



(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)



1-0045071

(51)^{2020.01} H04N 19/159; G06T 7/579; H04N
19/176; G06T 7/529; H04N 19/105

(13) B

(21) 1-2021-07406

(22) 24/04/2020

(86) PCT/US2020/029906 24/04/2020

(87) WO2020/219940 29/10/2020

(30) 62/838,618 25/04/2019 US

(45) 25/04/2025 445

(43) 25/02/2022 407A

(71) OP SOLUTIONS, LLC (US)

368 Middle Street, Amherst, Massachusetts 01002, United States of America

(72) KALVA, Hari (IN); FURHT, Borivoje (US); ADZIC, Velibor (US).

(74) Công ty Luật TNHH T&G (TGVN)

(54) BỘ GIẢI MÃ VÀ PHƯƠNG PHÁP GIẢI MÃ

(21) 1-2021-07406

(57) Sáng chế đề cập tới bộ giải mã chứa hệ mạch được đặt cấu hình để nhận luồng bit; xác định, cho khối hiện tại và sử dụng luồng bit, rằng chế độ hợp nhất được cho phép thực hiện; bước xây dựng danh sách ứng viên hợp nhất chứa việc thêm vectơ chuyển động toàn cục vào danh sách ứng viên vectơ chuyển động; và xây dựng lại dữ liệu điểm ảnh của khối hiện tại và sử dụng danh sách ứng viên vectơ chuyển động. Sáng chế cũng đề xuất phương pháp giải mã.

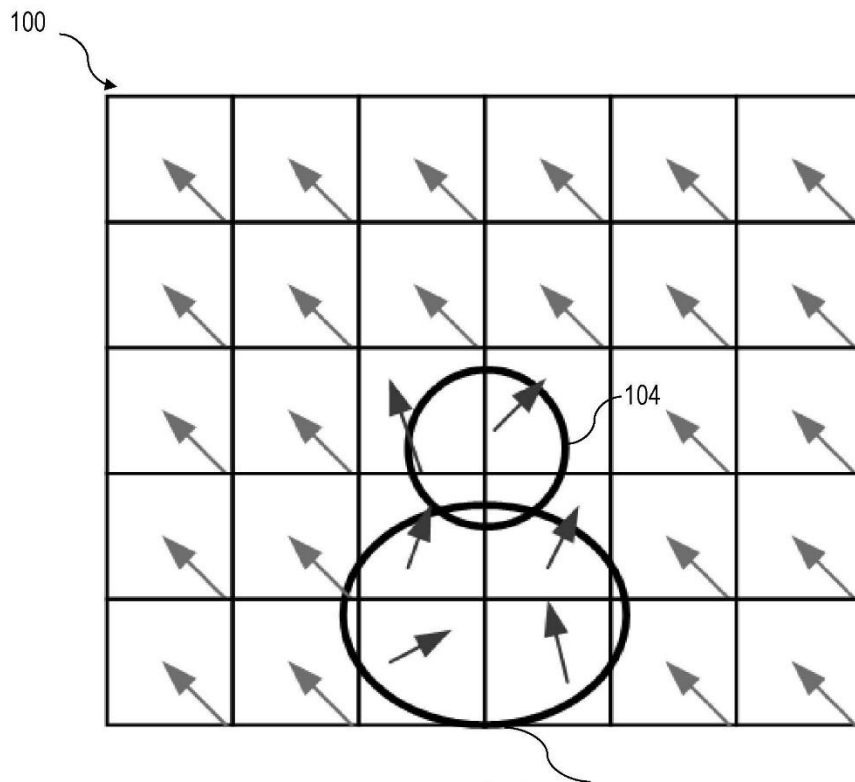


FIG. 1

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập chung tới lĩnh vực nén video. Cụ thể là, sáng chế này đề cập tới chuyển động toàn cục cho các ứng viên chế độ hợp nhất trong việc dự đoán liên thông.

Tình trạng kỹ thuật của sáng chế

Bộ mã hóa-giải mã video có thể chứa mạch điện tử hoặc phần mềm để nén hoặc giải nén video kỹ thuật số. Nó có thể biến đổi video không được nén thành định dạng được nén hoặc ngược lại. Trong văn cảnh của việc nén video, thiết bị nén video (và/hoặc thực hiện một số chức năng của chúng) thông thường có thể được gọi là bộ ghi mã, và thiết bị giải nén video (và/hoặc thực hiện một số chức năng của chúng) có thể được gọi là bộ giải mã.

Định dạng của dữ liệu được nén có thể tuân theo mô tả nén video tiêu chuẩn. Việc nén có thể là bị mất mát ở chỗ video được nén thiếu một số thông tin hiện có mặt trong video gốc. Trình tự của nó có thể chứa phần video được giải nén có thể có chất lượng thấp hơn video không được nén gốc bởi vì có thông tin không đủ để tái tạo một cách chính xác video gốc.

Có thể có các quan hệ phức tạp giữa chất lượng video, lượng của dữ liệu được sử dụng để biểu diễn video (ví dụ, được xác định bởi tốc độ bit), độ phức tạp của các thuật toán ghi mã và giải mã, độ nhạy với các tổn thất và lỗi dữ liệu, dễ chỉnh sửa, truy cập ngẫu nhiên, trễ từ đầu cuối tới đầu cuối (ví dụ, độ trễ), và dạng tương tự.

Việc bù chuyển động có thể chứa cách tiếp cận để dự đoán khung video hoặc một phần của nó cho khung tham chiếu đã cho, như các khung trước đó và/hoặc các khung tương lai, bằng cách xem xét tới chuyển động của camera và/hoặc các đối tượng của video. Nó có thể được áp dụng trong việc ghi mã và giải mã của dữ liệu video cho việc nén video, ví dụ trong việc ghi mã và giải mã sử dụng tiêu chuẩn nhóm các chuyên gia hình ảnh động (Motion Picture Experts Group - MPEG)-2 (ngoài ra được gọi là mã hóa video tiên tiến (advanced video coding - AVC)). Việc bù chuyển động có thể mô tả ảnh theo nghĩa của

việc biến đổi của ảnh tham chiếu thành ảnh hiện tại. Ảnh tham chiếu có thể là ở trước đó theo thời gian khi so sánh với ảnh hiện tại, từ tương lai khi so sánh với ảnh hiện tại. Khi các hình ảnh có thể được tổng hợp một cách chính xác từ các hình ảnh được truyền và/hoặc được lưu trữ trước đó, thì hiệu quả nên có thể được cải thiện.

Bản chất kỹ thuật của sáng chế

Theo một khía cạnh, bộ giải mã chứa hệ mạch được đặt cấu hình để nhận luồng bit, xác định, cho khối hiện tại và sử dụng luồng bit, mà chế độ hợp nhất được cho phép thực hiện, xây dựng danh sách ứng viên chế độ hợp nhất, trong đó việc xây dựng danh sách ứng viên hợp nhất còn bao gồm việc thêm vectơ chuyển động toàn cục vào danh sách ứng viên vectơ chuyển động, và xây dựng lại dữ liệu điểm ảnh của khối hiện tại sử dụng danh sách ứng viên vectơ chuyển động.

Theo khía cạnh khác, phương pháp chứa bước nhận, bởi bộ giải mã, luồng bit, xác định, cho khối hiện tại và sử dụng luồng bit, rằng chế độ hợp nhất được cho phép thực hiện, xây dựng danh sách ứng viên hợp nhất, trong đó bước xây dựng danh sách ứng viên hợp nhất còn bao gồm việc thêm vectơ chuyển động toàn cục vào danh sách ứng viên vectơ chuyển động, và xây dựng lại dữ liệu điểm ảnh của khối hiện tại và sử dụng danh sách ứng viên vectơ chuyển động.

Các chi tiết của một hoặc nhiều biến thể của đối tượng được mô tả ở đây được thể hiện trong các hình vẽ kèm theo và trong phần mô tả bên dưới. Các dấu hiệu và các ưu điểm khác của đối tượng của sáng chế được mô tả trong bản mô tả này sẽ trở nên rõ ràng từ phần mô tả và các hình vẽ, và từ các điểm yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

Với mục đích minh họa sáng chế, các hình vẽ thể hiện các khía cạnh của một hoặc nhiều phương án thực hiện của sáng chế. Tuy nhiên, cần hiểu rằng sáng chế này không bị hạn chế vào các sắp xếp chính xác và các phương tiện chính xác được thể hiện trên các hình vẽ, trong đó:

Fig.1 là giản đồ đang minh họa các vectơ chuyển động của khung làm ví dụ với chuyển động toàn cục và cục bộ;

Fig.2 minh họa ba mô hình chuyển động làm ví dụ có thể được tận dụng cho chuyển động toàn cục chứa các trị số chỉ số của chúng (0, 1, hoặc 2);

Fig.3 là giản đồ tiến trình công nghệ theo một số áp dụng làm ví dụ của đối tượng hiện tại;

Fig.4 là giản đồ khối của bộ giải mã làm ví dụ theo một số ứng dụng làm ví dụ của đối tượng hiện tại;

Fig.5 là giản đồ tiến trình công nghệ theo một số áp dụng làm ví dụ của đối tượng hiện tại;

Fig.6 là giản đồ khối hệ thống của bộ ghi mã làm ví dụ theo một số ứng dụng làm ví dụ của đối tượng hiện tại; và

Fig.7 là giản đồ khối của hệ thống tính toán có thể được sử dụng để áp dụng một hoặc nhiều phương pháp bất kỳ trong số các phương pháp được bộc lộ ở đây và một hoặc nhiều phần của chúng.

Các hình vẽ không nhất thiết phải cùng một tỉ lệ và có thể được minh họa bởi các đường ảo, các biểu diễn giản đồ và các hình phân mảnh. Theo các trường hợp cụ thể, các chi tiết không cần thiết cho việc hiểu các phương án thực hiện hoặc làm cho các chi tiết khác trở nên khó nhận biết, có thể được bỏ qua. Các ký hiệu tham chiếu giống nhau trong các hình vẽ khác nhau chỉ tới các phần tử giống nhau.

