nhất phép xen tĩa đối với việc tạo danh

sách IBC ứng viên hợp nhất của khối hiện tại được giảm từ 9 xuống 7. Giải pháp này giảm đáng kể độ phức tạp tạo IBC danh sách hợp nhất cho cả bộ mã hóa lẫn bộ giải mã.

Trong ví dụ, danh sách IBC ứng viên hợp nhất cụ thể được tạo như sau:

Nếu khối lân cận A1 khả dụng và đang sử dụng chế độ IBC, chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (A1 không khả dụng hoặc A1 không sử dụng chế độ IBC), không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (ứng viên thứ nhất, không tia).

Nếu khối lân cận B1 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B1 giống như A1. Nếu vector khối của B1 không giống như vector khối của A1, chèn vector khối của khối B1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B1 giống như vector khối A1, hoặc B1 không khả dụng hoặc B1 không sử dụng chế độ IBC), không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia A1 và B1).

Nếu khối lân cận B0 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B0 giống như B1. Nếu vector khối của B0 không giống như vector khối của B1, chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B0 giống như vector khối B1, hoặc B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), không chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia B0 và B1).

Nếu khối lân cận A0 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của A0 giống như A1. Nếu vector khối của A0 không giống như vector khối của A1, chèn vector khối của khối A0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của A0 giống như vector khối A1, hoặc A0 không khả dụng hoặc A0 không sử dụng chế độ IBC), không chèn vector khối của khối A0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia A0 và A1).

Nếu khối lân cận B2 khả dụng và đang sử dụng chế độ IBC và kích thước

danhsách hợp nhất IBC hiện tại nhỏ hơn 4, tia (hoặc xác định) liệu vector khối của B2 giống như A1 và liệu vector khối của B2 giống như B1. Nếu vector khối của B2 không giống như vector khối của A1 và vector khối của B2 không giống như B1, chèn vector khối của khối B2 vào danhsách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B2 giống như vector khối A1 or B1, hoặc B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), không chèn vector khối của khối B0 vào danhsách ứng viên hợp nhất IBC của khối hiện tại (tia B2 và A1 và tia B2 và B1).

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC và kích thước danhsách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_k không giống như B1, chèn vector khối của H_k vào danhsách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danhsách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_k giống như vector khối của B1), không chèn vector khối của H_k vào danhsách ứng viên hợp nhất IBC của khối hiện tại (tia H_k và B1).

Nếu ứng viên HMVP cuối cùng thứ hai H_{k-1} khả dụng và đang sử dụng chế độ IBC, và kích thước danhsách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_{k-1} không giống như B1, chèn vector khối của H_{k-1} vào danhsách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_{k-1} không khả dụng hoặc H_{k-1} không sử dụng chế độ IBC hoặc kích thước danhsách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_{k-1} giống như vector khối của B1), không chèn vector khối của H_{k-1} vào danhsách ứng viên hợp nhất IBC của khối hiện tại (tia H_{k-1} và B1).

Chèn các ứng viên HMVP còn lại lần lượt, nếu ứng viên khả dụng và đang sử dụng chế độ IBC và kích thước danhsách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1 (không tia).

Chèn ứng viên theo cặp, kích thước danhsách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất (không tia).

Giải pháp 6

Theo giải pháp 6, trong giải pháp ứng viên cuối cùng của danh sách ứng viên hợp nhất IBC được phép nhờ sử dụng chế độ HMVP, quá trình tia thực hiện tạo danh sách hợp nhất IBC trong giải pháp này không bị thay đổi, nhưng phương pháp này đề cập đến phương pháp tạo danh sách hợp nhất hiệu quả hơn.

Trong ví dụ, danh sách IBC ứng viên hợp nhất cụ thể được tạo như sau:

Nếu khối lân cận A1 khả dụng và đang sử dụng chế độ IBC, chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (A1 không khả dụng hoặc A1 không sử dụng chế độ IBC), không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (ứng viên thứ nhất, không tia).

Nếu khối lân cận B1 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B1 giống như A1. Nếu vector khối của B1 không giống như vector khối của A1, chèn vector khối của khối B1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B1 giống như vector khối A1, hoặc B1 không khả dụng or B1 không sử dụng chế độ IBC), không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia A1 và B1).

Nếu khối lân cận B0 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B0 giống như B1. Nếu vector khối của B0 không giống như vector khối của B1, chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B0 giống như vector khối B1, hoặc B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), không chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia B0 và B1).

Nếu khối lân cận A0 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của A0 giống như A1. Nếu vector khối của A0 không giống như vector khối của A1, chèn vector khối của khối A0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của A0 giống như vector khối A1, hoặc A0 không khả dụng hoặc A0 không sử dụng chế độ IBC),

không chèn vector khối của khối A0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia A0 và A1).

Nếu khối lân cận B2 khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách hợp nhất IBC hiện tại nhỏ hơn 4, tia (hoặc xác định) liệu vector khối của B2 giống như A1 và liệu vector khối của B2 giống như B1. Nếu vector khối của B2 không giống như vector khối của A1 và vector khối của B2 không giống như B1, chèn vector khối của khối B2 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B2 giống như vector khối A1 hoặc B1, hoặc B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), không chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia B2 và A1, tia B2 và B1).

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất, và vector khối của H_k không giống như A1 và B1, chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất, hoặc vector khối của H_k giống như vector khối của A1, hoặc vector khối của H_k giống như vector khối của B1), không chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia H_k và A1, H_k và B1).

Nếu ứng viên HMVP cuối cùng thứ hai H_{k-1} khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất và vector khối của H_{k-1} không giống như A1 và B1, chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_{k-1} không khả dụng hoặc H_{k-1} không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất hoặc vector khối của H_{k-1} giống như vector khối của A1, hoặc vector khối của H_{k-1} giống như vector khối của B1), không chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia H_{k-1} và A1, H_{k-1} và B1).

Chèn các ứng viên HMVP còn lại lần lượt, nếu ứng viên khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất (không tĩa).

Chèn ứng viên theo cặp, kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất (không tĩa).

Giải pháp 7

Theo giải pháp 7, tia vector khối của khối lân cận không gian được duy trì. Đối với tia ứng viên HMVP, vector khối của ứng viên HMVP cuối cùng H_k và ứng viên HMVP cuối cùng thứ hai H_{k-1} được tĩa bằng vector khối của khối lân cận không gian thứ nhất của A1 và B1. Nếu A1 hoặc B1 không nằm trong danh sách hợp nhất IBC, thì không ứng viên HMVP được chứng minh với ứng viên không gian.

Phép tĩa ứng viên HMVP còn lại được loại bỏ. Theo phương án thực hiện, phép tĩa sau là quá trình tĩa trường hợp xấu nhất có thể trong khi tạo danh sách hợp nhất IBC:

- Tĩa giữa A1 và B1
- Tĩa giữa A0 và A1
- Tĩa giữa B0 và B1
- Tĩa giữa B2 và A1
- Tĩa giữa B2 và B1
- Tĩa giữa ứng viên HMVP cuối cùng H_k và A1 hoặc H_k và B1
- Tĩa giữa ứng viên HMVP cuối cùng thứ hai H_{k-1} và A1 hoặc H_{k-1} và B1

Trong trường hợp này, số lượng lớn nhất phép xén tia đối với việc tạo danh sách IBC ứng viên hợp nhất của khối hiện tại được giảm từ 9 xuống 7. Giải pháp này giảm đáng kể độ phức tạp tạo IBC danh sách hợp nhất cho cả bộ mã hóa lẫn bộ giải mã.

Trong ví dụ, danh sách IBC ứng viên hợp nhất cụ thể được tạo như sau:

Nếu khối lân cận A1 khả dụng và đang sử dụng chế độ IBC, chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (A1 không khả dụng hoặc A1 không sử dụng chế độ IBC), không

chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (ứng viên thứ nhất, không tia).

Nếu khối lân cận B1 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B1 giống như A1. Nếu vector khối của B1 không giống như vector khối của A1, chèn vector khối của khối B1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B1 giống như vector khối A1, hoặc B1 không khả dụng hoặc B1 không sử dụng chế độ IBC), không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia A1 và B1).

Nếu khối lân cận B0 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B0 giống như B1. Nếu vector khối của B0 không giống như vector khối của B1, chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B0 giống như vector khối B1, hoặc B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), không chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia B0 và B1).

Nếu khối lân cận A0 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của A0 giống như A1. Nếu vector khối của A0 không giống như vector khối của A1, chèn vector khối của khối A0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của A0 giống như vector khối A1, hoặc A0 không khả dụng hoặc A0 không sử dụng chế độ IBC), không chèn vector khối của khối A0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tia A0 và A1).

Nếu khối lân cận B2 khả dụng và đang sử dụng chế độ IBC và kích thước danh sách hợp nhất IBC hiện tại nhỏ hơn 4, tia (hoặc xác định) liệu vector khối của B2 giống như A1 và liệu vector khối của B2 giống như B1. Nếu vector khối của B2 không giống như vector khối của A1 và vector khối của B2 không giống như B1, chèn vector khối của khối B2 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B2 giống như vector khối A1 hoặc B1, hoặc B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), không chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại

(tĩa B2 và A1 và tĩa B2 và B1).

Nếu A1 đã nằm trong danh sách hợp nhất IBC, thì

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_k không giống như A1, chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_k giống như vector khối của A1), không chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_k và A1).

Nếu ứng viên HMVP cuối cùng thứ hai H_{k-1} khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_{k-1} không giống như A1, chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_{k-1} không khả dụng hoặc H_{k-1} không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_{k-1} giống như vector khối của A1), không chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_{k-1} và A1).

Ngược lại nếu A1 không nằm trong danh sách hợp nhất IBC và B1 đã ở trong danh sách hợp nhất IBC, thì

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_k không giống như B1, chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_k giống như vector khối của B1), không chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_k và B1).

Nếu ứng viên HMVP cuối cùng thứ hai H_{k-1} khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_{k-1} không giống như B1, chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_{k-1} không khả dụng hoặc H_{k-1} không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_{k-1} giống như vector khối của B1), không chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_{k-1} và B1).

Ngược lại nếu A1 và B1 không nằm trong danh sách hợp nhất IBC thì,

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1), không chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại (không tĩa).

Nếu ứng viên HMVP cuối cùng thứ hai H_{k-1} khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_{k-1} không khả dụng hoặc H_{k-1} không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1), không chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại (không tĩa).

Chèn các ứng viên HMVP còn lại lần lượt, nếu ứng viên khả dụng và đang sử dụng chế độ IBC và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1 (không tĩa).

Chèn ứng viên theo cặp, kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất (không tĩa).

Giải pháp 8

Theo giải pháp 8, tia vector khối của khối lân cận không gian được duy trì. Đối với phép tia ứng viên HMVP,

vector khối của ứng viên HMVP cuối cùng H_k và ứng viên HMVP cuối cùng thứ hai H_{k-1} được tia với vector khối của khối lân cận không gian của A1, khi A1 đã ở trong danh sách hợp nhất IBC, và B1 không nằm trong danh sách hợp nhất IBC;

vector khối của ứng viên HMVP cuối cùng H_k và ứng viên HMVP cuối cùng thứ hai H_{k-1} được tia với vector khối của khối lân cận không gian của B1, khi B1 đã ở trong danh sách hợp nhất IBC, và A1 không nằm trong danh sách hợp nhất IBC;

vector khối của ứng viên HMVP cuối cùng H_k được tia bằng vector khối của khối lân cận không gian của B1, khi cả A1 và B1 đã ở trong danh sách hợp nhất IBC;

Nếu A1 hoặc B1 không nằm trong danh sách hợp nhất IBC, thì ứng viên HMVP không được chứng minh với ứng viên không gian.

Trong trường hợp này, số lượng lớn nhất phép xen tia đối với việc tạo danh sách IBC ứng viên hợp nhất của khối hiện tại được giảm từ 9 xuống 7. Giải pháp này giảm đáng kể độ phức tạp tạo IBC danh sách hợp nhất cho cả bộ mã hóa lẫn bộ giải mã.

Trong ví dụ, danh sách IBC ứng viên hợp nhất cụ thể được tạo như sau:

Nếu khối lân cận A1 khả dụng và đang sử dụng chế độ IBC, chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (A1 không khả dụng hoặc A1 không sử dụng chế độ IBC), không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (ứng viên thứ nhất, không tia).

Nếu khối lân cận B1 khả dụng và đang sử dụng chế độ IBC, tia (hoặc xác định) liệu vector khối của B1 giống như A1. Nếu vector khối của B1 không giống như vector khối của A1, chèn vector khối của khối B1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B1 giống như vector khối A1, hoặc B1 không khả dụng hoặc B1 không sử dụng chế độ IBC),

không chèn vector khối của khối A1 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tía A1 và B1).

Nếu khối lân cận B0 khả dụng và đang sử dụng chế độ IBC, tía (hoặc xác định) liệu vector khối của B0 giống như B1. Nếu vector khối của B0 không giống như vector khối của B1, chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B0 giống như vector khối B1, hoặc B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), không chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tía B0 và B1).

Nếu khối lân cận A0 khả dụng và đang sử dụng chế độ IBC, tía (hoặc xác định) liệu vector khối của A0 giống như A1. Nếu vector khối của A0 không giống như vector khối của A1, chèn vector khối của khối A0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của A0 giống như vector khối A1, hoặc A0 không khả dụng hoặc A0 không sử dụng chế độ IBC), không chèn vector khối của khối A0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tía A0 và A1).

Nếu khối lân cận B2 khả dụng và đang sử dụng chế độ IBC và kích thước danh sách hợp nhất IBC hiện tại nhỏ hơn 4, tía (hoặc xác định) liệu vector khối của B2 giống như A1 và liệu vector khối của B2 giống như B1. Nếu vector khối của B2 không giống như vector khối của A1 và vector khối của B2 không giống như B1, chèn vector khối của khối B2 vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của B2 giống như vector khối A1 hoặc B1, hoặc B0 không khả dụng hoặc B0 không sử dụng chế độ IBC), không chèn vector khối của khối B0 vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tía B2 và A1 và tía B2 và B1).

Nếu A1 đã nằm trong danh sách hợp nhất IBC và B1 không nằm trong danh sách, thì

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_k không giống như A1, chèn vector khối của H_k vào danh sách ứng viên hợp nhất

IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_k giống như vector khối của $A1$), không chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_k và $A1$).

Nếu ứng viên HMVP cuối cùng thứ hai H_{k-1} khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_{k-1} không giống như $A1$, chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_{k-1} không khả dụng hoặc H_{k-1} không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_{k-1} giống như vector khối của $A1$), không chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_{k-1} và $A1$).