Mô tả chi tiết sáng chế

Chuyển động toàn cục trong video đề cập tới chuyển động xuất hiện trong toàn bộ khung. Chuyển động toàn cục có thể được gây ra bởi chuyển động camera, ví dụ, việc lia và phóng to thu nhỏ camera tạo ra chuyển động trong khung có thể thường ảnh hưởng tới toàn bộ khung. Chuyển động có mặt trong các phần của video có thể được đề cập tới là chuyển động cục bộ. Chuyển động cục bộ có thể được gây ra bởi các vật thể đang di chuyển

trong cảnh. Ví dụ, vật thể đang di chuyển từ trái sang phải trong cảnh. Các video có thể chứa tổ hợp của các chuyển động cục bộ và chuyển động toàn cục. Một số áp dụng của đối tượng hiện tại có thể tạo ra các cách tiếp cận hiệu quả để liên lạc chuyển động toàn cục tới bộ giải mã và sử dụng các vectơ chuyển động toàn cục trong việc cải thiện hiệu quả nén.

Fig.1 là giản đồ đang minh họa các vectơ chuyển động của khung làm ví dụ 100 với chuyển động toàn cục và cục bộ; Khung 100 có thể chứa số các khối của các điểm ảnh được minh họa như là các hình vuông, và các vectơ chuyển động được kết hợp của chúng được minh họa như là các mũi tên. Các hình vuông (ví dụ, các khối của các điểm ảnh) với các mũi tên đang chỉ lên và tới bên trái chỉ thị các khối với chuyển động có thể được xem xét như là chuyển động toàn cục và các hình vuông với các mũi tên đang chỉ tới các hướng khác (được chỉ thị bởi 104) chỉ thị các khối với chuyển động cục bộ. Trong ví dụ được minh họa của Fig.1, nhiều khối trong số các khối có cùng chuyển động toàn cục. Việc tạo tín hiệu chuyển động toàn cục trong phần đầu, như tập hợp thông số hình ảnh (picture parameter set - PPS) hoặc tập hợp thông số chuỗi (sequence parameter set - SPS) và sử dụng chuyển động toàn cục được tạo tín hiệu có thể làm giảm thông tin vectơ chuyển động cần bởi các khối và có thể tạo thành việc dự đoán được cải tiến. Mặc dù cho các mục đích minh họa, nhưng các ví dụ được mô tả bên dưới đề cập tới việc xác định và/hoặc ứng dụng của các vectơ chuyển động toàn cục và cục bộ tại mức khối, các vectơ chuyển động toàn cục có thể được xác định và/hoặc được áp dụng cho vùng bất kỳ của khung và/hoặc ảnh, chứa các vùng tạo nên từ nhiều khối, các vùng được bao bởi dạng hình học bất kỳ như, mà không bị hạn chế, là các vùng được định nghĩa bởi việc mã hóa hình học và/hoặc dạng mũ mà trong đó một hoặc nhiều đường và/hoặc các đường cong đang bao quanh hình dạng, có thể được tạo góc và/hoặc được làm cong, và/hoặc toàn bộ của khung và/hoặc ảnh. Mặc dù việc tạo tín hiệu được mô tả ở đây như là đang được thực hiện tại mức khung và/hoặc trong phần đầu và/hoặc bộ thông số của khung, nhưng việc tạo tín hiệu có thể thay đổi hoặc còn được thực hiện tại mức dưới ảnh, trong đó phần dưới ảnh có thể chứa vùng bất kỳ của khung và/hoặc ảnh như được mô tả ở trên.

Như là một ví dụ, và vẫn đề cập tới Fig.1, chuyển động tịnh tiến đơn giản có thể được mô tả sử dụng vectơ chuyển động (MV) với hai thành phần MV_x , MV_y mô tả sự dịch chuyển của các khối và/hoặc các điểm ảnh trong khung hiện tại. Chuyển động phức tạp hơn như việc quay, phóng to/thu nhỏ, và lượn có thể được mô tả sử dụng các vectơ chuyển động afin, trong đó “vectơ chuyển động afin,” như được sử dụng trong bản mô tả này, là vectơ đang mô tả sự dịch chuyển đồng đều của bộ các điểm ảnh hoặc các điểm được biểu diễn trong ảnh video và/hoặc ảnh, như bộ các điểm ảnh đang minh họa vật thể đang di chuyển qua hình trong video mà không thay đổi hình dạng trong suốt chuyển động. Một số cách tiếp cận tới việc ghi mã và/hoặc giải mã video có thể sử dụng các mô hình afin 4 thông số hoặc 6 thông số cho việc bù chuyển động trong việc mã hóa ảnh liên thông.

Tiếp tục đề cập tới Fig.1 và như là ví dụ, chuyển động afin sáu thông số có thể được mô tả như:

$$x' = ax + by + c$$

$$y' = dx + ey + f$$

Chuyển động afin bốn thông số có thể được mô tả như sau:

$$x' = ax + by + c$$

$$y' = -bx + ay + f$$

trong đó (x,y) và (x',y') là các vị trí điểm ảnh trong các ảnh hiện tại và tham chiếu, theo cách tương ứng; a , b , c , d , e , và f là các thông số của mô hình chuyển động afin.

Vẫn đề cập tới Fig.1, các thông số được sử dụng để mô tả chuyển động afin có thể được tạo tín hiệu cho bộ giải mã để áp dụng việc bù chuyển động afin tại bộ giải mã. Theo một số cách tiếp cận, các thông số chuyển động có thể được tạo tín hiệu một cách rõ ràng hoặc bằng cách tạo tín hiệu các vectơ chuyển động điểm điều khiển (control point motion vectors - CPMV) tịnh tiến và sau đó dẫn ra các thông số chuyển động afin từ các vectơ chuyển động tịnh tiến. Hai vectơ chuyển động điểm điều khiển (CPMV) có thể được tận dụng để dẫn ra các thông số chuyển động afin cho mô hình chuyển động afin bốn thông số và ba vectơ chuyển động điểm điều khiển (CPMV) tịnh tiến có thể được tận dụng để thu

được các thông số cho mô hình chuyển động sáu thông số. Việc tạo tín hiệu các thông số chuyển động afin sử dụng các vectơ chuyển động điểm điều khiển có thể cho phép sử dụng các phương pháp mã hóa vectơ chuyển động hiệu quả để tạo tín hiệu cho các thông số chuyển động afin.

Theo một phương án thực hiện, và vẫn đề cập tới Fig.1, `sps_affine_enabled_flag` trong PPS và/hoặc SPS có thể chỉ rõ xem liệu việc bù chuyển động dựa trên mô hình afin có được thực hiện cho việc dự đoán liên thông hay không. Nếu `sps_affine_enabled_flag` là bằng với 0, thì cú pháp có thể bị ràng buộc sao cho không có việc bù chuyển động dựa trên mô hình afin nào được sử dụng trong chuỗi video muộn hơn mã (code later video sequence - CLVS), và `inter_affine_flag` và `cu_affine_type_flag` có thể không có mặt trong cú pháp đơn vị mã hóa của CLVS. Theo cách khác (`sps_affine_enabled_flag` là bằng với 1), việc bù chuyển động dựa trên mô hình afin có thể được sử dụng trong CLVS.

Tiếp tục đề cập tới Fig.1, `sps_affine_type_flag` trong PPS và/hoặc SPS có thể chỉ rõ xem liệu việc bù chuyển động dựa trên mô hình afin 6 thông số có thể được sử dụng cho việc dự đoán liên thông hay không. Nếu `sps_affine_type_flag` là bằng với 0, thì cú pháp có thể bị ràng buộc sao cho việc bù chuyển động dựa trên mô hình afin 6 thông số được sử dụng trong CLVS, và `cu_affine_type_flag` có thể không có mặt trong cú pháp đơn vị mã hóa trong CLVS. Theo cách khác (`sps_affine_type_flag` bằng với 1), thì việc bù chuyển động dựa trên mô hình afin 6 thông số có thể được sử dụng trong CLVS. Khi không có mặt, trị số của `sps_affine_type_flag` có thể được suy ra là bằng với 0.

Tiếp tục xem xét Fig.1, việc xử lý danh sách của các ứng viên dự đoán MV có thể là bước được thực hiện trong một số cách tiếp cận nén tận dụng việc bù chuyển động tại bộ giải mã. Một số cách tiếp cận có thể định nghĩa việc sử dụng của các ứng viên vectơ chuyển động theo không gian và thời gian. Chuyển động toàn cục được tạo tín hiệu trong phần đầu, như SPS hoặc PPS, có thể chỉ thị sự có mặt của chuyển động toàn cục trong video. Chuyển động toàn cục có thể được mong đợi là chung cho hầu hết các khối trong khung. Việc mã hóa vectơ chuyển động có thể được cải thiện và tốc độ bit được giảm bằng cách sử dụng chuyển động toàn cục như là ứng viên dự đoán. MV ứng viên được thêm vào danh

sách dự đoán MV có thể được chọn phụ thuộc vào mô hình chuyển động được sử dụng để biểu diễn chuyển động toàn cục và mô hình chuyển động được sử dụng trong việc mã hóa liên thông.

Vấn đề Fig.1, một số ứng dụng của chuyển động toàn cục được mô tả với một hoặc nhiều vectơ chuyển động điểm điều khiển (control point motion vector - CPMV) phụ thuộc vào mô hình chuyển động được sử dụng. Do đó, phụ thuộc vào mô hình chuyển động được sử dụng, từ một tới ba vectơ chuyển động điểm điều khiển có thể là sẵn có và có thể được sử dụng làm các ứng viên cho việc dự đoán. Theo một số áp dụng, tất cả các CPMV sẵn có có thể được thêm vào như là các ứng viên dự đoán vào danh sách. Trường hợp chung của việc thêm tất cả các CPMV sẵn có có thể làm tăng khả năng có thể có của việc tìm việc dự đoán vectơ chuyển động tốt và cải thiện hiệu quả nén. Bảng 1:

Mô hình chuyển động toàn cục được sử dụng (MM, các MV sẵn có)	PU MV đang được mã hóa	Ứng viên dự đoán GM được thêm vào
MM=0, MV0	MV0 (MM=0)	MV0
MM=0, MV0	MV1 (MM=1 hoặc 2)	MV0
MM=0, MV0	MV2 (MM=2)	MV0
MM=1, MV0, MV1	MV0 (MM=0)	MV0, MV1
MM=1, MV0, MV1	MV1 (MM=1 hoặc 2)	MV0, MV1
MM=1, MV0, MV1	MV2 (MM=2)	MV0, MV1
MM=2, MV0, MV1, MV2	MV0 (MM=0)	MV0, MV1, MV2
MM=2, MV0, MV1, MV2	MV1 (MM=1 hoặc 2)	MV0, MV1, MV2
MM=2, MV0, MV1, MV2	MV2 (MM=2)	MV0, MV1, MV2

Tiếp tục đề cập tới Fig.1, việc tạo danh sách của các ứng viên dự đoán vectơ chuyển động (motion vector - MV) có thể là bước trong việc thực hiện việc bù chuyển động tại bộ giải mã.

Vấn đề cập tới Fig.1, chuyển động toàn cục được tạo tín hiệu trong phần đầu, như SPS, có thể chỉ thị sự có mặt của chuyển động toàn cục trong video. Chuyển động toàn cục này có khả năng có mặt trong nhiều khối trong khung. Do đó, khối đã cho có thể có khả năng có chuyển động tương tự như chuyển động toàn cục. Việc mã hóa vectơ chuyển động có thể được cải thiện và tốc độ bit được giảm bằng cách sử dụng chuyển động toàn cục như

là ứng viên dự đoán. MV ứng viên có thể được thêm vào danh sách dự đoán MV và MV ứng viên có thể được chọn phụ thuộc vào mô hình chuyển động được sử dụng để biểu diễn chuyển động toàn cục và mô hình chuyển động được sử dụng trong việc mã hóa liên thông.

Vấn đề cập tới Fig.1, chuyển động toàn cục có thể được mô tả với một hoặc nhiều vectơ chuyển động điểm điều khiển (control point motion vector - CPMV) phụ thuộc vào mô hình chuyển động được sử dụng. Do đó, phụ thuộc vào mô hình chuyển động được sử dụng, từ một tới ba vectơ chuyển động điểm điều khiển có thể là sẵn có và có thể được sử dụng làm các ứng viên cho việc dự đoán. Danh sách của các ứng viên dự đoán MV có thể được làm giảm bởi việc thêm có chọn lọc một CPMV vào danh sách ứng viên dự đoán. Việc làm giảm kích cỡ danh sách có thể làm giảm độ phức tạp tính toán và cải thiện hiệu quả nén.

Theo một số áp dụng, và vấn đề cập tới Fig.1, CPMV được chọn làm ứng viên có thể theo việc ánh xạ được định nghĩa từ trước, như được minh họa trong bảng 3 dưới đây:

Mô hình chuyển động toàn cục được sử dụng (M, các MV sẵn có)	PU MV đang được mã hóa	Ứng viên dự đoán GM được thêm vào
M=0, MV0	MV0 (M=0, hoặc 1, hoặc 2)	MV0
M=0, MV0	MV1 (M=1 hoặc 2)	MV0
M=0, MV0	MV2 (M=2)	MV0
M=1, MV0, MV1	MV0 (M=0, hoặc 1, hoặc 2)	MV0
M=1, MV0, MV1	MV1 (M=1 hoặc 2)	MV1
M=1, MV0, MV1	MV2 (M=2)	MV0
M=2, MV0, MV1, MV2	MV0 (M=0, hoặc 1, hoặc 2)	MV0
M=2, MV0, MV1, MV2	MV1 (M=1 hoặc 2)	MV1
M=2, MV0, MV1, MV2	MV2 (M=2)	MV2

Tiếp tục cập tới Fig.1, việc sử dụng của ứng viên dự đoán chọn lọc từ chuyển động toàn cục có thể được tạo tín hiệu trong tập hợp thông số hình ảnh của tập hợp thông số chuỗi, nhờ đó làm giảm độ phức tạp ghi mã và giải mã.

Vấn đề cập tới Fig.1, do khối có khả năng có chuyển động tương tự như chuyển động toàn cục, nên việc thêm vectơ chuyển động toàn cục như là ứng viên thứ nhất trong danh sách dự đoán có thể làm giảm các bit cần thiết để tạo tín hiệu các ứng viên dự đoán được sử dụng và để ghi mã các khác biệt vectơ chuyển động.

Tiếp tục đề cập tới Fig.1, việc tạo danh sách của các ứng viên dự đoán vectơ chuyển động (motion vector - MV) có thể là bước trong việc thực hiện việc bù chuyển động tại bộ giải mã.

Vấn đề cập tới Fig.1, chuyển động toàn cục được tạo tín hiệu trong phần đầu, như SPS, có thể chỉ thị sự có mặt của chuyển động toàn cục trong video. Chuyển động toàn cục này có thể có khả năng có mặt trong nhiều khối trong khung. Do đó, khối đã cho có thể có khả năng có chuyển động tương tự như chuyển động toàn cục. Việc mã hóa vectơ chuyển động có thể được cải thiện và tốc độ bit được giảm bằng cách sử dụng chuyển động toàn cục như là ứng viên dự đoán.

Vấn đề cập tới Fig.1, ví dụ, MV ứng viên được thêm vào danh sách dự đoán MV có thể được chọn theo cách thích ứng phụ thuộc vào vectơ chuyển động điểm điều khiển (control point motion vector - CPMV) nào được chọn làm ứng viên dự đoán trong các khối lân cận (ví dụ, các đơn vị dự đoán (prediction unit - PU)).

Ví dụ, và vẫn đề cập tới Fig.1, nếu các PU lân cận sử dụng CPMV cụ thể làm MV dự đoán cho điểm điều khiển cụ thể, thì CPMV đó có thể được thêm vào danh sách ứng viên MV. Nếu các PU lân cận sử dụng nhiều hơn một CPMV làm MV dự đoán (ví dụ, các PU bên trái sử dụng CPMV0 và PU trên đỉnh sử dụng CPMV1), thì tất cả các CPMV được sử dụng làm các ứng viên dự đoán có thể được thêm vào trong danh sách. Nếu các PU lân cận không sử dụng CPMV, thì không có các CPMV nào có thể được thêm vào danh sách dự đoán. Theo một số áp dụng, thông tin chuyển động toàn cục có thể được thêm vào như là ứng viên thứ nhất trong danh sách dự đoán.

Vẫn tiếp tục tham khảo tới Fig.1, việc sử dụng của ứng viên dự đoán tương thích từ chuyển động toàn cục có thể được tạo tín hiệu trong phần đầu, như tập hợp thông số hình

ảnh (picture parameter set - PPS) hoặc tập hợp thông số chuỗi (sequence parameter set - SPS), nhờ đó làm giảm độ phức tạp ghi mã và giải mã.

Vấn đề cập tới Fig.1, do khối có khả năng có chuyển động tương tự như chuyển động toàn cục, nên việc thêm vectơ chuyển động toàn cục như là ứng viên thứ nhất trong danh sách dự đoán có thể làm giảm các bit cần thiết để tạo tín hiệu các ứng viên dự đoán được sử dụng và để ghi mã các khác biệt vectơ chuyển động. Việc tạo danh sách của các ứng viên dự đoán vectơ chuyển động (motion vector - MV) có thể là bước trong việc thực hiện việc bù chuyển động tại bộ giải mã.

Vấn đề cập tới Fig.1, chuyển động toàn cục được tạo tín hiệu trong phần đầu, như PPS hoặc SPS, có thể chỉ thị sự có mặt của chuyển động toàn cục trong video. Chuyển động toàn cục này có khả năng có mặt trong nhiều khối trong khung. Do đó, khối đã cho có thể có khả năng có chuyển động tương tự như chuyển động toàn cục. Việc mã hóa vectơ chuyển động có thể được cải thiện và tốc độ bit được giảm bằng cách sử dụng chuyển động toàn cục như là ứng viên dự đoán. Hơn nữa, việc mã hóa liên thông có thể được cải thiện và tốc độ bit được làm giảm bằng cách sử dụng mô hình chuyển động toàn cục và các điểm điều khiển làm các ứng viên hợp nhất.

Theo một số áp dụng, và còn đề cập tới Fig.1, khi chuyển động toàn cục được tạo tín hiệu trong phần đầu, như PPS hoặc SPS, thì mô hình chuyển động và các vectơ chuyển động điểm điều khiển (control point motion vector - CPMV) (cùng nhau được đề cập tới như là các vectơ chuyển động toàn cục) có thể được thêm vào danh sách ứng viên hợp nhất. Việc sử dụng của các ứng viên hợp nhất chuyển động toàn cục có thể cải thiện hiệu quả hoạt động nén.

Theo một số áp dụng, và vẫn đề cập tới Fig.1, chuyển động toàn cục mạnh mẽ có thể có khả năng để tạo kết quả trong phần lớn các khối với chuyển động toàn cục. Trong các trường hợp này, hiệu quả nén có thể được cải thiện bằng việc tạo tín hiệu việc sử dụng của `global_merge_mode` cho khối đã cho (ví dụ, đơn vị dự đoán (PU)) nằm trong khung. Khi `global_merge_mode` được tạo tín hiệu, các thông số chuyển động toàn cục có thể được sử dụng cho việc bù chuyển động trong khối (ví dụ, PU) và không có danh sách ứng viên hợp

nhất nào có thể được tạo ra. Phương pháp này cũng có thể làm giảm độ phức tạp tính toán tại bộ giải mã. Fig.2 minh họa ba mô hình chuyển động làm ví dụ có thể được tận dụng cho chuyển động toàn cục chứa các trị số chỉ số của chúng (0, 1, hoặc 2).