Ngược lại nếu $A1$ không nằm trong danh sách hợp nhất IBC và $B1$ đã nằm trong danh sách hợp nhất IBC, thì

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_k không giống như $B1$, chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_k giống như vector khối của $B1$), không chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_k và $B1$).

Nếu ứng viên HMVP cuối cùng thứ hai H_{k-1} khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_{k-1} không giống như $B1$, chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_{k-1} không khả dụng hoặc H_{k-1} không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất

hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_{k-1} giống như vector khối của B1), không chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_{k-1} và B1).

Ngược lại nếu cả A1 và B1 đã nằm trong danh sách hợp nhất IBC, thì

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, và vector khối của H_k không giống như A1 và vector khối của H_k không giống như B1, chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_k giống như vector khối của A1 hoặc vector khối của H_k giống như vector khối của B1), không chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tĩa H_k và A1 và B1).

Ngược lại nếu A1 và B1 không nằm trong danh sách hợp nhất IBC thì,

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1), không chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại (không tĩa).

Nếu ứng viên HMVP cuối cùng thứ hai H_{k-1} khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, chèn vector khối của H_{k-1} vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_{k-1} không khả dụng hoặc H_{k-1} không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1), không chèn vector khối của H_{k-1} vào danh sách

ứng viên hợp nhất IBC của khối hiện tại (không tĩa).

Chèn các ứng viên HMVP còn lại lần lượt, nếu ứng viên khả dụng và đang sử dụng chế độ IBC và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1 (không tĩa).

Chèn ứng viên theo cặp, kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất (không tĩa).

Giải pháp 9

Theo giải pháp 9, ở phía trên giải pháp 1, lân cận không gian B0 dựa trên tính khả dụng của B0 và B1; lân cận không gian A0 dựa trên tính khả dụng của A0 và A1; lân cận không gian B2 dựa trên tính khả dụng của B2, B1 và A1,

Trong ví dụ,

Nếu khối lân cận B0 khả dụng và đang sử dụng chế độ IBC và B1 không khả dụng, vector khối của khối B0 được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (B0 không khả dụng hoặc B0 không sử dụng chế độ IBC or B1 khả dụng), vector khối của khối B0 không được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại (không tĩa).

Nếu khối lân cận A0 khả dụng và đang sử dụng chế độ IBC và A1 không khả dụng, vector khối của khối A0 được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (A0 không khả dụng hoặc A0 không sử dụng chế độ IBC hoặc A1 khả dụng), vector khối của khối A0 không được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại (không tĩa).

Nếu khối lân cận B2 khả dụng và đang sử dụng chế độ IBC both A1 và B1 are not available, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn 4, vector khối của khối B2 được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (B2 không khả dụng hoặc B2 không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn 4 hoặc A1 khả dụng hoặc B1 khả dụng), vector khối của khối B2 không được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại (không tĩa).

Trong giải pháp này, không tĩa thêm, nhưng đạt được hiệu suất tạo mã tốt hơn so với giải pháp 1.

Giải pháp 10

Theo giải pháp 10, giải pháp bất kỳ trong các giải pháp từ 1 đến 9 có thể được kết hợp với nhau để giảm độ phức tạp tạo danh sách hợp nhất IBC.

Trong ví dụ đối với giải pháp 10, giải pháp 2 và giải pháp 3 được kết hợp. Do các khối lân cận không gian bên trái (a_1) và bên trên (B_1) là quan trọng để dự báo khối hiện tại nhờ sử dụng chế độ IBC. Trong ví dụ này, khối lân cận không gian tia giữa A_1 và B_1 được duy trì, các vector khối của khối lân cận không gian còn lại không được chèn vào danh sách hợp nhất IBC. Để tia ứng viên HMVP, vector khối của ứng viên HMVP cuối cùng H_k được tia bằng vector khối khối lân cận không gian A_1 và B_1 . Theo phương án thực hiện, phép tia sau có thể được thực hiện trong khi tạo danh sách hợp nhất IBC:

- Tia giữa A_1 và B_1
- Tia giữa ứng viên HMVP cuối cùng H_k và A_1
- Tia giữa ứng viên HMVP cuối cùng H_k và B_1 .

Trong trường hợp này, số lượng lớn nhất phép xen tia đối với việc tạo danh sách IBC ứng viên hợp nhất của khối hiện tại được giảm từ 9 xuống 3. Giải pháp này giảm đáng kể độ phức tạp tạo IBC danh sách hợp nhất cho cả bộ mã hóa lẫn bộ giải mã.

Trong ví dụ, ứng viên cuối cùng của danh sách ứng viên hợp nhất IBC được phép nhờ sử dụng chế độ HMVP.

Trong ví dụ, danh sách IBC ứng viên hợp nhất cụ thể được tạo như sau:

Nếu khối lân cận A_1 khả dụng và đang sử dụng chế độ IBC, vector khối của khối A_1 được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (A_1 không khả dụng hoặc A_1 không sử dụng chế độ IBC), vector khối của khối A_1 không được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại (ứng viên thứ nhất, không tia).

Nếu khối lân cận B_1 khả dụng và đang sử dụng chế độ IBC và vector khối của khối A_1 được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại, tia (hoặc xác định) liệu vector khối của B_1 giống như A_1 . Nếu vector khối của B_1 không giống như vector khối của A_1 , vector khối của khối B_1 được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (vector khối của

B1 giống như vector khối A1, hoặc B1 không khả dụng hoặc B1 không sử dụng chế độ IBC), vector khối của khối B1 không được đưa vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tỉa A1 và B1).

Nếu ứng viên HMVP cuối cùng H_k khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn giá trị, giá trị bằng số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1; và vector khối của H_k không giống như A1 và B1, chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại. Ngược lại (H_k không khả dụng hoặc H_k không sử dụng chế độ IBC hoặc kích thước danh sách IBC ứng viên hợp nhất hiện tại không nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1, hoặc vector khối của H_k giống như vector khối của A1, hoặc vector khối của H_k giống như vector khối của B1) không chèn vector khối của H_k vào danh sách ứng viên hợp nhất IBC của khối hiện tại (tỉa H_k và A1, H_k và B1).

Chèn các ứng viên HMVP còn lại lần lượt, nếu ứng viên khả dụng và đang sử dụng chế độ IBC, và kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng IBC ứng viên hợp nhất lớn nhất trừ đi 1 (không tỉa).

Trong một số ví dụ, việc chèn thêm ứng viên theo cặp, kích thước danh sách IBC ứng viên hợp nhất hiện tại nhỏ hơn số lượng ứng viên hợp nhất IBC lớn nhất (không tỉa).

Trong ví dụ, đối với quá trình giải mã đối với các CU được tạo mã trong chế độ dự báo IBC.

8.6.1 Quá trình giải mã tổng quát cho các CU được tạo mã trong chế độ dự báo IBC

Các đầu vào cho quá trình này là:

vị trí độ sáng (x_{Cb} , y_{Cb}) xác định mẫu trên bên trái của CB hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,

biến $cbWidth$ xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,

biến $cbHeight$ xác định chiều cao của CB hiện tại trong các mẫu độ sáng,

biến $treeType$ xác định liệu cây đơn hoặc cây kép được sử dụng và nếu cây

kép được sử dụng, xác định liệu cây hiện tại tương ứng với các thành phần độ sáng hoặc sắc độ.

Đầu ra của quá trình này là ảnh được tái tạo được chỉnh sửa trước khi lọc trong vòng.

Quá trình dẫn xuất đối với các QP được gọi bằng vị trí độ sáng (x_{Cb} , y_{Cb}), chiều rộng của CB hiện tại trong các mẫu độ sáng $cbWidth$ và chiều cao của CB hiện tại trong các mẫu độ sáng $cbHeight$, và biến $treeType$ làm các đầu vào.

Biến $IsGt4by4$ được dẫn xuất như sau:

$$IsGt4by4 = (cbWidth * cbHeight) > 16 \quad (1111).$$

Quá trình giải mã cho các CU được tạo mã trong chế độ dự báo IBC gồm các bước có thứ tự sau:

1. Các thành phần vectơ khối của CU hiện tại được suy ra như sau:

Quá trình dẫn xuất đối với các thành phần vectơ khối được gọi bằng vị trí CB độ sáng (x_{Cb} , y_{Cb}), chiều rộng CB độ sáng $cbWidth$ và chiều cao CB độ sáng $cbHeight$ làm các đầu vào, và vectơ khối độ sáng bvL làm đầu ra.

Khi $treeType$ bằng `SINGLE_TREE`, quá trình dẫn xuất cho khối sắc độ được gọi bằng vectơ khối độ sáng bvL làm đầu vào, và vectơ khối sắc độ bvC làm đầu ra.

2. Các mẫu dự báo của CU hiện tại được suy ra như sau:

Quá trình giải mã cho các khối IBC như được xác định trong mệnh đề 8.6.3.1 được gọi bằng vị trí CB độ sáng (x_{Cb} , y_{Cb}), chiều rộng CB độ sáng $cbWidth$ và chiều cao CB độ sáng $cbHeight$, vectơ khối độ sáng bvL , biến $cIdx$ được thiết lập bằng 0 làm các đầu vào, và các mẫu dự báo IBC ($predSamples$) vốn là mảng $(cbWidth) \times (cbHeight)$ $predSamplesL$ của các mẫu độ sáng dự báo làm đầu ra.

Khi $treeType$ bằng `SINGLE_TREE`, các mẫu dự báo của CU hiện tại được suy ra như sau:

-Quá trình giải mã cho các khối IBC như được xác định trong mệnh đề 8.6.3.1 được gọi bằng vị trí CB độ sáng (x_{Cb} , y_{Cb}), chiều rộng CB độ sáng $cbWidth$ và chiều cao CB độ sáng $cbHeight$, vectơ khối sắc độ bvC và biến $cIdx$ được thiết lập bằng 1 làm các đầu vào, và các mẫu dự báo IBC ($predSamples$) vốn là

mảng $(cbWidth/SubWidthC) \times (cbHeight/SubHeightC)$ $predSamplesCb$ của các mẫu sắc độ dự báo cho các thành phần sắc độ Cb làm đầu ra.

- Quá trình giải mã cho các khối IBC như được xác định trong mệnh đề 8.6.3.1 được gọi bằng vị trí CB độ sáng (xCb, yCb) , chiều rộng CB độ sáng $cbWidth$ và chiều cao CB độ sáng $cbHeight$, vectơ khối sắc độ bvC và biến $cIdx$ được thiết lập bằng 2 làm các đầu vào, và các mẫu dự báo IBC ($predSamples$) vốn là mảng $(cbWidth/SubWidthC) \times (cbHeight/SubHeightC)$ $predSamplesCr$ của các mẫu sắc độ dự báo cho các thành phần sắc độ Cr làm các đầu ra.

3. Các mẫu dư của CU hiện tại được suy ra như sau:

- Quá trình giải mã cho tín hiệu dư của các CB được tạo mã trong chế độ dự báo ngoài được gọi bằng vị trí $(xTb0, yTb0)$ được thiết lập bằng vị trí độ sáng (xCb, yCb) , chiều rộng $nTbW$ được thiết lập bằng chiều rộng CB độ sáng $cbWidth$, chiều cao $nTbH$ được thiết lập bằng chiều cao CB độ sáng $cbHeight$ và biến $cIdx$ được thiết lập bằng 0 làm các đầu vào, và mảng $resSamplesL$ làm đầu ra.

- Khi $treeType$ bằng `SINGLE_TREE`, quá trình giải mã cho tín hiệu dư của các CB được tạo mã trong chế độ dự báo ngoài được gọi bằng vị trí $(xTb0, yTb0)$ được thiết lập bằng vị trí sắc độ $(xCb/SubWidthC, yCb/SubHeightC)$, chiều rộng $nTbW$ được thiết lập bằng chiều rộng CB sắc độ $cbWidth/SubWidthC$, chiều cao $nTbH$ được thiết lập bằng chiều cao CB sắc độ $cbHeight/SubHeightC$ and biến $cIdx$ được thiết lập bằng 1 làm các đầu vào, và mảng $resSamplesCb$ làm đầu ra.

- Khi $treeType$ bằng `SINGLE_TREE`, quá trình giải mã cho tín hiệu dư của các CB được tạo mã trong chế độ dự báo ngoài được gọi bằng vị trí $(xTb0, yTb0)$ được thiết lập bằng vị trí sắc độ $(xCb/SubWidthC, yCb/SubHeightC)$, chiều rộng $nTbW$ được thiết lập bằng chiều rộng CB sắc độ $cbWidth/SubWidthC$, chiều cao $nTbH$ được thiết lập bằng chiều cao CB sắc độ $cbHeight/SubHeightC$ và biến $cIdx$ được thiết lập bằng 2 làm các đầu vào, và mảng $resSamplesCr$ làm đầu ra.

4. Các mẫu được tái tạo của CU hiện tại được suy ra như sau:

- Quá trình tái tạo ảnh cho thành phần màu được gọi bằng vị trí khối

(xCurr, yCurr) được thiết lập bằng (xCb, yCb), chiều rộng khối nCurrSw được thiết lập bằng cbWidth, chiều cao khối nCurrSh được thiết lập bằng cbHeight, biến cIdx được thiết lập bằng 0, mảng (cbWidth)x(cbHeight) predSamples được thiết lập bằng predSamplesL và mảng (cbWidth)x(cbHeight) resSamples được thiết lập bằng resSamplesL làm các đầu vào, và đầu ra này là ảnh được tái tạo được chỉnh sửa trước khi lọc trong vòng.

- Khi treeType bằng SINGLE_TREE, quá trình tái tạo ảnh cho thành phần màu được gọi bằng vị trí khối (xCurr, yCurr) được thiết lập bằng (xCb/SubWidthC, yCb/SubHeightC), chiều rộng khối nCurrSw được thiết lập bằng cbWidth/SubWidthC, chiều cao khối nCurrSh được thiết lập bằng cbHeight/SubHeightC, biến cIdx được thiết lập bằng 1, mảng (cbWidth/SubWidthC)x(cbHeight/SubHeightC) predSamples được thiết lập bằng predSamplesCb và mảng (cbWidth/SubWidthC)x(cbHeight/SubHeightC) resSamples được thiết lập bằng resSamplesCb làm các đầu vào, và đầu ra này là ảnh được tái tạo được chỉnh sửa trước khi lọc trong vòng.

- Khi treeType bằng SINGLE_TREE, quá trình tái tạo ảnh cho thành phần màu được gọi bằng vị trí khối (xCurr, yCurr) được thiết lập bằng (xCb/SubWidthC, yCb/SubHeightC), chiều rộng khối nCurrSw được thiết lập bằng cbWidth/SubWidthC, chiều cao khối nCurrSh được thiết lập bằng cbHeight/SubHeightC, biến cIdx được thiết lập bằng 2, mảng (cbWidth/SubWidthC)x(cbHeight/SubHeightC) predSamples được thiết lập bằng predSamplesCr và mảng (cbWidth/SubWidthC)x(cbHeight/SubHeightC) resSamples được thiết lập bằng resSamplesCr làm các đầu vào, và đầu ra này là ảnh được tái tạo được chỉnh sửa trước khi lọc trong vòng.