Vấn đề cập tới Fig.2, các PPS có thể được sử dụng để tạo tín hiệu các thông số có thể thay đổi giữa các ảnh của chuỗi. Các thông số vẫn giữ không đổi cho chuỗi của các ảnh có thể được tạo tín hiệu trong tập hợp thông số chuỗi để làm giảm kích cỡ của PPS và làm giảm tỉ lệ bit của video. Tập hợp thông số hình ảnh (picture parameter set - PPS) làm ví dụ được thể hiện trên bảng 2:

pic_parameter_set_rbsp() {	Bộ mô tả
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	u(4)
mixed_nalu_types_in_pic_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
pps_conformance_window_flag	u(1)
if(pps_conformance_window_flag) {	
pps_conf_win_left_offset	ue(v)
pps_conf_win_right_offset	ue(v)
pps_conf_win_top_offset	ue(v)
pps_conf_win_bottom_offset	ue(v)
}	
scaling_window_explicit_signalling_flag	u(1)
if(scaling_window_explicit_signalling_flag) {	
scaling_win_left_offset	ue(v)
scaling_win_right_offset	ue(v)
scaling_win_top_offset	ue(v)
scaling_win_bottom_offset	ue(v)
}	
output_flag_present_flag	u(1)
subpic_id_mapping_in_pps_flag	u(1)
if(subpic_id_mapping_in_pps_flag) {	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpic_minus1; i++)	
pps_subpic_id[i]	u(v)
}	
no_pic_partition_flag	u(1)

if(!no_pic_partition_flag) {	
pps_log2_ctu_size_minus5	u(2)
num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
if(NumTilesInPic > 1)	
rect_slice_flag	u(1)
if(rect_slice_flag)	
single_slice_per_subpic_flag	u(1)
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
if(num_slices_in_pic_minus1 > 0)	
tile_idx_delta_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
if(NumTileColumns > 1)	
slice_width_in_tiles_minus1[i]	ue(v)
if(NumTileRows > 1 && (tile_idx_delta_present_flag SliceTopLeftTileIdx[i] % NumTileColumns == 0))	
slice_height_in_tiles_minus1[i]	ue(v)
if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0 && RowHeight[SliceTopLeftTileIdx[i] / NumTileColumns] > 1) {	
num_exp_slices_in_tile[i]	ue(v)
for(j = 0; j < num_exp_slices_in_tile[i]; j++)	
exp_slice_height_in_ctus_minus1[i][j]	ue(v)
i += NumSlicesInTile[i] - 1	
}	
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1)	
tile_idx_delta[i]	se(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
}	
cabac_init_present_flag	u(1)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)

rpl1_idx_present_flag	u(1)
init_qp_minus26	se(v)
cu_qp_delta_enabled_flag	u(1)
pps_chroma_tool_offsets_present_flag	u(1)
if(pps_chroma_tool_offsets_present_flag) {	
pps_cb_qp_offset	se(v)
pps_cr_qp_offset	se(v)
pps_joint_cbr_qp_offset_present_flag	u(1)
if(pps_joint_cbr_qp_offset_present_flag)	
pps_joint_cbr_qp_offset_value	se(v)
pps_slice_chroma_qp_offsets_present_flag	u(1)
pps_cu_chroma_qp_offset_list_enabled_flag	u(1)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
chroma_qp_offset_list_len_minus1	ue(v)
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
if(pps_joint_cbr_qp_offset_present_flag)	
joint_cbr_qp_offset_list[i]	se(v)
}	
}	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
pps_cb_beta_offset_div2	se(v)
pps_cb_tc_offset_div2	se(v)
pps_cr_beta_offset_div2	se(v)
pps_cr_tc_offset_div2	se(v)
}	
}	
rpl_info_in_ph_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
dbf_info_in_ph_flag	u(1)
sao_info_in_ph_flag	u(1)

alf_info_in_ph_flag	u(1)
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && rpl_info_in_ph_flag)	
wp_info_in_ph_flag	u(1)
qp_delta_info_in_ph_flag	u(1)
pps_ref_wraparound_enabled_flag	u(1)
if(pps_ref_wraparound_enabled_flag)	
pps_ref_wraparound_offset	ue(v)
picture_header_extension_present_flag	u(1)
slice_header_extension_present_flag	u(1)
pps_extension_flag	u(1)
if(pps_extension_flag)	
while(more_rbsp_data())	
pps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

Các trường bổ sung có thể được thêm vào PPS để tạo tín hiệu chuyển động toàn cục. Trong trường hợp của chuyển động toàn cục, sự có mặt của các thông số chuyển động toàn cục trong trình tự của các ảnh có thể được tạo tín hiệu trong SPS và PPS có thể tham chiếu SPS bởi SPS ID. SPS theo một số cách tiếp cận tới việc giải mã có thể được biến đổi để thêm trường để tạo tín hiệu cho sự hiện diện của các thông số chuyển động toàn cục trong SPS. Ví dụ trường một bit có thể được thêm vào SPS. Nếu bit `global_motion_present` là 1, thì các thông số liên quan tới chuyển động toàn cục có thể được mong đợi trong PPS. Nếu bit `global_motion_present` là 0, thì không có các trường liên quan tới thông số chuyển động toàn cục nào có mặt trong PPS. Ví dụ, PPS của bảng 2 có thể được mở rộng để chứa trường `global_motion_present`, ví dụ, như được thể hiện trên bảng 3:

sequence_parameter_set_rbsp() {	Bộ mô tả
sps_sequence_parameter_set_id	ue(v)
·	
·	
·	
global_motion_present	u(1)

rbsp_trailing_bits()	
}	

Theo cách tương tự, PPS có thể chứa trường pps_global_motion_parameters cho khung, ví dụ như được thể hiện trên bảng 4:

pic_parameter_set_rbsp() {	Bộ mô tả
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
.	
.	
.	
pps_global_motion_parameters()	
rbsp_trailing_bits()	
}	

Chi tiết hơn, PPS có thể chứa các trường để đặc trưng cho các thông số chuyển động toàn cục sử dụng các vectơ chuyển động điểm điều khiển, ví dụ như được thể hiện trên bảng 5:

pps_global_motion_parameters () {	Bộ mô tả
motion_model_used	u(2)
mv0_x	se(v)
mv1_y	se(v)
if(motion_model_used == 1){	
mv1_x	se(v)
mv1_y	se(v)
}	
if(motion_model_used == 2){	
mv2_x	se(v)
mv2_y	se(v)
}	
}	

Như một ví dụ không làm hạn chế nữa, bảng 6 bên dưới có thể biểu diễn SPS làm ví dụ:

seq_parameter_set_rbsp() {	Bộ mô tả
sps_seq_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sublayers_minus1	u(3)

sps_reserved_zero_4bits	u(4)
sps_ptl_dpb_hrd_params_present_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
profile_tier_level(1, sps_max_sublayers_minus1)	
gdr_enabled_flag	u(1)
chroma_format_idc	u(2)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
res_change_in_clvs_allowed_flag	u(1)
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
sps_conformance_window_flag	u(1)
if(sps_conformance_window_flag) {	
sps_conf_win_left_offset	ue(v)
sps_conf_win_right_offset	ue(v)
sps_conf_win_top_offset	ue(v)
sps_conf_win_bottom_offset	ue(v)
}	
sps_log2_ctu_size_minus5	u(2)
subpic_info_present_flag	u(1)
if(subpic_info_present_flag) {	
sps_num_subpics_minus1	ue(v)
sps_independent_subpics_flag	u(1)
for(i = 0; sps_num_subpics_minus1 > 0 && i <=	
sps_num_subpics_minus1; i++) {	
if(i > 0 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_ctu_top_left_x[i]	u(v)
if(i > 0 && pic_height_max_in_luma_samples > CtbSizeY) {	
subpic_ctu_top_left_y[i]	u(v)
if(i < sps_num_subpics_minus1 &&	
pic_width_max_in_luma_samples > CtbSizeY)	
subpic_width_minus1[i]	u(v)
if(i < sps_num_subpics_minus1 &&	
pic_height_max_in_luma_samples > CtbSizeY)	
subpic_height_minus1[i]	u(v)
if(!sps_independent_subpics_flag) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
sps_subpic_id_len_minus1	ue(v)
subpic_id_mapping_explicitly_signalled_flag	u(1)

if(subpic_id_mapping_explicitly_signalled_flag) {	
subpic_id_mapping_in_sps_flag	u(1)
if(subpic_id_mapping_in_sps_flag)	
for(i = 0; i <= sps_num_subpics_minus1; i++)	
sps_subpic_id[i]	u(v)
}	
}	
bit_depth_minus8	ue(v)
sps_entropy_coding_sync_enabled_flag	u(1)
if(sps_entropy_coding_sync_enabled_flag)	
sps_wpp_entry_point_offsets_present_flag	u(1)
sps_weighted_pred_flag	u(1)
sps_weighted_bipred_flag	u(1)
log2_max_pic_order_cnt_lsb_minus4	u(4)
sps_poc_msb_flag	u(1)
if(sps_poc_msb_flag)	
poc_msb_len_minus1	ue(v)
num_extra_ph_bits_bytes	u(2)
extra_ph_bits_struct(num_extra_ph_bits_bytes)	
num_extra_sh_bits_bytes	u(2)
extra_sh_bits_struct(num_extra_sh_bits_bytes)	
if(sps_max_sublayers_minus1 > 0)	
sps_sublayer_dpb_params_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
dpb_parameters(sps_max_sublayers_minus1, sps_sublayer_dpb_params_flag)	
long_term_ref_pics_flag	u(1)
inter_layer_ref_pics_present_flag	u(1)
sps_idr_rpl_present_flag	u(1)
rpl1_same_as_rpl0_flag	u(1)
for(i = 0; i < rpl1_same_as_rpl0_flag ? 1 : 2; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
if(ChromaArrayType != 0)	
qtbtt_dual_tree_intra_flag	u(1)
log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	u(1)
sps_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)

if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
sps_log2_diff_min_qt_min_cb_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_inter_slice	ue(v)
if(sps_max_mtt_hierarchy_depth_inter_slice != 0) {	
sps_log2_diff_max_bt_min_qt_inter_slice	ue(v)
sps_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
if(qtbtt_dual_tree_intra_flag) {	
sps_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
sps_max_luma_transform_size_64_flag	u(1)
if(ChromaArrayType != 0) {	
sps_joint_cbr_enabled_flag	u(1)
same_qp_table_for_chroma	u(1)
numQpTables = same_qp_table_for_chroma ? 1 : (	
sps_joint_cbr_enabled_flag ? 3 : 2)	
for(i = 0; i < numQpTables; i++) {	
qp_table_start_minus26[i]	se(v)
num_points_in_qp_table_minus1[i]	ue(v)
for(j = 0; j <= num_points_in_qp_table_minus1[i]; j++) {	
delta_qp_in_val_minus1[i][j]	ue(v)
delta_qp_diff_val[i][j]	ue(v)
}	
}	
}	
sps_sao_enabled_flag	u(1)
sps_alf_enabled_flag	u(1)
if(sps_alf_enabled_flag && ChromaArrayType != 0)	
sps_ccalf_enabled_flag	u(1)
sps_transform_skip_enabled_flag	u(1)
if(sps_transform_skip_enabled_flag) {	
log2_transform_skip_max_size_minus2	ue(v)
sps_bdpcm_enabled_flag	u(1)