8.6.2 Quá trình dẫn xuất đối với các thành phần vectơ khối cho các khối IBC

8.6.2.1 Tổng quát

Các đầu vào cho quá trình này là:

vị trí độ sáng (xCb, yCb) của mẫu trên bên trái của CB độ sáng hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,

biến cbWidth xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,

biến cbHeight xác định chiều cao của CB hiện tại trong các mẫu độ sáng.

Các đầu ra của quá trình này là:

vectơ khối độ sáng có độ chính xác mẫu phân số $1/16$ bvL.

Vectơ khối độ sáng mvL được dẫn xuất như sau:

-Quá trình dẫn xuất để dự báo IBC vectơ khối độ sáng như được xác định trong mệnh đề 8.6.2.2 được gọi bằng vị trí độ sáng (xCb, yCb), các biến cbWidth và các đầu vào cbHeight, và đầu ra là vectơ khối độ sáng bvL.

Khi `general_merge_flag[xCb][yCb]` bằng 0, phần sau áp dụng:

1. Biến bvd được dẫn xuất như sau:

$$\text{bvd}[0] = \text{MvdL0}[\text{xCb}][\text{yCb}][0] \quad (1112)$$

$$\text{bvd}[1] = \text{MvdL0}[\text{xCb}][\text{yCb}][1] \quad (1113)$$

2. Quá trình làm tròn đối với các MV được gọi bằng mvX được thiết lập bằng bvL, `rightShift` được thiết lập bằng `AmvrShift`, và `leftShift` được thiết lập bằng `AmvrShift` làm các đầu vào và bvL được làm tròn làm đầu ra.

3. Vectơ khối độ sáng bvL được thay đổi như sau:

$$u[0] = (\text{bvL}[0] + \text{bvd}[0] + 2^{18}) \% 2^{18} \quad (1114)$$

$$\text{bvL}[0] = (u[0] \geq 2^{17}) ? (u[0] - 2^{18}) : u[0] \quad (1115)$$

$$u[1] = (\text{bvL}[1] + \text{bvd}[1] + 2^{18}) \% 2^{18} \quad (1116)$$

$$\text{bvL}[1] = (u[1] \geq 2^{17}) ? (u[1] - 2^{18}) : u[1] \quad (1117)$$

Lưu ý 1 – Các giá trị kết quả của `bvL[0]` và `bvL[1]` như nêu trên sẽ luôn nằm trong cả khoảng từ -2^{17} đến $2^{17} - 1$.

Khi `IsGt4by4` bằng TRUE, quá trình cập nhật cho danh sách bộ dự báo vectơ khối dựa trên lịch sử như được xác định trong mệnh đề 8.6.2.6 được gọi bằng vectơ khối độ sáng bvL.

Sự tương thích dòng bit yêu cầu rằng vectơ khối độ sáng bvL phải tuân theo các ràng buộc sau:

$$\text{CtbSizeY} \geq ((\text{yCb} + (\text{bvL}[1] \gg 4)) \& (\text{CtbSizeY} - 1)) + \text{cbHeight}.$$

$$\begin{aligned} & \text{IbcVirBuf}[0][(\text{x} + (\text{bvL}[0] \gg 4)) \& (\text{IbcBufWidthY} - 1)][(\text{y} + (\text{bvL}[1] \gg 4)) \\ & \& (\text{CtbSizeY} - 1)] \text{ sẽ không bằng } -1 \text{ với } \text{x} = \text{xCb}.. \text{xCb} + \text{cbWidth} - 1 \text{ và} \\ & \text{y} = \text{yCb}.. \text{yCb} + \text{cbHeight} - 1. \end{aligned}$$

8.6.2.2 Quá trình dẫn xuất để dự báo IBC vectơ khối độ sáng

Quá trình này chỉ được gọi khi `CuPredMode[0][xCb][yCb]` bằng

MODE_IBC, trong đó (xCb, yCb) xác định mẫu trên bên trái của CB độ sáng hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại.

Các đầu vào cho quá trình này là:

a vị trí độ sáng (xCb, yCb) của mẫu trên bên trái của CB độ sáng hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,

biến cbWidth xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,

biến cbHeight xác định chiều cao của CB hiện tại trong các mẫu độ sáng.

Các đầu ra của quá trình này là:

vector khối độ sáng có độ chính xác mẫu phân số 1/16 bvL.

Vector khối độ sáng bvL được dẫn xuất nhờ các bước có thứ tự sau:

1. Khi IsGt4by4 bằng TRUE, quá trình dẫn xuất đối với các ứng viên vector khối không gian từ các CU lân cận như được xác định trong mệnh đề 8.6.2.3 được gọi bằng vị trí CB độ sáng (xCb, yCb), chiều rộng CB độ sáng cbWidth và chiều cao cbHeight làm các đầu vào, và các đầu ra là các cờ khả dụng availableFlagA1, availableFlagB1 và các vector khối bvA1 và bvB1.

2. Khi IsGt4by4 bằng TRUE, danh sách ứng viên vector khối, bvCandList, được tạo như sau:

i = 0

if(availableFlagA1)

bvCandList [i++] = bvA1 (1118)

if(availableFlagB1)

bvCandList [i++] = bvB1

3. Biến numCurrCand được dẫn xuất như sau:

IsGt4by4 bằng TRUE, numCurrCand được thiết lập bằng số lượng ứng viên hợp nhất trong bvCandList.

Ngược lại (IsGt4by4 bằng FALSE), numCurrCand được thiết lập bằng 0.

4. Khi numCurrCand < MaxNumIbcMergeCand và NumHmvpIbcCand > 0, quá trình dẫn xuất của các ứng viên vector khối dựa trên lịch sử IBC như được nêu trong 8.6.2.4 được gọi bằng bvCandList, và numCurrCand làm các đầu vào, và bvCandList and numCurrCand cải biến làm các đầu ra.

5. Khi numCurrCand < MaxNumIbcMergeCand, phần sau áp dụng cho đến khi

numCurrCand bằng MaxNumIbcMergeCand:

bvCandList[numCurrCand][0] được thiết lập bằng 0.

bvCandList[numCurrCand][1] được thiết lập bằng 0.

numCurrCand được tăng lên 1.

6. Biến bvIdx được dẫn xuất như sau:

$$bvIdx = \text{general_merge_flag}[xCb][yCb] ? \text{merge_idx}[xCb][yCb] : \text{mvp_l0_flag}[xCb][yCb] \quad (1119)$$

7. Các phép gán sau được thực hiện:

$$bvL[0] = bvCandList[mvIdx][0] \quad (1120)$$

$$bvL[1] = bvCandList[mvIdx][1] \quad (1121)$$

8.6.2.3 Quá trình dẫn xuất đối với các ứng viên vector khối không gian IBC

Các đầu vào cho quá trình này là:

a vị trí độ sáng (xCb, yCb) của mẫu trên bên trái của CB độ sáng hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,

biến cbWidth xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,

biến cbHeight xác định chiều cao của CB hiện tại trong các mẫu độ sáng.

Các đầu ra của quá trình này là như sau:

các cờ khả dụng availableFlagA1 và availableFlagB1 của các CU lân cận,

các vector khối có độ chính xác mẫu phân số 1/16 bvA1, và bvB1 của các

CU lân cận,

Để dẫn xuất availableFlagA1 và mvA1, điều kiện sau áp dụng:

Vị trí độ sáng (xNbA1, yNbA1) bên trong CB độ sáng lân cận được thiết lập bằng $(xCb - 1, yCb + cbHeight - 1)$.

Quá trình dẫn xuất đối với tính khả dụng khối lân cận được gọi bằng vị trí độ sáng hiện tại (xCurr, yCurr) được thiết lập bằng (xCb, yCb), vị trí độ sáng lân cận (xNbA1, yNbA1), checkPredModeY được thiết lập bằng TRUE, và cIdx được thiết lập bằng 0 làm các đầu vào, và đầu ra được gán cho cờ khả dụng khối availableA1.

Các biến availableFlagA1 và bvA1 được suy ra như sau:

Nếu availableA1 bằng FALSE, availableFlagA1 được thiết lập bằng 0 và cả hai thành phần của bvA1 được thiết lập bằng 0.

Ngược lại, availableFlagA1 được thiết lập bằng 1 và các phép gán sau được thực hiện:

$$bvA1 = MvL0[xNbA1][yNbA1] \quad (1122)$$

Để dẫn xuất availableFlagB1 và bvB1, điều kiện sau áp dụng:

Vị trí độ sáng (xNbB1, yNbB1) bên trong CB độ sáng lân cận được thiết lập bằng (xCb + cbWidth - 1, yCb - 1).

Quá trình dẫn xuất đối với tính khả dụng khối lân cận được gọi bằng vị trí độ sáng hiện tại (xCurr, yCurr) được thiết lập bằng (xCb, yCb), vị trí độ sáng lân cận (xNbB1, yNbB1), checkPredModeY được thiết lập bằng TRUE, và cIdx được thiết lập bằng 0 làm các đầu vào, và đầu ra được gán cho cờ khả dụng khối availableB1.

Các biến availableFlagB1 và bvB1 được suy ra như sau:

Nếu một hoặc nhiều điều kiện sau là true (đúng), availableFlagB1 được thiết lập bằng 0 và cả hai thành phần của bvB1 được thiết lập bằng 0:

availableB1 bằng FALSE.

availableA1 bằng TRUE và các vị trí độ sáng (xNbA1, yNbA1) và (xNbB1, yNbB1) có cùng các vectơ khối.

Ngược lại, availableFlagB1 được thiết lập bằng 1 và các phép gán sau được thực hiện:

$$bvB1 = MvL0[xNbB1][yNbB1] \quad (1123).$$

8.6.2.4 Quá trình dẫn xuất cho các ứng viên vectơ khối dựa trên lịch sử IBC

Các đầu vào cho quá trình này là:

danh sách ứng viên vectơ khối bvCandList,

số lượng ứng viên vectơ khối khả dụng trong danh sách numCurrCand.

Các đầu ra cho quá trình này là:

danh sách ứng viên vectơ khối được thay đổi bvCandList,

số lượng được thay đổi của các ứng viên MV trong danh sách numCurrCand.

Các biến isPrunedA1 và isPrunedB1 đều được gán bằng FALSE.

Đối với mỗi ứng viên trong HmvpIbcCandList[hMvpIdx] có chỉ số hMvpIdx = 1..NumHmvpIbcCand, các bước có thứ tự sau được lặp lại cho đến

khi numCurrCand bằng MaxNumIbcMergeCand:

1. Biến sameMotion được dẫn xuất như sau:

Nếu tất cả các điều kiện sau là đúng cho ứng viên vector khối bất kỳ N với N là A1 hoặc B1, sameMotion và isPrunedN đều được thiết lập bằng TRUE:

IsGt4by4 bằng TRUE.

hMvpIdx bằng 1.

Ứng viên HmvpIbcCandList[NumHmvpIbcCand – hMvpIdx] bằng ứng viên vector khối N.

isPrunedN bằng FALSE.

Ngược lại, sameMotion được thiết lập bằng FALSE.

2. Khi sameMotion bằng FALSE, ứng viên HmvpIbcCandList[NumHmvpIbcCand – hMvpIdx] được bổ sung vào danh sách ứng viên vector khối như sau:

bvCandList[numCurrCand++] =

HmvpIbcCandList[NumHmvpIbcCand – hMvpIdx] (1124).

8.6.2.5 Quá trình dẫn xuất cho các vector khối sắc độ

Đầu vào cho quá trình này là:

vector khối độ sáng có độ chính xác mẫu phân số 1/16 bvL.

Đầu ra của quá trình này là vector khối sắc độ có độ chính xác mẫu phân số 1/32 bvC.

Vector khối sắc độ được suy ra từ vector khối độ sáng tương ứng.

Vector khối sắc độ bvC được dẫn xuất như sau:

$$bvC[0] = ((bvL[0] \gg (3 + SubWidthC)) * 32 \quad (1125)$$

$$bvC[1] = ((bvL[1] \gg (3 + SubHeightC)) * 32 \quad (1126)$$

8.6.2.6 Quá trình cập nhật đối với danh sách ứng viên bộ dự báo vector khối dựa trên lịch sử

Các đầu vào cho quá trình này là:

vector khối độ sáng bvL có độ chính xác mẫu phân số 1/16.

Danh sách ứng viên HmvpIbcCandList được thay đổi nhờ các bước có thứ tự sau:

1. Biến identicalCandExist được thiết lập bằng FALSE và biến removeIdx được

thiết lập bằng 0.

2. Khi $\text{NumHmvpIbcCand} > 0$, đối với mỗi chỉ số hMvpIdx có $\text{hMvpIdx} = 0..\text{NumHmvpIbcCand} - 1$, các bước sau áp dụng cho đến khi $\text{identicalCandExist}$ bằng TRUE:

Khi bvL bằng $\text{HmvpIbcCandList}[\text{hMvpIdx}]$, $\text{identicalCandExist}$ được thiết lập bằng TRUE và removeIdx được thiết lập bằng hMvpIdx .

3. Danh sách ứng viên HmvpIbcCandList được cập nhật như sau:

Nếu $\text{identicalCandExist}$ bằng TRUE hoặc NumHmvpIbcCand bằng 5, điều kiện sau áp dụng:

Đối với mỗi chỉ số i có $i = (\text{removeIdx} + 1)..\text{NumHmvpIbcCand} - 1$, $\text{HmvpIbcCandList}[i - 1]$ được thiết lập bằng $\text{HmvpIbcCandList}[i]$.

$\text{HmvpIbcCandList}[\text{NumHmvpIbcCand} - 1]$ được thiết lập bằng bvL .

Ngược lại ($\text{identicalCandExist}$ bằng FALSE và $\text{NumHmvpIbcCand} < 5$), điều kiện sau áp dụng:

$\text{HmvpIbcCandList}[\text{NumHmvpIbcCand}++]$ được thiết lập bằng bvL .

8.6.3 Quá trình giải mã cho các khối IBC

8.6.3.1 Tổng quát

Quá trình này được gọi khi giải mã CU được tạo mã trong chế độ dự báo IBC.

Các đầu vào cho quá trình này là:

a vị trí độ sáng (xCb , yCb) xác định mẫu trên bên trái của CB hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,

biến cbWidth xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,

biến cbHeight xác định chiều cao của CB hiện tại trong các mẫu độ sáng,

vector khối bv ,

biến cIdx xác định chỉ số thành phần màu của khối hiện tại.

Các đầu ra của quá trình này là:

mảng predSamples của các mẫu dự báo.

Khi $\text{cIdx} = 0$, đối với $x = \text{xCb}..\text{xCb} + \text{cbWidth} - 1$ và $y = \text{yCb}..\text{yCb} + \text{cbHeight} - 1$, điều kiện sau áp dụng:

$$\text{xVb} = (x + (\text{bv}[0] \gg 4)) \& (\text{IbcBufWidthY} - 1) \quad (1127)$$

$$yVb = (y + (bv[1] \gg 4)) \& (CtbSizeY - 1) \quad (1128)$$

$$predSamples[x][y] = ibcVirBuf[0][xVb][yVb] \quad (1129)$$

Khi $cIdx$ không bằng 0, với $x = xCb/subWidthC..xCb/subWidthC + cbWidth/subWidthC - 1$ và $y = yCb/subHeightC..yCb/subHeightC + cbHeight/subHeightC - 1$, điều kiện sau áp dụng:

$$xVb = (x + (bv[0] \gg 5)) \& (IbcBufWidthC - 1) \quad (1130)$$

$$yVb = (y + (bv[1] \gg 5)) \& ((CtbSizeY/subHeightC) - 1) \quad (1131)$$

$$predSamples[x][y] = ibcVirBuf[cIdx][xVb][yVb] \quad (1132)$$

Khi $cIdx$ bằng 0, các phép gán sau được thực hiện đối với $x = 0..cbWidth - 1$ và $y = 0..cbHeight - 1$:

$$MvL0[xCb + x][yCb + y] = bv \quad (1133)$$

$$MvL1[xCb + x][yCb + y] = 0 \quad (1134)$$

$$RefIdxL0[xCb + x][yCb + y] = -1 \quad (1135)$$

$$RefIdxL1[xCb + x][yCb + y] = -1 \quad (1136)$$

$$PredFlagL0[xCb + x][yCb + y] = 0 \quad (1137)$$

$$PredFlagL1[xCb + x][yCb + y] = 0 \quad (1138)$$

$$BcwIdx[xCb + x][yCb + y] = 0 \quad (1139).$$

Sau đây là phần giải thích các ứng dụng của phương pháp mã hóa cũng như phương pháp giải mã như được thể hiện trên các phương án thực hiện nêu trên, và hệ thống sử dụng chúng.

Fig.8 là sơ đồ khối thể hiện hệ thống cung cấp nội dung 3100 để thực hiện dịch vụ phân tán nội dung. Hệ thống cung cấp nội dung 3100 bao gồm thiết bị chụp 3102, thiết bị đầu cuối 3106, và một cách tùy chọn bao gồm màn hiển thị 3126. Thiết bị chụp 3102 truyền thông với thiết bị đầu cuối 3106 trên liên kết truyền thông 3104. Liên kết truyền thông có thể bao gồm kênh truyền thông 13 nêu trên. Liên kết truyền thông 3104 bao gồm mà không bị giới hạn ở WiFi, Ethernet, Cáp, không dây (3G/4G/5G), USB, hoặc loại tổ hợp bất kỳ của nó, hoặc tương tự.

Thiết bị chụp 3102 tạo dữ liệu, và có thể mã hóa dữ liệu bằng phương pháp mã hóa như được thể hiện trên các phương án thực hiện nêu trên. Theo cách

khác, thiết bị chụp 3102 có thể phân tán dữ liệu đến máy chủ phát trực tuyến (streaming) (không được thể hiện trên các hình vẽ), và máy chủ mã hóa dữ liệu và truyền dữ liệu được mã hóa đến thiết bị đầu cuối 3106. Thiết bị chụp 3102 bao gồm mà không bị giới hạn ở camera, điện thoại thông minh hoặc máy tính bảng, máy tính hoặc máy tính xách tay, hệ thống hội nghị video, PDA, bộ phận lắp trong xe, hoặc tổ hợp của chúng, hoặc tương tự. Chẳng hạn, thiết bị chụp 3102 có thể bao gồm thiết bị nguồn 12 như được nêu trên. Khi dữ liệu bao gồm video, bộ mã hóa video 20 được bao gồm trong thiết bị chụp 3102 có thể thực sự thực hiện xử lý mã hóa video. Khi dữ liệu bao gồm audio (tức là, giọng nói), bộ mã hóa audio được bao gồm trong thiết bị chụp 3102 có thể thực sự thực hiện xử lý mã hóa audio. Đối với một số trường hợp thực tế, thiết bị chụp 3102 phân tán dữ liệu audio và video được mã hóa bằng cách ghép kênh chúng đồng thời. Đối với các trường hợp thực tế khác, chẳng hạn trong hệ thống hội nghị video, dữ liệu audio được mã hóa và dữ liệu video được mã hóa không được ghép kênh. Thiết bị chụp 3102 phân tán dữ liệu audio được mã hóa và dữ liệu video được mã hóa đến thiết bị đầu cuối 3106 một cách riêng rẽ.

Trong hệ thống cung cấp nội dung 3100, thiết bị đầu cuối 310 nhận và tái tạo dữ liệu được mã hóa. Thiết bị đầu cuối 3106 có thể là thiết bị có khả năng nhận và khôi phục dữ liệu, chẳng hạn điện thoại thông minh hoặc máy tính bảng 3108, máy tính hoặc máy tính xách tay 3110, bộ ghi video mạng (network video recorder, NVR)/bộ ghi video số (digital video recorder, DVR) 3112, TV 3114, bộ thu tín hiệu truyền hình số (set top box, STB) 3116, hệ thống hội nghị video 3118, hệ thống giám sát video 3120, bộ hỗ trợ số cá nhân (personal digital assistant, PDA) 3122, thiết bị lắp trong xe 3124, hoặc tổ hợp bất kỳ của chúng, hoặc tương tự có thể giải mã dữ liệu được mã hóa nêu trên. Chẳng hạn, thiết bị đầu cuối 3106 có thể bao gồm thiết bị đích 14 như được nêu trên. Khi dữ liệu được mã hóa bao gồm video, bộ giải mã video 30 được bao gồm trong thiết bị đầu cuối được ưu tiên để thực hiện giải mã video. Khi dữ liệu được mã hóa bao gồm audio, bộ giải mã audio được bao gồm trong thiết bị đầu cuối được ưu tiên để thực hiện xử lý giải mã audio.

Đối với thiết bị đầu cuối có màn hiển thị, chẳng hạn, điện thoại thông minh hoặc máy tính bảng 3108, máy tính hoặc máy tính xách tay 3110, NVR/DVR 3112, TV 3114, PDA 3122, hoặc thiết bị lắp trong xe 3124, thiết bị đầu cuối có thể nạp dữ liệu được giải mã cho màn hiển thị. Đối với thiết bị đầu cuối không có màn hiển thị, chẳng hạn, STB 3116, hệ thống hội nghị video 3118, hoặc hệ thống giám sát video 3120, màn hiển thị bên ngoài 3126 được tiếp xúc ở đó để nhận và thể hiện dữ liệu được giải mã.

Khi mỗi thiết bị trong hệ thống này thực hiện mã hóa hoặc giải mã, thiết bị mã hóa ảnh hoặc thiết bị giải mã ảnh, như được thể hiện trên các phương án thực hiện nêu trên, có thể được sử dụng.

Fig.9 là sơ đồ thể hiện cấu trúc của ví dụ của thiết bị đầu cuối 3106. Sau khi thiết bị đầu cuối 3106 nhận dòng dữ liệu từ thiết bị chụp 3102, khối xử lý giao thức 3202 phân tích giao thức truyền của dòng. Giao thức bao gồm mà không bị giới hạn ở giao thức phát trực tuyến thời gian thực (Real Time Streaming Protocol, RTSP), giao thức truyền siêu văn bản (Hyper Text Transfer Protocol, HTTP), giao thức phát trực tiếp trên mạng HTTP (HTTP Live streaming protocol, HLS), MPEG-DASH, giao thức truyền thời gian thực (Real-time Transport protocol, RTP), giao thức nhắn tin thời gian thực (Real Time Messaging Protocol, RTMP), hoặc loại tổ hợp bất kỳ của nó, hoặc tương tự.

Sau khi khối xử lý giao thức 3202 xử lý dòng, tệp tin dòng được tạo. Tệp tin được xuất ra khỏi phân kênh 3204. Khối phân kênh 3204 có thể tách riêng dữ liệu được ghép kênh thành dữ liệu audio được mã hóa và dữ liệu video được mã hóa. Như được nêu trên, đối với một số trường hợp thực tế, chẳng hạn trong hệ thống hội nghị video, dữ liệu audio được mã hóa và dữ liệu video được mã hóa không được ghép kênh. Trong trường hợp này, dữ liệu được mã hóa được truyền đến bộ giải mã video 3206 và bộ giải mã audio 3208 mà không qua khối phân kênh 3204.

Qua quá trình phân kênh, dòng sơ cấp (elementary stream, ES) video, ES audio, và một cách tùy chọn phụ đề được tạo. Bộ giải mã video 3206, bao gồm bộ giải mã video 30 như được giải thích theo các phương án thực hiện nêu trên, giải mã ES video bằng phương pháp giải mã như được thể hiện trên các

phương án thực hiện nêu trên để tạo khung video, và cấp dữ liệu này cho khối đồng bộ 3212. Bộ mã hóa audio 3208, giải mã ES audio để tạo khung audio, và cấp dữ liệu này cho khối đồng bộ 3212. Theo cách khác, khung video có thể lưu trữ trong bộ đệm (không được thể hiện trên Fig.9) trước khi cấp nó cho khối đồng bộ 3212. Một cách tương tự, khung audio có thể lưu trữ trong bộ đệm (không được thể hiện trên Fig.9) trước khi cấp nó cho khối đồng bộ 3212.

Khối đồng bộ 3212 đồng bộ khung video và khung audio, và cấp video/audio cho màn hiển thị video/audio 3214. Chẳng hạn, khối đồng bộ 3212 đồng bộ hiển thị thông tin video và audio. Thông tin có thể tạo mã trong cú pháp nhờ sử dụng các nhãn thời gian liên quan đến hiển thị audio được tạo mã và dữ liệu hình ảnh và các nhãn thời gian liên quan đến việc truyền chính dòng dữ liệu.

Nếu phụ đề được bao gồm trong dòng dữ liệu, bộ giải mã phụ đề 3210 giải mã phụ đề, và đồng bộ nó với khung video và khung audio, và cấp video/audio/phụ đề cho màn hiển thị video/audio/phụ đề 3216.

Sáng chế không bị giới hạn ở hệ thống nêu trên, và thiết bị mã hóa ảnh hoặc thiết bị giải mã ảnh theo các phương án thực hiện nêu trên có thể được đưa vào hệ thống khác, chẳng hạn, hệ thống ô tô.

Ví dụ 1. Phương pháp tạo danh sách hợp nhất ứng viên đối với chế độ sao chép trong khối (Intra block copy, IBC), phương pháp bao gồm các bước:

chèn vector khối của khối lân cận bên trái của khối hiện tại vào danh sách hợp nhất ban đầu của khối hiện tại (trong ví dụ, danh sách hợp nhất ban đầu là danh sách trống trước bước chèn này), khi khối lân cận bên trái khả dụng và khối lân cận bên trái đang sử dụng chế độ IBC;

chèn vector khối của khối lân cận bên trên của khối hiện tại vào danh sách hợp nhất ban đầu (trong ví dụ, danh sách hợp nhất ban đầu là danh sách trống trước bước này, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trái của khối hiện tại), khi khối lân cận bên trên khả dụng, khối lân cận bên trên đang sử dụng chế độ IBC và vector khối của khối lân cận bên trên không giống như vector khối của khối lân cận bên trái;

chèn vector khối của khối lân cận trên bên phải của khối hiện tại vào danh

sách hợp nhất ban đầu (trong ví dụ, danh sách hợp nhất ban đầu là danh sách trống trước bước này, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trên của khối hiện tại, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trái của khối hiện tại, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trên của khối hiện tại và vector khối của khối lân cận bên trái của khối hiện tại), khi khối lân cận trên bên phải khả dụng và khối lân cận trên bên phải đang sử dụng chế độ IBC.

Ví dụ 2. Phương pháp theo ví dụ 1, trong đó phương pháp còn bao gồm bước:

chèn vector khối của khối lân cận dưới bên trái của khối hiện tại vào danh sách hợp nhất ban đầu (trong ví dụ, danh sách hợp nhất ban đầu là danh sách trống trước bước này, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trên của khối hiện tại, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trái của khối hiện tại, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trên của khối hiện tại và vector khối của khối lân cận bên trái của khối hiện tại, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trên của khối hiện tại, vector khối của khối lân cận bên trái của khối hiện tại và vector khối của khối lân cận trên bên phải), khi khối lân cận dưới bên trái khả dụng và khối lân cận dưới bên trái đang sử dụng chế độ IBC.

Ví dụ 3. Phương pháp theo ví dụ 1 hoặc 2, trong đó phương pháp còn bao gồm các bước:

chèn vector khối của khối lân cận trên bên trái của khối hiện tại vào danh sách hợp nhất ban đầu (trong ví dụ, danh sách hợp nhất ban đầu là danh sách trống trước bước này, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trên của khối hiện tại, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trái của khối hiện tại, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trên của khối hiện tại và vector khối của khối lân cận bên trái của khối hiện tại, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trên của khối hiện tại, vector khối của khối lân cận bên trái của khối hiện tại và vector khối của

khối lân cận trên bên phải, hoặc danh sách hợp nhất ban đầu bao gồm vector khối của khối lân cận bên trên của khối hiện tại, vector khối của khối lân cận bên trái của khối hiện tại, vector khối của khối lân cận trên bên phải và vector khối của khối lân cận dưới bên trái), khi khối lân cận trên bên trái khả dụng, the khối lân cận trên bên trái đang sử dụng chế độ IBC và số lượng vector khối trong danh sách hợp nhất ban đầu nhỏ hơn ngưỡng (chẳng hạn, ngưỡng = 4).

Ví dụ 4. Bộ mã hóa (20) bao gồm hệ mạch xử lý để thực hiện phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 3.

Ví dụ 5. Bộ giải mã (30) bao gồm hệ mạch xử lý để thực hiện phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 3.

Ví dụ 6. Sản phẩm chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 3.

Ví dụ 7. Bộ giải mã hoặc bộ mã hóa, bao gồm:

một hoặc nhiều bộ xử lý; và

Vật ghi máy tính đọc được bất biến được ghép nối với các bộ xử lý và lưu trữ chương trình để thực thi bởi các bộ xử lý, trong đó chương trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 3.

Các toán tử

Các toán tử được sử dụng trong sáng chế giống như các toán tử được sử dụng trong ngôn ngữ lập trình C. Tuy nhiên, các kết quả của phép chia số nguyên và các phép dịch số học được định nghĩa chính xác hơn, và các phép toán bổ sung được định nghĩa, chẳng hạn hàm nâng lũy thừa và phép chia có giá trị thực. Các quy ước đánh số và đếm số nói chung bắt đầu từ 0, chẳng hạn, “thứ nhất” tương đương với thứ 0, “thứ hai” tương đương với thứ nhất, v.v..