}	
sps_ref_wraparound_enabled_flag	u(1)
sps_temporal_mvp_enabled_flag	u(1)
if(sps_temporal_mvp_enabled_flag)	
sps_sbtmvp_enabled_flag	u(1)
sps_amvr_enabled_flag	u(1)
sps_bdof_enabled_flag	u(1)
if(sps_bdof_enabled_flag)	
sps_bdof_pic_present_flag	u(1)
sps_smvd_enabled_flag	u(1)
sps_dmvr_enabled_flag	u(1)
if(sps_dmvr_enabled_flag)	
sps_dmvr_pic_present_flag	u(1)
sps_mmvd_enabled_flag	u(1)
sps_isp_enabled_flag	u(1)
sps_mrl_enabled_flag	u(1)
sps_mip_enabled_flag	u(1)
if(ChromaArrayType != 0)	
sps_cclm_enabled_flag	u(1)
if(chroma_format_idc == 1) {	
sps_chroma_horizontal_collocated_flag	u(1)
sps_chroma_vertical_collocated_flag	u(1)
}	
sps_mts_enabled_flag	u(1)
if(sps_mts_enabled_flag) {	
sps_explicit_mts_intra_enabled_flag	u(1)
sps_explicit_mts_inter_enabled_flag	u(1)
}	
six_minus_max_num_merge_cand	ue(v)
sps_sbt_enabled_flag	u(1)
sps_affine_enabled_flag	u(1)
if(sps_affine_enabled_flag) {	
five_minus_max_num_subblock_merge_cand	ue(v)
sps_affine_type_flag	u(1)
if(sps_amvr_enabled_flag)	
sps_affine_amvr_enabled_flag	u(1)
sps_affine_prof_enabled_flag	u(1)
if(sps_affine_prof_enabled_flag)	
sps_prof_pic_present_flag	u(1)
}	
sps_palette_enabled_flag	u(1)

if(ChromaArrayType == 3 && !sps_max_luma_transform_size_64_flag)	
sps_act_enabled_flag	u(1)
if(sps_transform_skip_enabled_flag sps_palette_enabled_flag)	
min_qp_prime_ts_minus4	ue(v)
sps_bcw_enabled_flag	u(1)
sps_ibc_enabled_flag	u(1)
if(sps_ibc_enabled_flag)	
six_minus_max_num_ibc_merge_cand	ue(v)
sps_ciip_enabled_flag	u(1)
if(sps_mmvd_enabled_flag)	
sps_fpel_mmvd_enabled_flag	u(1)
if(MaxNumMergeCand >= 2) {	
sps_gpm_enabled_flag	u(1)
if(sps_gpm_enabled_flag && MaxNumMergeCand >= 3)	
max_num_merge_cand_minus_max_num_gpm_cand	ue(v)
}	
sps_lmcs_enabled_flag	u(1)
sps_lfnst_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
if(sps_ladf_enabled_flag) {	
sps_num_ladf_intervals_minus2	u(2)
sps_ladf_lowest_interval_qp_offset	se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
sps_ladf_qp_offset[i]	se(v)
sps_ladf_delta_threshold_minus1[i]	ue(v)
}	
}	
log2_parallel_merge_level_minus2	ue(v)
sps_explicit_scaling_list_enabled_flag	u(1)
sps_dep_quant_enabled_flag	u(1)
if(!sps_dep_quant_enabled_flag)	
sps_sign_data_hiding_enabled_flag	u(1)
sps_virtual_boundaries_enabled_flag	u(1)
if(sps_virtual_boundaries_enabled_flag) {	
sps_virtual_boundaries_present_flag	u(1)
if(sps_virtual_boundaries_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)

for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(sps_ptl_dpb_hrd_params_present_flag) {	
sps_general_hrd_params_present_flag	u(1)
if(sps_general_hrd_params_present_flag) {	
general_hrd_parameters()	
if(sps_max_sublayers_minus1 > 0)	
sps_sublayer_cpb_params_present_flag	u(1)
firstSubLayer = sps_sublayer_cpb_params_present_flag ? 0 :	
sps_max_sublayers_minus1	
ols_hrd_parameters(firstSubLayer, sps_max_sublayers_minus1)	
}	
}	
field_seq_flag	u(1)
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters() /* Specified in ITU-T H.SEI ISO/IEC 23002-7	
*/	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

Bảng SPS như trên có thể được mở rộng và được mô tả ở trên để tích hợp bộ chỉ thị có mặt chuyển động toàn cục như được thể hiện trên bảng 7:

sequence_parameter_set_rbsp() {	Bộ mô tả
sps_sequence_parameter_set_id	ue(v)
.	
.	
.	
global_motion_present	u(1)
rbsp_trailing_bits()	
}	

Các trường bổ sung có thể được tích hợp trong SPS để phản ánh các bộ chỉ thị khác như được mô tả trong sáng chế này.

Theo một phương án thực hiện, và vẫn đề cập tới Fig.2, `sps_affine_enabled_flag` trong PPS và/hoặc SPS có thể chỉ rõ xem liệu việc bù chuyển động dựa trên mô hình afin có được thực hiện cho việc dự đoán liên thông hay không. Nếu `sps_affine_enabled_flag` là bằng với 0, thì cú pháp có thể bị ràng buộc sao cho không có việc bù chuyển động dựa trên mô hình afin nào được sử dụng trong chuỗi video muộn hơn mã (code later video sequence - CLVS), và `inter_affine_flag` và `cu_affine_type_flag` có thể không có mặt trong cú pháp đơn vị mã hóa của CLVS. Theo cách khác (`sps_affine_enabled_flag` là bằng với 1), việc bù chuyển động dựa trên mô hình afin có thể được sử dụng trong CLVS.

Tiếp tục đề cập tới Fig.2, `sps_affine_type_flag` trong PPS và/hoặc SPS có thể chỉ rõ xem liệu việc bù chuyển động dựa trên mô hình afin 6 thông số có thể được sử dụng cho việc dự đoán liên thông hay không. Nếu `sps_affine_type_flag` là bằng với 0, thì cú pháp có thể bị ràng buộc sao cho việc bù chuyển động dựa trên mô hình afin 6 thông số được sử dụng trong CLVS, và `cu_affine_type_flag` có thể không có mặt trong cú pháp đơn vị mã hóa trong CLVS. Theo cách khác (`sps_affine_type_flag` bằng với 1), thì việc bù chuyển động dựa trên mô hình afin 6 thông số có thể được sử dụng trong CLVS. Khi không có mặt, trị số của `sps_affine_type_flag` có thể được suy ra là bằng với 0.

Vẫn đề cập tới Fig.2, các CPMV tịnh tiến có thể được tạo tín hiệu trong PPS. Các điểm điều khiển có thể được định nghĩa từ trước. Ví dụ, điểm điều khiển MV 0 có thể là tương đối so với góc trên bên trái của ảnh, MV 1 có thể là tương đối so với góc trên bên phải, và MV 3 có thể là tương đối so với góc dưới bên trái của ảnh. Bảng 5 minh họa cách tiếp cận làm ví dụ cho việc tạo tín hiệu dữ liệu CPMV đang phụ thuộc vào mô hình chuyển động được sử dụng.

Theo phương án thực hiện làm ví dụ, và vẫn đề cập tới Fig.2, mảng `amvr_precision_idx`, vốn có thể được tạo tín hiệu theo đơn vị mã hóa, cây mã hóa, hoặc dạng tương tự, có thể chỉ rõ độ phân giải `AmvrShift` của khác biệt vectơ chuyển động, vốn có thể được định nghĩa như là ví dụ không hạn chế như được thể hiện trên bảng 8 như được thể hiện bên dưới. Các chỉ số mảng `x0`, `y0` có thể cụ thể hóa vị trí (`x0`, `y0`) của mẫu độ chói trên bên trái của khối mã hóa được xem xét so với mẫu độ chói trên bên trái của ảnh;