Các toán tử số học

Các toán tử số học sau được định nghĩa như sau:

+	Phép cộng
−	Phép trừ (như là toán tử hai đối số) hoặc âm (như là toán tử tiền)

	nguyên)
*	Phép nhân, bao gồm phép nhân ma trận
x^y	Nâng lũy thừa. Xác định x là lũy thừa của y. Trong các ngữ cảnh khác, ký hiệu này được sử dụng để viết lên trên chứ không phải để hiểu như là nâng lũy thừa.
/	Phép chia số nguyên với cắt bỏ kết quả về phía 0. Chẳng hạn, $7/4$ và $-7/-4$ được cắt gọn thành 1 và $-7/4$ và $7/-4$ được cắt gọn thành -1.
$\div$	Được sử dụng để ký hiệu phép chia trong các phương trình toán học trong đó không làm tròn hoặc cắt gọn.
$\frac{x}{y}$	Được sử dụng để ký hiệu phép chia trong các phương trình toán học trong đó không làm tròn hoặc cắt gọn.
$\sum_{i=x}^y f(i)$	Tổng của $f(i)$ với i lấy tất cả các giá trị nguyên từ x lên đến y và bao gồm y.
$x \% y$	Modulus. Số dư của x được chia cho y, được định nghĩa chỉ cho các số nguyên x với y với $x \geq 0$ và $y > 0$.

Các toán tử logic

Các toán tử logic sau được định nghĩa như sau:

$x \&\& y$ Logic Boolean “và” của x và y

$x || y$ Logic Boolean “hoặc” của x và y

! Logic Boolean “không”

$x ? y : z$ Nếu x là TRUE hoặc không bằng 0, ước tính giá trị y; Ngược lại, ước tính giá trị z.

Các toán tử quan hệ

Các toán tử quan hệ sau được định nghĩa như sau:

> Lớn hơn

>= Lớn hơn hoặc bằng

< Nhỏ hơn

<= Nhỏ hơn hoặc bằng

$==$ Bằng

$!=$ không bằng

Khi toán tử quan hệ được áp dụng cho phần tử cú pháp hoặc biến đã được gán giá trị “na” (không áp dụng), giá trị “na” được xử lý như là giá trị riêng biệt đối với phần tử cú pháp hoặc biến. Giá trị “na” được xem xét không bằng giá trị khác bất kỳ.

Các toán tử theo bit

Các toán tử theo bit dưới đây được định nghĩa như sau:

$\&$ “và” theo bit. Khi hoạt động trên các đối số số nguyên, hoạt động trên biểu diễn bù 2 của giá trị nguyên. Khi hoạt động trên đối số nhị phân chứa ít bit hơn đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung nhiều bit quan trọng hơn bằng 0.

$|$ “hoặc” theo bit. Khi hoạt động trên các đối số số nguyên, hoạt động trên biểu diễn bù 2 của giá trị nguyên. Khi hoạt động trên đối số nhị phân chứa ít bit hơn đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung nhiều bit quan trọng hơn bằng 0.

$\wedge$ “loại trừ or” theo bit. Khi hoạt động trên các đối số số nguyên, hoạt động trên biểu diễn bù 2 của giá trị nguyên. Khi hoạt động trên đối số nhị phân chứa ít bit hơn đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung nhiều bit quan trọng hơn bằng 0.

$x \gg y$ Dịch sang phải số học của biểu diễn số nguyên bù 2 của các số nhị phân $x \times y$. Hàm này được định nghĩa chỉ cho các giá trị nguyên không âm của y . Các bit được dịch thành các MSB như là kết quả của phép dịch phải có giá trị bằng MSB của x trước phép dịch.

$x \ll y$ Phép dịch trái số học của biểu diễn số nguyên bù 2 của các số nhị phân $x \times y$. Hàm này được định nghĩa chỉ cho các giá trị nguyên không âm của y . Các bit được dịch chuyển thành các bit có trọng số nhỏ nhất (least significant bit, LSB) như là kết quả của phép dịch trái có giá trị bằng 0.

Các toán tử gán

Các toán tử số học dưới đây được định nghĩa như sau:

= Toán tử gán

++ Gia tăng, tức là, $x++$ tương đương với $x = x + 1$; khi được sử dụng trong chỉ số mảng, ước tính giá trị của biến trước phép tăng.

-- Giảm, tức là, $x--$ tương đương với $x = x - 1$; khi được sử dụng trong chỉ số mảng, ước tính giá trị của biến trước phép giảm.

+= Tăng theo lượng được xác định, tức là, $x += 3$ tương đương với $x = x + 3$, và

$x += (-3)$ tương đương với $x = x + (-3)$.

-= Giảm bởi giá trị được xác định, tức là, $x -= 3$ tương đương với $x = x - 3$, và

$x -= (-3)$ tương đương với $x = x - (-3)$.

Ký hiệu khoảng

Ký hiệu sau được sử dụng để xác định khoảng của các giá trị:

$x = y..z$ lấy các giá trị nguyên bắt đầu từ y đến z , tròn gói, với x , y , và z là các số nguyên và z lớn hơn y .

Các hàm số học

Các hàm số học dưới đây được định nghĩa:

$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases}$$

Asin(x) hàm lượng giác sin ngược, hoạt động trên đối số x nằm trong khoảng từ $-1,0$ đến $1,0$, tròn gói, với giá trị đầu ra nằm trong khoảng từ $-\pi/2$ đến $\pi/2$, tròn gói, ở các đơn vị radian

Atan(x) hàm lượng giác tang ngược, hoạt động trên đối số x , có giá trị đầu ra trong khoảng từ $-\pi/2$ đến $\pi/2$, tròn gói, ở các đơn vị radian

$$\text{Atan2}(y, x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) & ; \quad x > 0 \\ \text{Atan}\left(\frac{y}{x}\right) + \pi & ; \quad x < 0 \ \&\& \ y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi & ; \quad x < 0 \ \&\& \ y < 0 \\ +\frac{\pi}{2} & ; \quad x == 0 \ \&\& \ y \geq 0 \\ -\frac{\pi}{2} & ; \quad \text{Ngược lại} \end{cases}$$

Ceil(x) số nguyên nhỏ nhất lớn hơn hoặc bằng x.

$$\text{Clip1}_Y(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_Y) - 1, x)$$

$$\text{Clip1}_C(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_C) - 1, x)$$

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{Ngược lại} \end{cases}$$

Cos(x) hàm lượng giác cosin hoạt động trên đối số x ở các đơn vị radian.

Floor(x) số nguyên lớn nhất nhỏ hơn hoặc bằng x.

$$\text{GetCurrMsb}(a, b, c, d) = \begin{cases} c + d & ; \quad b - a \geq d / 2 \\ c - d & ; \quad a - b > d / 2 \\ c & ; \quad \text{Ngược lại} \end{cases}$$

Ln(x) lôgarit tự nhiên của x (lôgarit cơ số e, trong đó e là hằng số cơ số lôgarit tự nhiên 2,718 281 828...).

Log2(x) lôgarit cơ số 2 của x.

Log10(x) lôgarit cơ số 10 của x.

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases}$$

$$\text{Max}(x, y) = \begin{cases} x & ; \quad x \geq y \\ y & ; \quad x < y \end{cases}$$

$$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0,5)$$

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x > 0 \\ 0 & ; \quad x == 0 \\ -1 & ; \quad x < 0 \end{cases}$$

Sin(x) hàm lượng giác sin hoạt động trên đối số x ở các đơn vị radian

$$\text{Sqrt}(x) = \sqrt{x}$$

$$\text{Swap}(x, y) = (y, x)$$

Tan(x) hàm lượng giác tang hoạt động trên đối số x ở các đơn vị radian

Thứ tự ưu tiên phép toán

Khi thứ tự ưu tiên trong biểu thức không được chỉ báo tường minh nhờ sử dụng các dấu ngoặc, các quy tắc sau áp dụng:

- Các phép toán có độ ưu tiên cao hơn được đánh giá trước phép toán bất kỳ có độ ưu tiên thấp hơn.
- Các phép toán có cùng độ ưu tiên được đánh giá lần lượt từ trái sang phải.

Bảng dưới đây xác định độ ưu tiên của các phép toán từ cao nhất đến thấp nhất; vị trí cao hơn trong bảng chỉ báo độ ưu tiên cao hơn.

Đối với các toán tử đó cũng được sử dụng trong ngôn ngữ lập trình C, thứ tự độ ưu tiên được sử dụng trong bản mô tả giống như được sử dụng trong ngôn ngữ lập trình C.

Bảng: Độ ưu tiên hoạt động từ cao nhất (trên cùng bảng) đến thấp nhất (dưới cùng bảng)

Phép toán (với các toán hạng x, y, và z)
"x++", "x--"
"!x", "-x" (như là toán tử tiền tố đơn nguyên)
x^y
"x * y", "x/y", "x ÷ y", " $\frac{x}{y}$ ", "x % y"
"x + y", "x - y" (như là toán tử hai đối số), " $\sum_{i=x}^y f(i)$ "
">>", "<<"
"x < y", "x <= y", "x > y", "x >= y"
"x == y", "x != y"
"x & y"
"x y"
"x && y"
"x y"
"x ? y : z"
"x..y"
"x = y", "x += y", "x -= y"

Mô tả văn bản các phép toán logic

Trong văn bản, mệnh đề của các phép toán logic như được mô tả dạng toán học trong biểu thức sau:

If (điều kiện 0)

```

    mệnh đề 0
else if (điều kiện 1)
    mệnh đề 1
...
else /* đánh dấu mang thông tin về điều kiện còn lại */
    mệnh đề n

```

có thể được mô tả theo cách thức sau:

... như sau / ... điều kiện sau áp dụng:

- If điều kiện 0, mệnh đề 0
- Ngược lại, if điều kiện 1, mệnh đề 1
- ...
- Ngược lại (đánh dấu mang thông tin về điều kiện còn lại), mệnh đề n

Mỗi mệnh đề “If ... Otherwise, if ... Otherwise, ...” trong văn bản được đưa vào với “... như sau” hoặc “... điều kiện sau áp dụng” theo ngay sau bởi “If ... “. Điều kiện cuối cùng của “If ... Otherwise, if ... Otherwise, ...” luôn là “Ngược lại, ...”. Các mệnh đề “If ... Otherwise, if ... Otherwise, ...” được đan xen có thể được nhận diện bằng cách so khớp “... như sau” hoặc “... điều kiện sau áp dụng” với kết thúc “Otherwise, ...”.

Trong văn bản, mệnh đề của các phép toán logic như sẽ được mô tả dưới dạng toán học trong biểu thức sau:

```

if(điều kiện 0a && điều kiện 0b )
    mệnh đề 0
else if( điều kiện 1a || điều kiện 1b )
    mệnh đề 1
...
else
    mệnh đề n

```

có thể được mô tả theo cách thức sau:

... như sau / ... điều kiện sau áp dụng:

- Nếu tất cả các điều kiện sau là đúng, mệnh đề 0:
- điều kiện 0a
- điều kiện 0b
- Ngược lại, nếu một hoặc các điều kiện sau là đúng, mệnh đề 1:
- điều kiện 1a
- điều kiện 1b
- ...
- Ngược lại, mệnh đề n

Trong văn bản, mệnh đề của các phép toán logic như sẽ được mô tả dưới dạng toán học trong biểu thức sau:

if(điều kiện 0)

mệnh đề 0

if(điều kiện 1)

mệnh đề 1

có thể được mô tả theo cách thức sau:

Khi điều kiện 0, mệnh đề 0

Khi điều kiện 1, mệnh đề 1.

Mặc dù các phương án thực hiện sáng chế đã được mô tả chủ yếu dựa trên tạo mã video, cần lưu ý rằng các phương án thực hiện hệ thống tạo mã 10, bộ mã hóa 20 lần bộ giải mã 30 (và tương ứng hệ thống 10) và các phương án thực hiện khác được nêu ở đây có thể cũng được tạo cấu hình để xử lý hoặc tạo mã ảnh tĩnh, tức là, xử lý hoặc tạo mã ảnh riêng rẽ độc lập với ảnh đứng trước hoặc liên tục như khi tạo mã video. Nói chung, chỉ các khối dự báo ngoài 244 (bộ mã hóa) và 344 (bộ giải mã) có thể không khả dụng trong trường hợp mã hóa xử lý ảnh bị giới hạn ở một ảnh 17. Tất cả các chức năng khác (cũng được gọi là các công cụ hoặc công nghệ) của bộ mã hóa video 20 và bộ giải mã video 30 có thể được sử dụng cân bằng để xử lý ảnh tĩnh, chẳng hạn, tính toán phần dư 204/304, biến đổi 206, lượng tử hóa 208, lượng tử hóa đảo 210/310, biến đổi (ngược) 212/312, phân vùng 262/362, dự báo trong 254/354, và/hoặc lọc vòng 220, 320, và mã hóa entropy 270 và giải mã entropy 304.

Các phương án thực hiện, chẳng hạn, của bộ mã hóa 20 và bộ giải mã 30, và các chức năng được nêu ở đây, chẳng hạn, dựa vào bộ mã hóa 20 và bộ giải mã 30, có thể được triển khai trong phần cứng, phần mềm, firmware, hoặc tổ hợp bất kỳ của nó. Nếu được triển khai trong phần mềm, các chức năng có thể được lưu trữ trên vật máy tính đọc được hoặc được truyền trên phương tiện truyền thông dưới dạng một hoặc nhiều lệnh hoặc mã và được thực thi bởi khối xử lý dựa trên phần cứng. Phương tiện máy tính đọc được có thể bao gồm phương tiện lưu trữ máy tính đọc được, tương ứng với phương tiện hữu hình chẳng hạn phương tiện lưu trữ dữ liệu, hoặc phương tiện truyền thông bao gồm phương tiện bất kỳ để truyền chương trình máy tính từ một điểm sang điểm khác, chẳng hạn, theo giao thức truyền thông. Theo cách này, phương tiện máy tính đọc được nói chung có thể tương ứng với (1) phương tiện lưu trữ máy tính đọc được hữu hình vốn là bất biến hoặc (2) phương tiện truyền thông chẳng hạn tín hiệu hoặc sóng mang. Phương tiện lưu trữ dữ liệu có thể là phương tiện khả dụng bất kỳ có thể được truy nhập bởi một hoặc nhiều máy tính hoặc một hoặc nhiều bộ xử lý để truy xuất lệnh, mã và/hoặc các cấu trúc dữ liệu để thực hiện các kỹ thuật này được mô tả theo sáng chế. Sản phẩm chương trình máy tính có thể bao gồm phương tiện máy tính đọc được.

Thông qua ví dụ và không giới hạn, phương tiện lưu trữ máy tính đọc được này có thể bao gồm RAM, ROM, EEPROM, CD-ROM hoặc bộ lưu trữ đĩa quang, bộ lưu trữ đĩa từ, hoặc các bộ lưu trữ từ tính khác, bộ nhớ nhanh, hoặc phương tiện khác bất kỳ có thể được sử dụng để lưu trữ mã chương trình mong muốn ở dạng các lệnh hoặc các cấu trúc dữ liệu và có thể được truy nhập bằng máy tính. Kết nối bất kỳ cũng được đặt tên đúng phương tiện máy tính đọc được. Chẳng hạn, nếu các lệnh được truyền từ website, máy chủ, hoặc nguồn từ xa khác nhờ sử dụng cáp đồng trục, cáp sợi quang, cặp xoắn, đường thuê bao số (digital subscriber line, DSL), hoặc công nghệ không dây chẳng hạn hồng ngoại, vô tuyến, và vi sóng, sau đó cáp đồng trục, cáp sợi quang, cặp xoắn, DSL, hoặc công nghệ không dây chẳng hạn hồng ngoại, vô tuyến, và vi sóng được bao gồm trong định nghĩa về môi trường. Tuy nhiên, cần hiểu rằng phương tiện lưu trữ máy tính đọc được và phương tiện lưu trữ dữ liệu không

bao gồm các kết nối, các sóng mang, các tín hiệu, hoặc các phương tiện khả biến khác, nhưng thay vào đó hướng đến các phương tiện lưu trữ hữu hình, bất biến. Disk và disc, như được sử dụng ở đây, bao gồm compact disc (CD), đĩa laze, đĩa quang, đĩa đa dụng số (digital versatile disc, DVD), đĩa mềm và đĩa Blu-ray, trong đó các disk thường tái tạo dữ liệu dạng từ tính, trong khi disc thường tái tạo dữ liệu dạng quang học bằng các tia laze. Các tổ hợp nêu trên cũng cần được bao gồm trong phạm vi của phương tiện máy tính đọc được.

Các lệnh có thể được thực thi bằng một hoặc nhiều bộ xử lý, chẳng hạn một hoặc nhiều bộ xử lý tín hiệu số (digital signal processor, DSP), bộ vi xử lý đa năng, mạch tích hợp ứng dụng cụ thể (application specific integrated circuit, ASIC), mảng cổng dạng trường lập trình được (field programmable logic array, FPGA), hoặc hệ mạch logic rời rạc hoặc tích hợp tương đương khác. Do đó, cụm từ “bộ xử lý,” như được sử dụng ở đây có thể đề cập đến cấu trúc bất kỳ trong các cấu trúc nêu trên hoặc cấu trúc khác thích hợp để thực hiện các kỹ thuật được nêu ở đây. Ngoài ra, theo một số khía cạnh, chức năng được nêu ở đây có thể được đặt trong phần cứng và/hoặc môđun phần mềm dành riêng được tạo cấu hình để mã hóa và giải mã, hoặc được đưa vào codec kết hợp. Các kỹ thuật cũng có thể được triển khai đầy đủ trong một hoặc nhiều mạch hoặc phần tử logic.

Các kỹ thuật của sáng chế có thể được triển khai trong các bộ phận hoặc thiết bị, bao gồm điện thoại cầm tay không dây, mạch tích hợp (integrated circuit, IC) hoặc tập hợp IC (chẳng hạn, chip set). Các thành phần, các môđun hoặc các khối khác nhau được mô tả theo sáng chế để nhấn mạnh các khía cạnh chức năng của các bộ phận được tạo cấu hình để thực hiện các kỹ thuật được bộc lộ, mà không nhất thiết yêu cầu thực hiện bằng các khối phần cứng khác nhau. Thay vào đó, như được nêu trên, các khối khác nhau có thể được kết hợp trong khối phần cứng của codec hoặc được tạo bởi nhóm các khối phần cứng liên tác, bao gồm một hoặc nhiều bộ xử lý như được nêu trên, cùng với phần mềm và/hoặc firmware thích hợp.

YÊU CẦU BẢO HỘ

1. Phương pháp tạo danh sách ứng viên hợp nhất đối với chế độ sao chép trong khối (Intra block copy, IBC), trong đó phương pháp bao gồm các bước:

chèn vector khối của khối lân cận bên trái của khối hiện tại vào danh sách hợp nhất ban đầu của khối hiện tại, khi khối lân cận bên trái khả dụng và khối lân cận bên trái sử dụng chế độ IBC;

chèn vector khối của khối lân cận bên trên của khối hiện tại vào danh sách hợp nhất ban đầu, khi khối lân cận bên trên là khả dụng, khối lân cận bên trên đang sử dụng chế độ IBC và vector khối của khối lân cận bên trên không giống như vector khối của khối lân cận bên trái;

chèn vector khối của ứng viên cuối cùng trong bộ dự báo vector chuyển động dựa trên lịch sử (history based motion vector predictor, HMVP) và danh sách hợp nhất ban đầu, khi vector khối của khối lân cận bên trên không giống như vector khối của ứng viên cuối cùng trong HMVP và khi vector khối của khối lân cận bên trái không giống như vector khối của ứng viên cuối cùng trong HMVP; và

chèn vector khối của ứng viên áp chót trong HMVP và danh sách hợp nhất ban đầu, trong đó không thực hiện tia đối với vector khối của ứng viên áp chót trong HMVP, và

trong đó vector khối cuối cùng trong danh sách hợp nhất ban đầu của khối hiện tại là vector khối của ứng viên trong HMVP.

2. Bộ giải mã bao gồm:

một hoặc nhiều bộ xử lý; và

vật ghi máy tính đọc được bất biến được ghép nối với một hoặc nhiều bộ xử lý và lưu trữ chương trình để thực thi bởi một hoặc nhiều bộ xử lý, trong đó chương trình, khi được thực thi bởi một hoặc nhiều bộ xử lý, tạo cấu hình bộ giải mã để

chèn vector khối của khối lân cận bên trái của khối hiện tại vào danh sách hợp nhất ban đầu của khối hiện tại, khi khối lân cận bên trái khả dụng và khối lân cận bên trái sử dụng chế độ IBC;

chèn vector khối của khối lân cận bên trên của khối hiện tại vào danh sách

hợp nhất ban đầu, khi khối lân cận bên trên là khả dụng, khối lân cận bên trên đang sử dụng chế độ IBC và vector khối của khối lân cận bên trên không giống như vector khối của khối lân cận bên trái;

chèn vector khối của ứng viên cuối cùng trong HMVP và danh sách hợp nhất ban đầu, khi vector khối của khối lân cận bên trên không giống như vector khối của ứng viên cuối cùng trong HMVP và khi vector khối của khối lân cận bên trái không giống như vector khối của ứng viên cuối cùng trong HMVP; và

chèn vector khối của ứng viên áp chót trong HMVP và danh sách hợp nhất ban đầu, trong đó không thực hiện tia đối với vector khối của ứng viên áp chót trong HMVP, và

trong đó vector khối cuối cùng trong danh sách hợp nhất ban đầu của khối hiện tại là vector khối của ứng viên trong HMVP.

3. Bộ giải mã bao gồm:

một hoặc nhiều bộ xử lý; và

vật ghi máy tính đọc được bất biến được ghép nối với một hoặc nhiều bộ xử lý và lưu trữ chương trình để thực thi bởi một hoặc nhiều bộ xử lý, trong đó chương trình, khi được thực thi bởi một hoặc nhiều bộ xử lý, tạo cấu hình bộ giải mã để

chèn vector khối của khối lân cận của khối hiện tại vào danh sách hợp nhất ban đầu của khối hiện tại, khi khối lân cận khả dụng và khối lân cận sử dụng chế độ IBC;

chèn vector khối của ứng viên cuối cùng trong HMVP và danh sách hợp nhất ban đầu, khi vector khối của khối lân cận không giống như vector khối của ứng viên cuối cùng trong HMVP; và

chèn vector khối của ứng viên áp chót trong HMVP và danh sách hợp nhất ban đầu, trong đó không thực hiện tia đối với vector khối của ứng viên áp chót trong HMVP.

4. Bộ giải mã theo điểm 3, trong đó một hoặc nhiều bộ xử lý còn được tạo cấu hình để:

thu được vector khối của khối hiện tại theo danh sách hợp nhất ban đầu sau các quá trình chèn nêu trên và chế độ ứng viên hợp nhất cho khối hiện tại.

5. Bộ giải mã bao gồm:

một hoặc nhiều bộ xử lý; và

vật ghi máy tính đọc được bất biến được ghép nối với một hoặc nhiều bộ xử lý và lưu trữ chương trình để thực thi bởi một hoặc nhiều bộ xử lý, trong đó chương trình, khi được thực thi bởi một hoặc nhiều bộ xử lý, tạo cấu hình bộ giải mã để

chèn vector khối của khối lân cận bên trái của khối hiện tại vào danh sách hợp nhất ban đầu của khối hiện tại, khi khối lân cận bên trái khả dụng và khối lân cận bên trái sử dụng chế độ IBC;

chèn vector khối của khối lân cận bên trên của khối hiện tại vào danh sách hợp nhất ban đầu, khi khối lân cận bên trên là khả dụng, khối lân cận bên trên đang sử dụng chế độ IBC và vector khối của khối lân cận bên trên không giống như vector khối của khối lân cận bên trái;

chèn vector khối của ứng viên cuối cùng trong HMVP và danh sách hợp nhất ban đầu, khi vector khối của khối lân cận bên trên không giống như vector khối của ứng viên cuối cùng trong HMVP và khi vector khối của khối lân cận bên trái không giống như vector khối của ứng viên cuối cùng trong HMVP; và

chèn vector khối của ứng viên áp chót trong HMVP và danh sách hợp nhất ban đầu, trong đó không thực hiện tia đối với vector khối của ứng viên áp chót trong HMVP.

6. Bộ giải mã theo điểm 5, trong đó các bộ xử lý còn được tạo cấu hình để thu được vector khối của khối hiện tại theo danh sách hợp nhất ban đầu sau các quá trình chèn nêu trên và chế độ ứng viên hợp nhất cho khối hiện tại.

7. Bộ giải mã theo điểm 6, trong đó danh sách hợp nhất ban đầu là danh sách trống trước quá trình chèn thứ nhất.

8. Bộ giải mã theo điểm 6, trong đó các quá trình chèn nêu trên được thực hiện theo thứ tự.

9. Bộ giải mã theo điểm 5, trong đó danh sách hợp nhất ban đầu là danh sách trống trước quá trình chèn thứ nhất.

10. Bộ giải mã theo điểm 9, trong đó các quá trình chèn nêu trên được thực hiện theo thứ tự.

11. Bộ giải mã theo điểm 5, trong đó các quá trình chèn nêu trên được thực hiện theo thứ tự.

12. Vật ghi bất biến lưu trữ mã chương trình để thực hiện các hoạt động bao gồm:

chèn vector khối của khối lân cận bên trái của khối hiện tại vào danh sách hợp nhất ban đầu của khối hiện tại, khi khối lân cận bên trái khả dụng và khối lân cận bên trái sử dụng chế độ IBC;

chèn vector khối của khối lân cận bên trên của khối hiện tại và danh sách hợp nhất ban đầu, khi khối lân cận bên trên là khả dụng, khối lân cận bên trên đang sử dụng chế độ IBC và vector khối của khối lân cận bên trên không giống như vector khối của khối lân cận bên trái;

chèn vector khối của ứng viên cuối cùng trong HMVP và danh sách hợp nhất ban đầu, khi vector khối của khối lân cận bên trên không giống như vector khối của ứng viên cuối cùng trong HMVP và khi vector khối của khối lân cận bên trái không giống như vector khối của ứng viên cuối cùng trong HMVP; và

chèn vector khối của ứng viên áp chót trong HMVP và danh sách hợp nhất ban đầu, trong đó không thực hiện tia đối với vector khối của ứng viên áp chót trong HMVP, và trong đó vector khối cuối cùng trong danh sách hợp nhất ban đầu của khối hiện tại là vector khối của ứng viên trong HMVP.