khi $\text{amvr_precision_idx}[x0][y0]$ không có mặt, nó có thể được suy ra là bằng với 0. Trong đó $\text{inter_affine_flag}[x0][y0]$ là bằng với 0, các biến $\text{MvdL0}[x0][y0][0]$, $\text{MvdL0}[x0][y0][1]$, $\text{MvdL1}[x0][y0][0]$, $\text{MvdL1}[x0][y0][1]$ đang biểu diễn các trị số khác biệt vectơ chuyển động tương ứng với gói được xem xét, có thể được biến đổi bằng cách dịch chuyển các trị số này bởi AmvrShift , ví dụ sử dụng $\text{MvdL0}[x0][y0][0] = \text{MvdL0}[x0][y0][0] \ll \text{AmvrShift}$; $\text{MvdL0}[x0][y0][1] = \text{MvdL0}[x0][y0][1] \ll \text{AmvrShift}$; $\text{MvdL1}[x0][y0][0] = \text{MvdL1}[x0][y0][0] \ll \text{AmvrShift}$; và $\text{MvdL1}[x0][y0][1] = \text{MvdL1}[x0][y0][1] \ll \text{AmvrShift}$. Trong đó $\text{inter_affine_flag}[x0][y0]$ là bằng với 1, các biến $\text{MvdCpL0}[x0][y0][0][0]$, $\text{MvdCpL0}[x0][y0][0][1]$, $\text{MvdCpL0}[x0][y0][1][0]$, $\text{MvdCpL0}[x0][y0][1][1]$, $\text{MvdCpL0}[x0][y0][2][0]$ và $\text{MvdCpL0}[x0][y0][2][1]$ có thể được biến đổi thông qua dịch chuyển, ví dụ như sau: $\text{MvdCpL0}[x0][y0][0][0] = \text{MvdCpL0}[x0][y0][0][0] \ll \text{AmvrShift}$; $\text{MvdCpL1}[x0][y0][0][1] = \text{MvdCpL1}[x0][y0][0][1] \ll \text{AmvrShift}$; $\text{MvdCpL0}[x0][y0][1][0] = \text{MvdCpL0}[x0][y0][1][0] \ll \text{AmvrShift}$; $\text{MvdCpL1}[x0][y0][1][1] = \text{MvdCpL1}[x0][y0][1][1] \ll \text{AmvrShift}$; $\text{MvdCpL0}[x0][y0][2][0] = \text{MvdCpL0}[x0][y0][2][0] \ll \text{AmvrShift}$; và $\text{MvdCpL1}[x0][y0][2][1] = \text{MvdCpL1}[x0][y0][2][1] \ll \text{AmvrShift}$

amvr_flag	amvr_precision_idx	AmvrShift		
		$\text{inter_affine_flag} == 1$	$\text{CuPredMode}[\text{chType}][x0][y0] == \text{MODE_IBC}$	$\text{inter_affine_flag} == 0$ && $\text{CuPredMode}[\text{chType}][x0][y0] != \text{MODE_IBC}$
0	-	2 (1/4 mẫu độ chói)	-	2 (1/4 mẫu độ chói)
1	0	0 (1/16 mẫu độ chói)	4 (1 mẫu độ chói)	3 (1/2 mẫu độ chói)
1	1	4 (1 mẫu độ chói)	6 (4 mẫu độ chói)	4 (1 mẫu độ chói)
1	2	-	-	6 (4 mẫu độ chói)

Fig.3 là giản đồ tiến trình công nghệ đang minh họa quy trình làm ví dụ 300 của việc xây dựng danh sách ứng viên hợp nhất chứa việc thêm vectơ chuyển động toàn cục vào danh sách ứng viên vectơ chuyển động.

Tại bước 305, luồng bit được nhận bởi bộ giải mã. Khối hiện tại có thể được chứa trong luồng bit mà bộ giải mã nhận. Luồng bit có thể chứa, ví dụ, dữ liệu được tìm thấy trong luồng của các bit là đầu vào tới bộ giải mã khi sử dụng việc nén dữ liệu. Luồng bit có thể chứa thông tin cần thiết để giải mã video. Bước nhận có thể chứa việc tách và/hoặc phân tích cú pháp khối và việc tạo tín hiệu được kết hợp cho thông tin từ luồng bit. Theo một số áp dụng, khối hiện tại có thể chứa đơn vị cây mã hóa (coding tree unit - CTU), đơn vị mã hóa (coding unit - CU), và/hoặc đơn vị dự đoán (prediction unit - PU). Tại bước 310, chế độ hợp nhất được cho phép thực hiện được xác định cho khối hiện tại và sử dụng luồng bit. Tại bước 315, danh sách ứng viên hợp nhất được xây dựng; bước xây dựng chứa việc thêm vectơ chuyển động toàn cục vào danh sách ứng viên vectơ chuyển động. Tại bước 340, dữ liệu điểm ảnh của khối hiện tại có thể được xây dựng lại sử dụng danh sách ứng viên vectơ chuyển động.

Fig.4 là giản đồ khối hệ thống đang minh họa bộ giải mã làm ví dụ 400 có khả năng giải mã luồng bit chứa việc xây dựng danh sách ứng viên hợp nhất chứa việc thêm vectơ chuyển động toàn cục vào danh sách ứng viên vectơ chuyển động. Bộ giải mã 400 có thể chứa bộ xử lý bộ giải mã entropy 404, bộ xử lý lượng tử hóa ngược và biến đổi ngược 408, bộ lọc khử khối 412, bộ đệm khung 416, bộ xử lý bù chuyển động 420 và/hoặc bộ xử lý dự đoán nội 424.

Trong khi vận hành, và vẫn đề cập tới Fig.4, luồng bit 428 có thể được nhận bởi bộ giải mã 400 và được nhập vào bộ xử lý bộ giải mã entropy 404, vốn có thể giải mã entropy các phần của luồng bit thành các hệ số được lượng tử hóa. Các hệ số được lượng tử hóa có thể được tạo ra cho bộ xử lý lượng tử hóa ngược và biến đổi ngược 408, vốn có thể thực hiện việc lượng tử hóa ngược và biến đổi ngược để tạo ra tín hiệu còn dư, vốn có thể được thêm vào đầu ra của bộ xử lý bù chuyển động 420 hoặc bộ xử lý dự đoán nội 424 theo chế độ xử lý. Đầu ra của bộ xử lý bù chuyển động 420 và bộ xử lý dự đoán nội 424 có thể chứa

việc dự đoán khối dựa trên khối đã được dự đoán trước đó. Tổng của phần dự đoán và phần còn dư có thể được xử lý bởi bộ lọc khử khối 412 và được lưu trong bộ đệm khung 416.

Fig.5 là giản đồ tiến trình công nghệ đang minh họa quy trình làm ví dụ 500 của việc ghi mã video với việc xây dựng danh sách ứng viên hợp nhất chứa việc thêm vectơ chuyển động toàn cục vào danh sách ứng viên vectơ chuyển động theo một số khía cạnh của đối tượng hiện tại có thể làm giảm độ phức tạp ghi mã trong khi vẫn tăng hiệu quả nén. Tại bước 505, khung video có thể trải qua việc phân mảnh khối ban đầu, ví dụ, sử dụng mẫu phân mảnh khối macro được tạo cấu trúc cây có thể chứa việc phân mảnh khung ảnh thành các CTU và các CU. Tại bước 510, chuyển động toàn cục có thể được xác định. Tại bước 515, khối có thể được ghi mã và được chứa trong luồng bit. Bước ghi mã có thể chứa việc xây dựng danh sách ứng viên hợp nhất chứa việc thêm vectơ chuyển động toàn cục vào danh sách ứng viên vectơ chuyển động. Việc ghi mã có thể chứa việc tận dụng các chế độ dự đoán liên thông và dự đoán nội, ví dụ.

Fig.6 là giản đồ khối hệ thống đang minh họa bộ ghi mã video làm ví dụ 600 có khả năng xây dựng danh sách ứng viên hợp nhất chứa việc thêm vectơ chuyển động toàn cục vào danh sách ứng viên vectơ chuyển động. Bộ ghi mã video làm ví dụ 600 có thể nhận video đầu vào 604, vốn có thể được phân mảnh từ đầu hoặc chia theo mẫu xử lý, như mẫu phân chia khối macro được tạo cấu trúc cây (ví dụ, cây tứ phân cộng cây nhị phân). Ví dụ của mẫu phân mảnh khối macro được tạo cấu trúc cây có thể chứa việc phân mảnh khung ảnh thành các phần tử khối lớn được gọi là các đơn vị cây mã hóa (coding tree unit - CTU). Theo một số áp dụng, mỗi CTU có thể còn được phân mảnh một hoặc nhiều lần thành nhiều khối con được gọi là các đơn vị mã hóa (Coding Unit - CU). Kết quả cuối cùng của việc phân chia này có thể chứa nhóm các khối con có thể được gọi là các đơn vị dự đoán (predictive unit - PU). Các đơn vị biến đổi (Transform unit - TU) cũng có thể được tận dụng.

Vấn đề cập tới Fig.6, bộ ghi mã video làm ví dụ 600 có thể chứa bộ xử lý dự đoán nội 612, bộ xử lý ước lượng/bù chuyển động 612, vốn cũng có thể được đề cập tới như là bộ xử lý dự đoán liên thông) có khả năng xây dựng danh sách ứng viên vectơ chuyển động

chứa việc thêm ứng viên véctor chuyển động toàn cục vào danh sách ứng viên véctor chuyển động, bộ xử lý biến đổi/lượng tử hóa 616, bộ xử lý lượng tử hóa ngược/biến đổi ngược 620, bộ lọc trong vòng 624, bộ đệm ảnh được giải mã 628, và/hoặc bộ xử lý mã hóa entropy 632. Các thông số luồng bit có thể được nhập vào bộ xử lý mã hóa entropy 632 để chứa trong luồng bit đầu ra 636.

Trong việc vận hành, và tiếp tục đề cập tới Fig.6, cho từng khối của khung của video đầu vào 604, xem liệu xử lý khối thông qua việc dự đoán ảnh nội hay sử dụng việc ước lượng/bù chuyển động có thể được xác định. Khối có thể được cung cấp cho bộ xử lý dự đoán nội 608 hoặc bộ xử lý ước lượng/bù chuyển động 612. Nếu khối cần được xử lý qua việc dự đoán nội, thì bộ xử lý dự đoán nội 608 có thể thực hiện việc xử lý để đưa ra bộ dự đoán. Nếu khối cần được xử lý thông qua việc ước lượng/việc bù chuyển động, thì bộ xử lý ước lượng/bù chuyển động 612 có thể thực hiện việc xử lý chứa việc xây dựng danh sách ứng viên véctor chuyển động chứa việc thêm ứng viên véctor chuyển động toàn cục vào danh sách ứng viên véctor chuyển động, nếu có thể áp dụng.