1/10

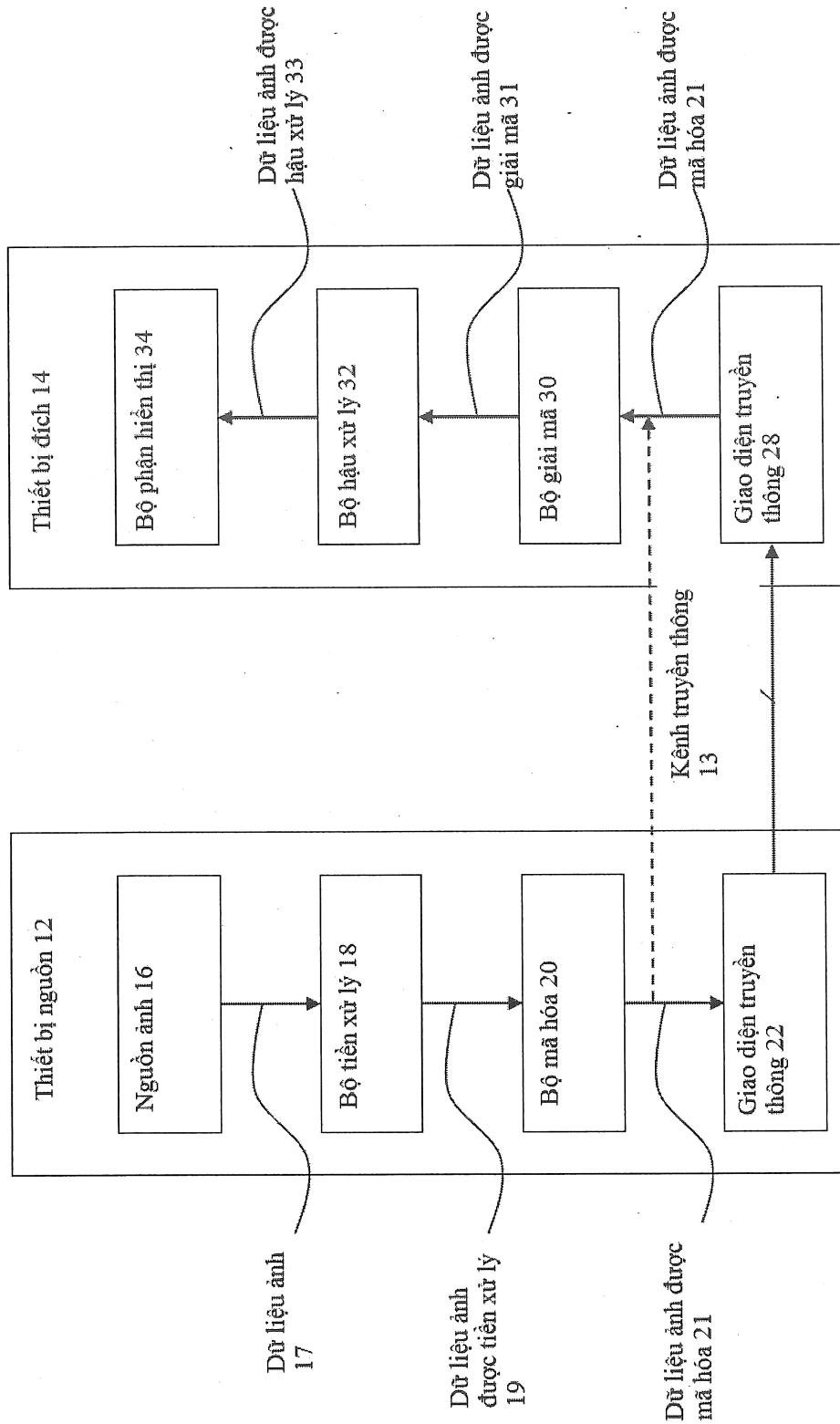


Fig.1A

10

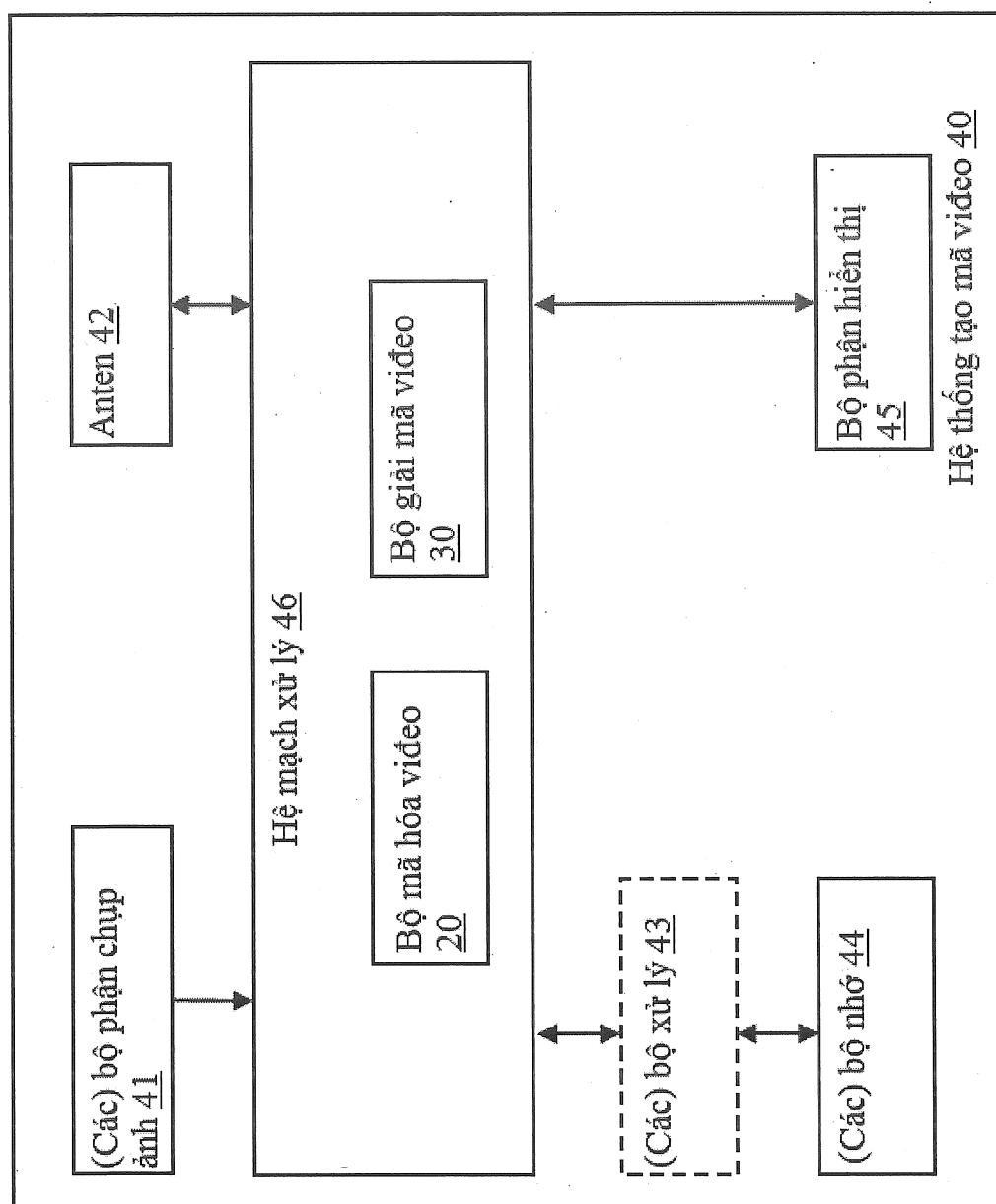


Fig.1B



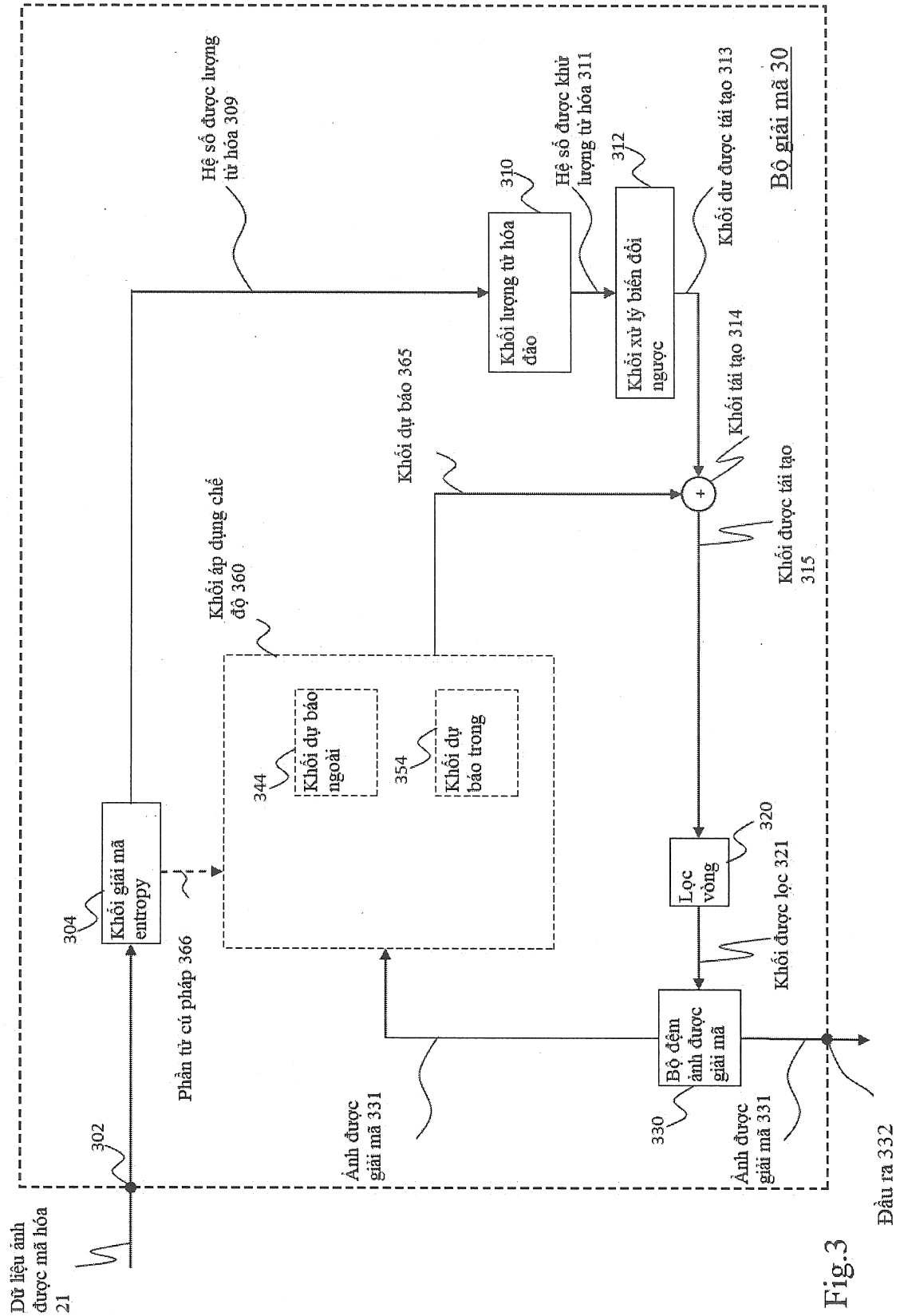


Fig.3

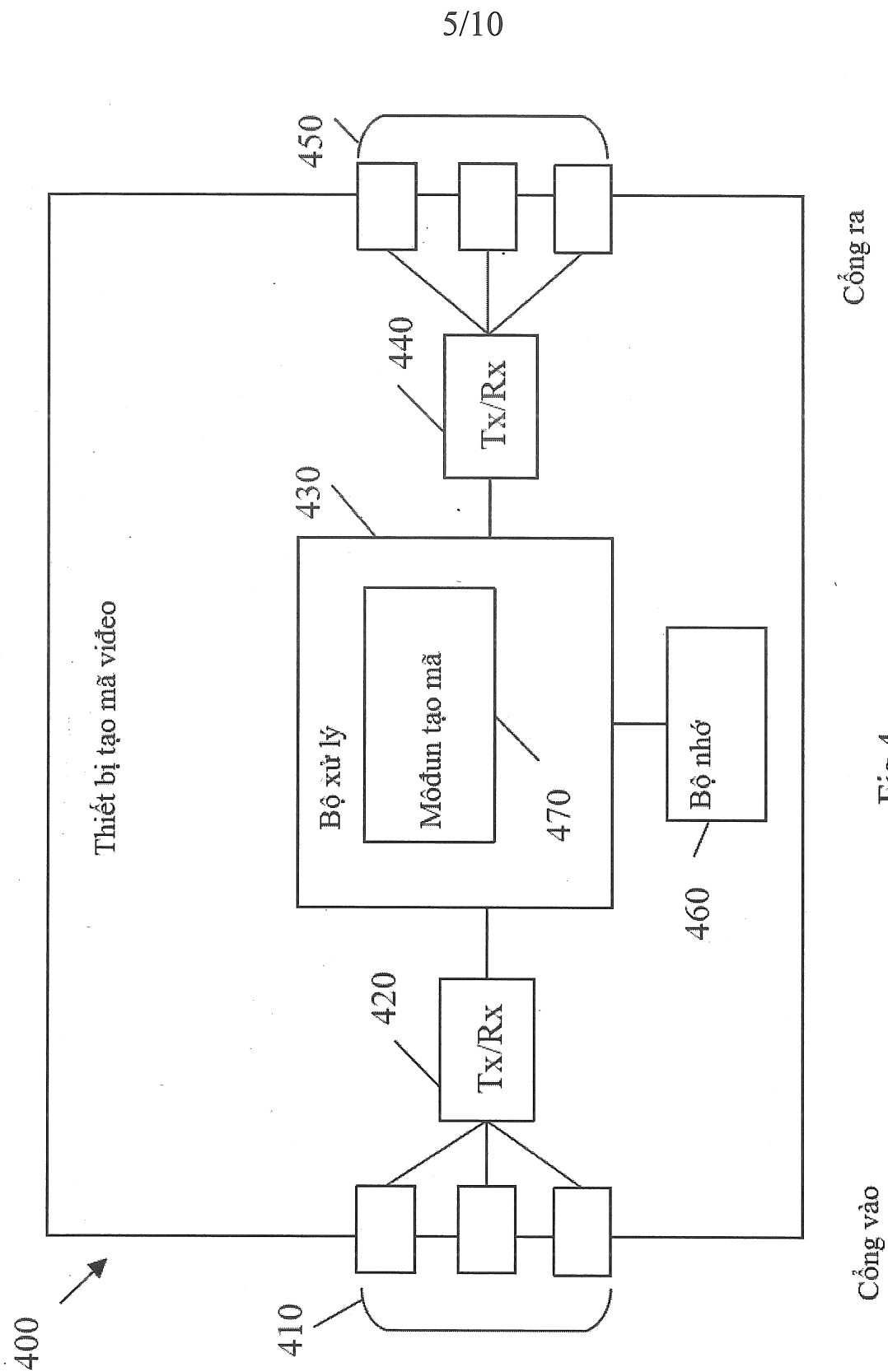


Fig.4

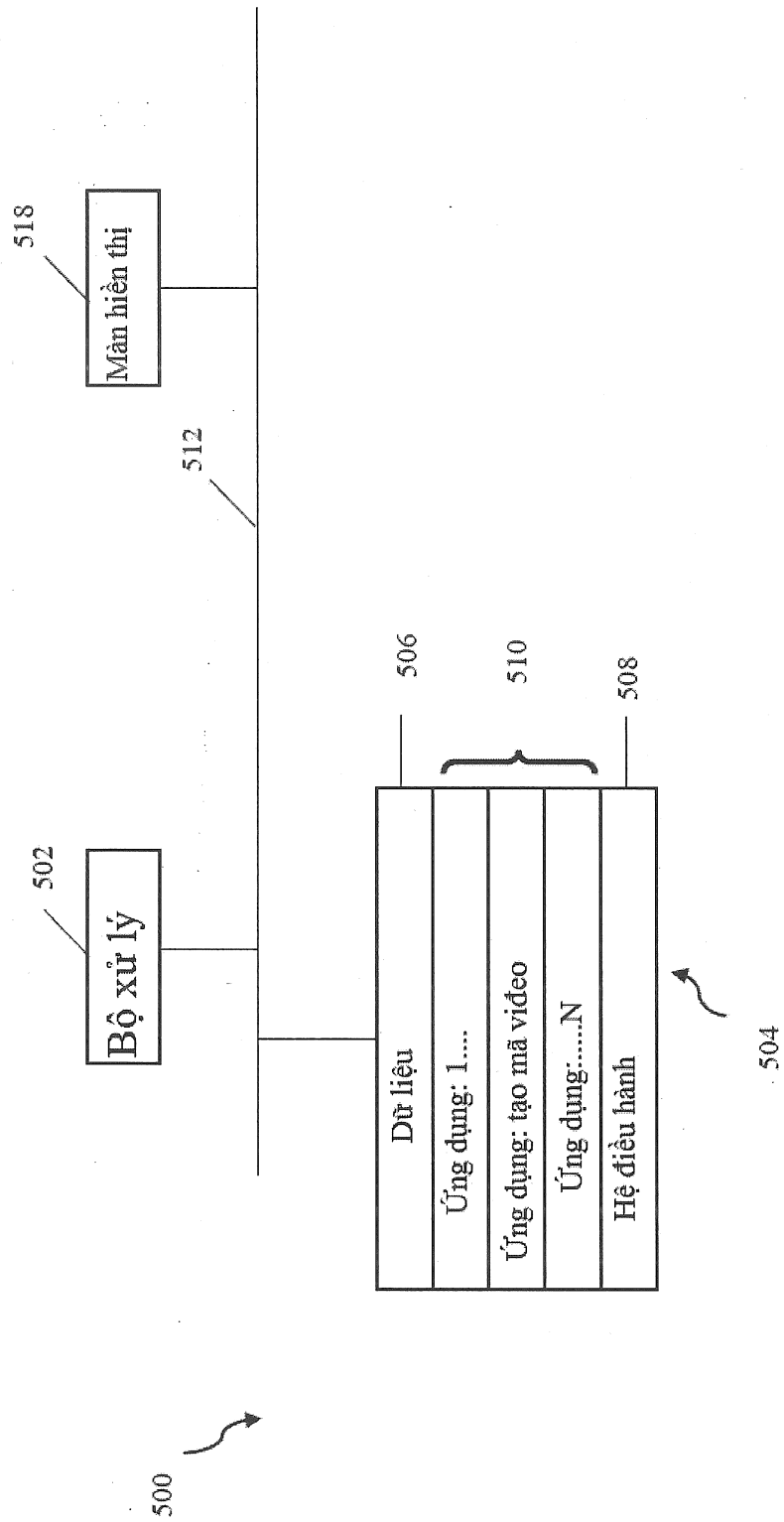


Fig.5

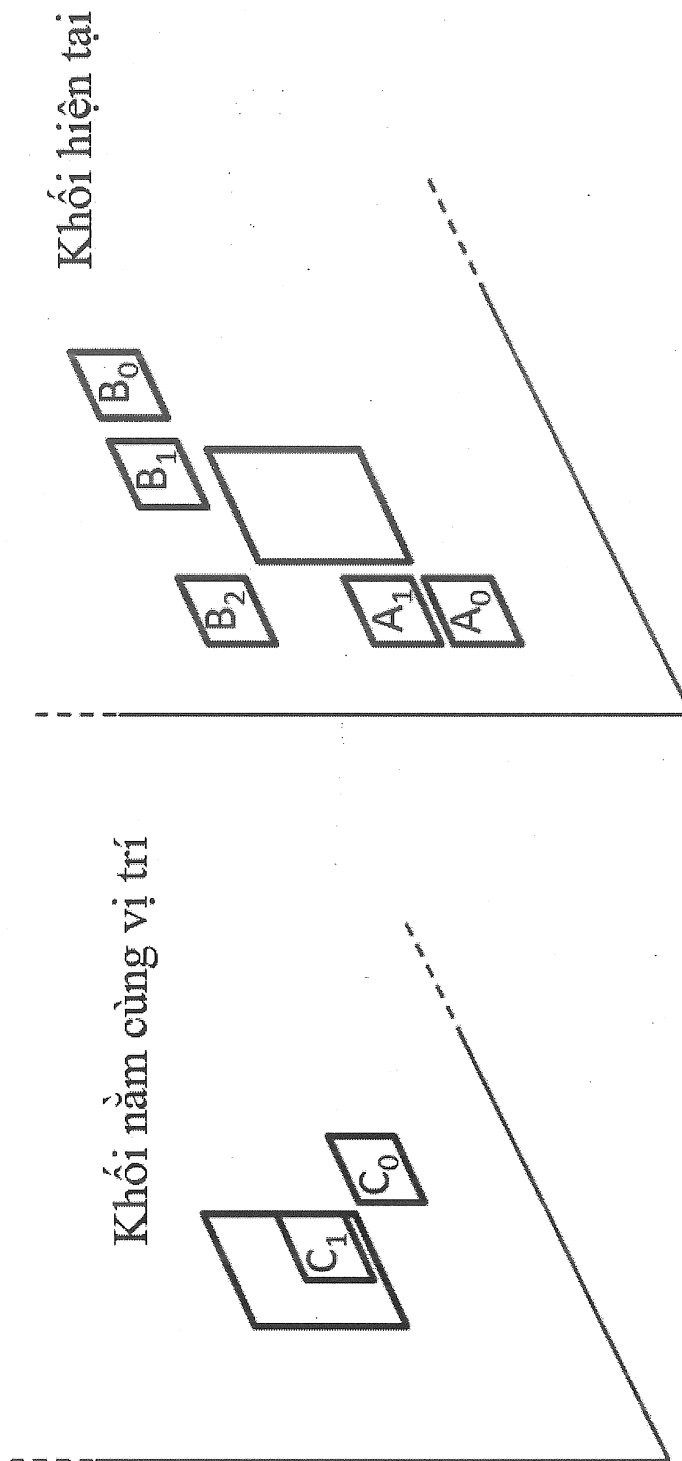


Fig.6

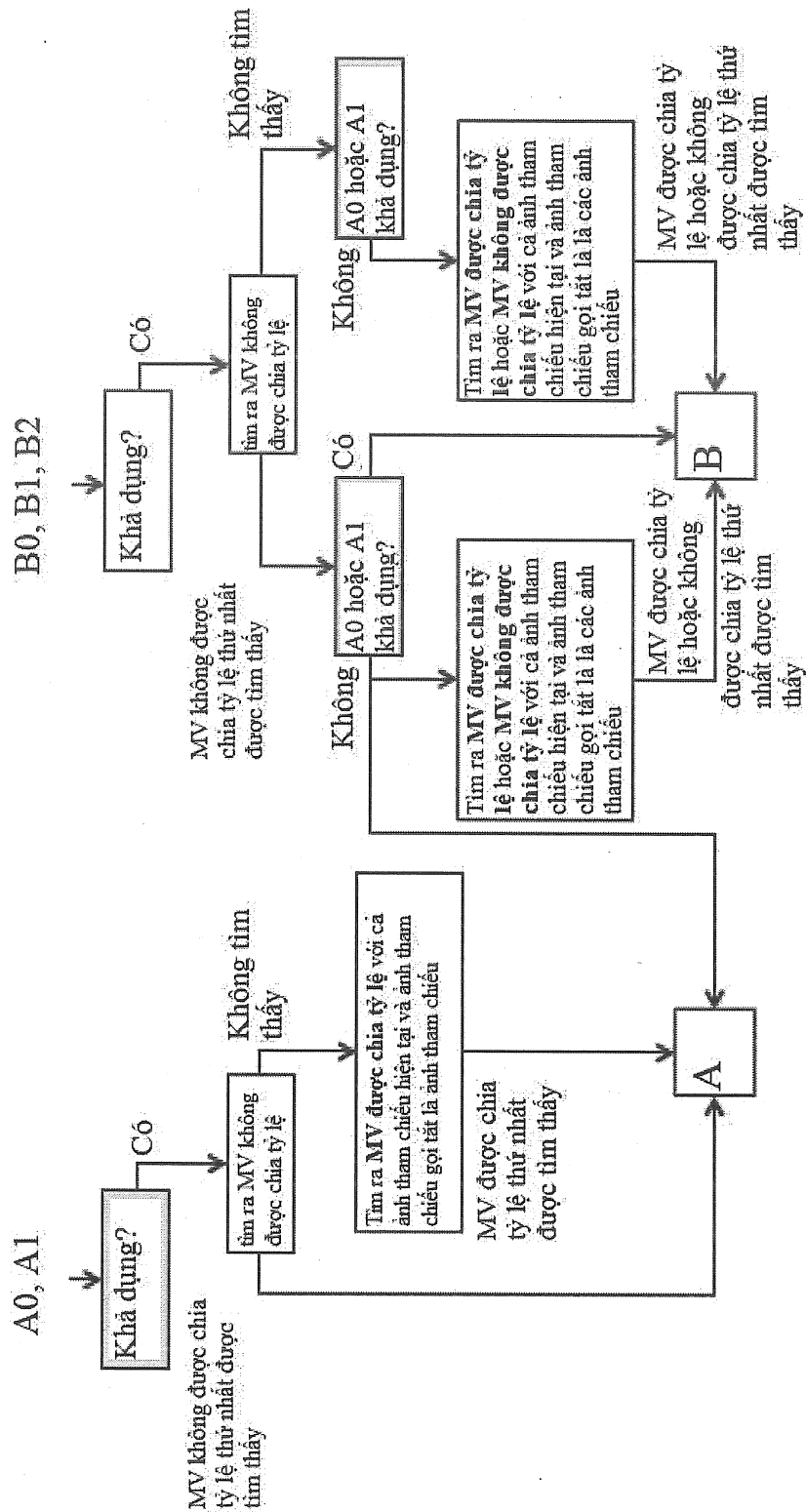


Fig. 7

9/10

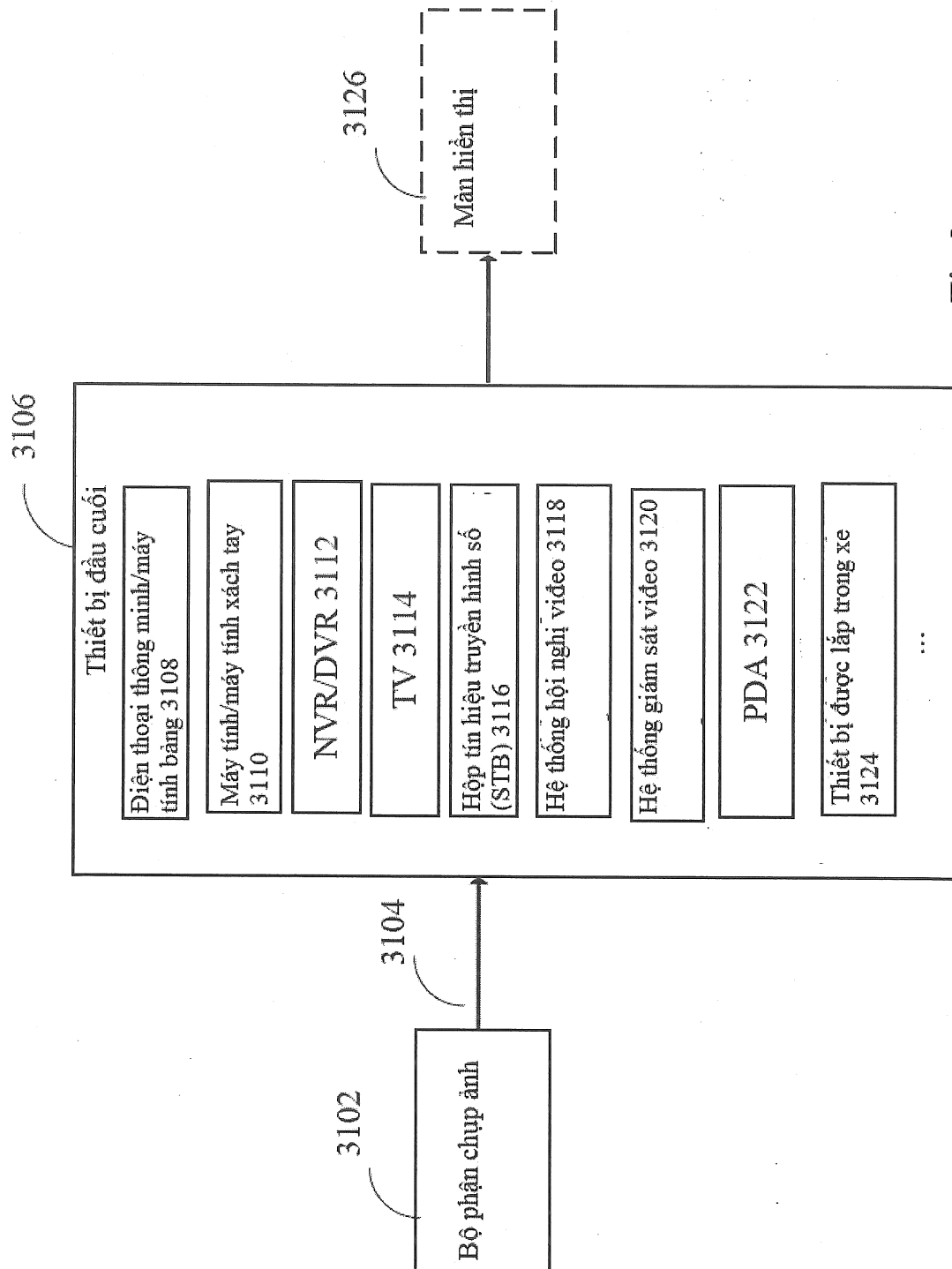


Fig. 8

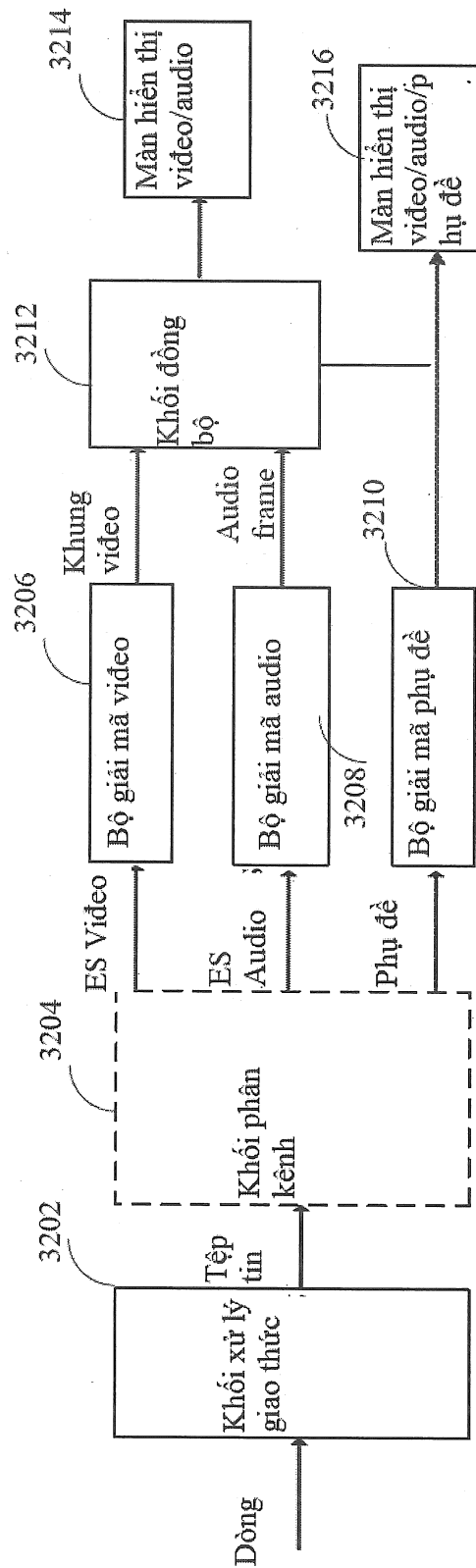


Fig.9