Vấn đề cập tới Fig.6, phần còn dư có thể được thực hiện bằng việc trừ bộ dự đoán từ video đầu vào. Phần còn dư có thể được nhận bởi bộ xử lý biến đổi/lượng tử hóa 616, vốn có thể thực hiện việc xử lý biến đổi (ví dụ, biến đổi cosin rời rạc (discrete cosine transform - DCT)) để tạo ra các hệ số, vốn có thể được lượng tử hóa. Các hệ số được lượng tử hóa và thông tin tạo tín hiệu được kết hợp bất kỳ có thể được tạo ra cho bộ xử lý mã hóa entropy 632 cho việc ghi mã entropy và việc chứa trong luồng bit đầu ra 636. Bộ xử lý ghi mã entropy 632 có thể trợ giúp việc ghi mã của thông tin tạo tín hiệu liên quan tới việc ghi mã khối hiện tại. Thêm vào đó, các hệ số được lượng tử hóa có thể được tạo ra cho bộ xử lý lượng tử hóa ngược/biến đổi ngược 620, vốn có thể tái tạo các điểm ảnh, vốn có thể được kết hợp với bộ dự đoán và được xử lý bởi bộ lọc trong vòng 624, việc đưa ra của nó có thể được lưu trong bộ đệm ảnh được giải mã 628 để sử dụng bởi bộ xử lý ước lượng/bù chuyển động 612 có khả năng xây dựng danh sách ứng viên véctor chuyển động chứa việc thêm ứng viên véctor chuyển động toàn cục vào danh sách ứng viên véctor chuyển động.

Vẫn tiếp tục tham khảo tới Fig.6, mặc dù một số ít biến thể đã được mô tả chi tiết ở trên, nhưng các điều chỉnh hoặc bổ sung khác là cũng khả thi. Ví dụ, theo một số áp dụng, các khối hiện tại có thể chứa các khối đối xứng bất kỳ (8×8 , 16×16 , 32×32 , 64×64 , 128×128 , và dạng tương tự) cũng như khối phi đối xứng (8×4 , 16×8 , và dạng tương tự).

Theo một số áp dụng, và vẫn đề cập tới Fig.6, cây quyết định tứ phân cộng nhị phân (QTBT) có thể được áp dụng. Trong QTBT, tại mức đơn vị cây mã hóa, các thông số phân mảnh của QTBT có thể được dẫn ra theo cách động để thích ứng với các đặc tính cục bộ mà không truyền phân trên đầu bất kỳ nào. Sau đó, tại mức đơn vị mã hóa, cấu trúc cây quyết định bộ phân loại chung (joint-classifier decision tree structure) có thể loại bỏ việc lặp lại không cần thiết và giám sát rủi ro của việc dự đoán lỗi. Theo một số áp dụng, chế độ cập nhật khối khung LTR có thể là sẵn có như là lựa chọn có sẵn tại mọi nút lá của QTBT.

Theo một số áp dụng, và vẫn tham khảo tới Fig.6, các thành phần cú pháp bổ sung có thể được tạo tín hiệu tại các mức phân cấp khác nhau của luồng bit. Ví dụ, cò có thể được cho phép thực hiện cho toàn bộ chuỗi bằng cách chứa cò cho phép được ghi mã trong tập hợp thông số chuỗi (Sequence Parameter Set - SPS). Hơn nữa, cò CTU có thể được mã hóa tại mức đơn vị cây mã hóa (CTU).

Cần chú ý rằng một hoặc nhiều khía cạnh và phương án thực hiện bất kỳ trong số các khía cạnh và các phương án thực hiện được mô tả ở đây có thể được áp dụng sử dụng hệ mạch điện tử kỹ thuật số, hệ mạch tích hợp, các mạch tích hợp ứng dụng cụ thể (application specific integrated circuit - ASIC) được thiết kế riêng, các mảng cổng lập trình được bằng trường (field programmable gate array - FPGA) phần cứng máy tính, phần sụn, phần mềm, và/hoặc các tổ hợp của chúng, như được hiện thực hóa và/hoặc được áp dụng trong một hoặc nhiều máy (ví dụ, một hoặc nhiều thiết bị tính toán được tận dụng như là thiết bị tính toán của người sử dụng cho tài liệu điện tử, một hoặc nhiều thiết bị chủ, như máy chủ tài liệu, v.v.) được lập trình theo các chỉ dẫn của bản mô tả này, như sẽ là rõ ràng đối với người có hiểu biết trung bình trong lĩnh vực kỹ thuật máy tính. Các khía cạnh hoặc dấu hiệu khác nhau này có thể chứa việc áp dụng trong một hoặc nhiều chương trình máy tính và/hoặc

phần mềm có thể thực thi được và/hoặc có thể diễn giải được trên hệ thống lập trình được chứa ít nhất một bộ xử lý có thể lập trình được, vốn có thể cho mục đích cụ thể hoặc mục đích chung, được ghép nối để nhận dữ liệu và các lệnh từ, và để truyền dữ liệu và các lệnh tới, hệ thống lưu trữ, ít nhất một thiết bị nhập vào, và ít nhất một thiết bị đưa ra. Việc mã hóa phần mềm thích hợp có thể được chuẩn bị sẵn sàng bởi người lập trình có kinh nghiệm dựa trên các chỉ dẫn này của sáng chế, như sẽ là rõ ràng đối với những người có hiểu biết trung bình trong lĩnh vực kỹ thuật phần mềm. Các khía cạnh và các việc ứng dụng được thảo luận ở trên áp dụng phần mềm và/hoặc các môđun phần mềm có thể còn bao gồm phần mềm cũng có thể chứa phần cứng thích hợp để trợ giúp việc áp dụng của các lệnh có thể thực thi được bởi máy của phần mềm và/hoặc môđun phần mềm.

Phần mềm này có thể là sản phẩm chương trình máy tính áp dụng môi trường lưu trữ đọc được bởi máy. Môi trường lưu trữ đọc được bởi máy có thể là môi trường là có khả năng lưu trữ và/hoặc ghi mã chuỗi các lệnh cho việc thực thi bởi máy (ví dụ, thiết bị tính toán) và làm cho máy thực hiện phương pháp bất kỳ trong số các phương pháp và/hoặc các phương án thực hiện được mô tả ở đây. Các ví dụ của môi trường lưu trữ đọc được bởi máy chứa, nhưng không giới hạn ở, đĩa từ, đĩa quang học (ví dụ, CD, CD-R, DVD, DVD-R, v.v.), đĩa từ-quang, thiết bị bộ nhớ chỉ đọc “read-only memory - ROM”, thiết bị bộ nhớ truy cập ngẫu nhiên “random access memory - RAM”, thẻ từ, thẻ quang học, thiết bị bộ nhớ trạng thái rắn, EPROM, EEPROM, các thiết bị logic có thể lập trình được (Programmable Logic Device - PLD), và/hoặc các tổ hợp bất kỳ của chúng. Vật ghi đọc được bởi máy, như được sử dụng ở đây, nhằm chứa môi trường đơn cũng như tập hợp của môi trường tách biệt về mặt vật lý, như, ví dụ, tập hợp của các đĩa nhỏ gọn (compact) hoặc một hoặc nhiều ổ đĩa cứng trong kết hợp với bộ nhớ máy tính. Như được sử dụng ở đây, môi trường lưu trữ đọc được bởi máy không chứa dạng chuyển tiếp của việc truyền tín hiệu.

Phần mềm này cũng có thể chứa thông tin (ví dụ, dữ liệu) được mang như là tín hiệu dữ liệu trên bộ mang dữ liệu, như sóng mang. Ví dụ, thông tin có thể thực thi được bởi máy có thể được chứa như là tín hiệu đang mang dữ liệu được áp dụng trong bộ mang dữ liệu mà trong đó tín hiệu ghi mã chuỗi lệnh, cho việc thực thi bởi máy (ví dụ, thiết bị tính toán)

và thông tin có liên quan bất kỳ (ví dụ, các cấu trúc dữ liệu và dữ liệu) làm cho máy thực hiện thành phần bất kỳ trong số các phương pháp và/hoặc các phương án thực hiện được mô tả ở đây.

Các ví dụ của thiết bị tính toán chứa, nhưng không giới hạn ở, thiết bị đọc sách điện tử, trạm làm việc máy tính, máy tính đầu cuối, máy tính chủ, thiết bị cầm tay (ví dụ, máy tính bảng, điện thoại thông minh, v.v.), ứng dụng web, bộ định tuyến mạng, chuyển mạch mạng, cầu mạng, máy bất kỳ có khả năng thực thi chuỗi các lệnh chỉ rõ hành động cần được thực hiện bởi máy đó, và các tổ hợp bất kỳ của chúng. Theo một ví dụ, thiết bị tính toán có thể chứa và/hoặc được chứa trong ki-ốt.

Fig.7 thể hiện biểu diễn dưới dạng giản đồ của một phương án thực hiện của thiết bị tính toán trong dạng làm ví dụ của hệ thống máy tính 700 mà nằm trong đó là bộ các lệnh để làm cho hệ thống điều khiển thực hiện thành phần bất kỳ trong số một hoặc nhiều thành phần trong số các khía cạnh và/hoặc các phương pháp của sáng chế này, có thể được thực thi. Đã dự tính rằng nhiều thiết bị tính toán có thể được tận dụng để áp dụng tập hợp các lệnh được đặt cấu hình cụ thể để làm cho một hoặc nhiều thiết bị trong số các thiết bị thực hiện một hoặc nhiều thành phần bất kỳ trong số các khía cạnh và/hoặc các phương pháp của sáng chế này. Hệ thống máy tính 700 chứa bộ xử lý 704 và bộ nhớ 708 liên lạc với nhau, và với các thành phần khác, thông qua bus 712. Bus 712 có thể chứa loại bất kỳ trong số nhiều loại cấu trúc bus chứa, nhưng không giới hạn ở, bus bộ nhớ, bộ điều khiển bộ nhớ, bus ngoại vi, bus cục bộ, và các tổ hợp bất kỳ của chúng, sử dụng kiến trúc bất kỳ trong số nhiều kiến trúc khác nhau.

Bộ nhớ 708 có thể chứa các thành phần khác nhau (ví dụ, môi trường đọc được bởi máy) chứa, nhưng không giới hạn ở, thành phần bộ nhớ truy cập ngẫu nhiên, thành phần chỉ đọc, và các tổ hợp bất kỳ của chúng. Theo một ví dụ, hệ thống đầu vào/đầu ra cơ bản 716 (basic input/output system - BIOS), chứa các thủ tục cơ bản để giúp truyền thông tin giữa các phần tử nằm trong hệ thống máy tính 700, như trong suốt quá trình khởi động, có thể được lưu trong bộ nhớ 708. Bộ nhớ 708 cũng có thể chứa (ví dụ, được lưu trên một hoặc nhiều môi trường đọc được bởi máy) các lệnh (ví dụ, phần mềm) 720 đang áp dụng

một hoặc nhiều thành phần bất kỳ trong số các khía cạnh và/hoặc các phương pháp của sáng chế này. Theo ví dụ khác, bộ nhớ 708 có thể còn chứa số bất kỳ các môđun chương trình chứa, nhưng không giới hạn ở, hệ điều hành, một hoặc nhiều chương trình ứng dụng, các môđun chương trình khác, dữ liệu chương trình, và các tổ hợp bất kỳ của chúng.

Hệ thống máy tính 700 cũng có thể chứa thiết bị lưu trữ 724. Các ví dụ của thiết bị lưu trữ (ví dụ, thiết bị lưu trữ 724) chứa, nhưng không giới hạn ở, ổ đĩa cứng, ổ đĩa từ, ổ đĩa quang học trong kết hợp với môi trường quang học, thiết bị lưu trữ trạng thái rắn, và các tổ hợp bất kỳ của chúng. Thiết bị lưu trữ 724 có thể được kết nối tới bus 712 bởi giao diện thích hợp (không được thể hiện). Các giao diện làm ví dụ chứa, nhưng không giới hạn ở, SCSI, việc gắn công nghệ tiên tiến (advanced technology attachment - ATA), ATA nối tiếp, bus nối tiếp đa năng (universal serial bus - USB), IEEE 1394 (FIREWIRE), và các tổ hợp bất kỳ của chúng. Theo một ví dụ, thiết bị lưu trữ 724 (hoặc một hoặc nhiều thành phần của chúng) có thể được giao tiếp theo các loại bỏ được với hệ thống máy tính 700 (ví dụ, thông qua bộ phận kết nối cổng bên ngoài (không được thể hiện)). Cụ thể là, thiết bị lưu trữ 724 và vật ghi đọc được bởi máy 728 được kết hợp có thể là bộ phận lưu trữ bất khả biến hoặc của các lệnh đọc được bởi máy, các cấu trúc dữ liệu, các môđun chương trình, và/hoặc dữ liệu khác cho hệ thống máy tính 700. Theo một ví dụ, phần mềm 720 có thể ở trên, hoàn toàn hoặc một phần, nằm trong vật ghi đọc được bởi máy 728. Theo ví dụ khác, phần mềm 720 có thể ở trên, hoàn toàn hoặc một phần, nằm trong bộ xử lý 704.

Hệ thống máy tính 700 cũng có thể chứa thiết bị nhập vào 732. Theo một ví dụ, người sử dụng của hệ thống máy tính 700 có thể nhập vào các lệnh và/hoặc thông tin khác vào trong hệ thống máy tính 700 thông qua thiết bị nhập vào 732. Các ví dụ của thiết bị nhập vào 732 chứa, nhưng không giới hạn ở, thiết bị nhập vào chữ-số (ví dụ, bàn phím), thiết bị trỏ, cần điều khiển, bảng trò chơi, thiết bị nhập vào audio (ví dụ, micro phon, hệ thống đáp ứng giọng nói, v.v.), thiết bị điều khiển con trỏ (ví dụ, chuột), bảng chạm, máy quét quang học, thiết bị chụp video (ví dụ, camera tĩnh, camera video), màn chạm, và các tổ hợp bất kỳ của chúng. Thiết bị nhập vào 732 có thể được giao tiếp tới bus 712 thông qua giao diện bất kỳ trong số các giao diện (không được thể hiện) chứa, nhưng không giới hạn ở, giao

diện nối tiếp, giao diện song song, cổng trò chơi, giao diện USB, giao diện FIREWIRE, giao diện trực tiếp tới bus 712, và các tổ hợp bất kỳ của chúng. Thiết bị nhập vào 732 có thể chứa giao diện màn chạm có thể là một phần của hoặc tách biệt với bộ phận hiển thị 736, như còn được thảo luận bên dưới. Thiết bị nhập vào 732 có thể được tận dụng như là thiết bị chọn của người sử dụng để chọn một hoặc nhiều biểu diễn đồ họa trong giao diện đồ họa như được mô tả ở trên.

Người sử dụng cũng có thể nhập vào các lệnh và/hoặc thông tin khác vào hệ thống máy tính 700 thông qua thiết bị lưu trữ 724 (ví dụ, ổ đĩa tháo ra được, ổ đĩa tác động nhanh, v.v.) và/hoặc thiết bị giao diện mạng 740. Thiết bị giao diện mạng, như thiết bị giao diện mạng 740, có thể được tận dụng để kết nối hệ thống máy tính 700 tới một hoặc nhiều thành phần trong số các mạng khác nhau, như mạng 744, và một hoặc nhiều thiết bị từ xa 748 được kết nối tới đó. Các ví dụ của thiết bị giao diện mạng chứa, nhưng không giới hạn ở, thẻ giao diện mạng (ví dụ, thẻ giao diện mạng di động, thẻ LAN), modem, và tổ hợp bất kỳ của chúng. Các ví dụ của mạng chứa, nhưng không giới hạn ở, mạng diện rộng (ví dụ, Internet, mạng doanh nghiệp), mạng diện cục bộ (ví dụ, mạng được kết hợp với văn phòng, tòa nhà, ký túc hoặc không gian địa lý tương đối nhỏ khác), mạng điện thoại, mạng dữ liệu được kết hợp với nhà cung cấp điện thoại/giọng nói (ví dụ, mạng dữ liệu và/hoặc giọng nói của nhà cung cấp truyền thông di động), kết nối trực tiếp giữa hai thiết bị tính toán, và các tổ hợp bất kỳ của chúng. Mạng, như mạng 744, có thể áp dụng chế độ liên lạc được nối dây và/hoặc không dây. Nói chung, cấu trúc liên kết mạng bất kỳ đều có thể được sử dụng. Thông tin (ví dụ, dữ liệu, phần mềm 720, v.v.) có thể được liên lạc tới và/hoặc từ hệ thống máy tính 700 thông qua thiết bị giao diện mạng 740.

Hệ thống máy tính 700 có thể còn chứa bộ làm thích ứng hiển thị video 752 để truyền thông hình ảnh có thể hiển thị được tới thiết bị hiển thị, như thiết bị hiển thị 736. Các ví dụ của thiết bị hiển thị chứa, nhưng không giới hạn ở, thiết bị hiển thị tinh thể lỏng (liquid crystal display - LCD), ống tia âm cực (cathode ray tube - CRT), thiết bị hiển thị plasma, thiết bị hiển thị diot phát quang (light emitting diode - LED), và các tổ hợp bất kỳ của chúng. Bộ thích ứng hiển thị 752 và thiết bị hiển thị 736 có thể được tận dụng trong kết

hợp với bộ xử lý 704 để tạo ra các sự biểu diễn về mặt đồ họa của các dấu hiệu của sáng chế này. Bên cạnh thiết bị hiển thị, hệ thống máy tính 700 có thể chứa một hoặc nhiều thiết bị đầu ra ngoại vi khác chứa, nhưng không giới hạn ở, loa audio, máy in, và các tổ hợp bất kỳ của chúng. Các thiết bị đầu ra ngoại vi này có thể được kết nối tới bus 712 thông qua giao diện ngoại vi 756. Các ví dụ của giao diện ngoại vi chứa, nhưng không giới hạn ở, cổng nối tiếp, kết nối USB, kết nối FIREWIRE, kết nối song song, và các tổ hợp bất kỳ của chúng.

Phần nêu trên là phần mô tả chi tiết của các phương án thực hiện minh họa của sáng chế này. Các cải biến và bổ sung khác nhau có thể được tạo ra mà không tách khỏi nguyên lý và phạm vi của sáng chế này. Các dấu hiệu của từng phương án thực hiện trong các phương án thực hiện khác nhau được mô tả ở trên có thể được kết hợp với các dấu hiệu của các phương án thực hiện được mô tả khác khi thích hợp để tạo ra vô số các tổ hợp dấu hiệu trong các phương án thực hiện mới, được kết hợp. Hơn nữa, mặc dù phần nêu trên mô tả nhiều phương án thực hiện tách biệt, nhưng phần được mô tả chỉ đơn thuần là minh họa của việc áp dụng của các nguyên lý của sáng chế này. Thêm vào đó, mặc dù các phương pháp cụ thể ở đây có thể được minh họa và/hoặc được mô tả như đang được thực hiện theo thứ tự cụ thể, nhưng việc đặt thứ tự này là có thể biến đổi lớn nằm trong hiểu biết của người có trình độ trung bình trong lĩnh vực để đạt được các phương án thực hiện như được bộc lộ ở đây. Theo đó, phần mô tả này có nghĩa là chỉ được lấy như là ví dụ, và theo cách khác là không làm hạn chế phạm vi bảo hộ của sáng chế.

Trong phần mô tả nêu trên và trong các yêu cầu bảo hộ, các cụm từ như “ít nhất một thành phần trong số” hoặc “một hoặc nhiều thành phần trong số” có thể xuất hiện theo sau bởi danh sách chi tiết của các phần tử hoặc các dấu hiệu. Thuật ngữ “và/hoặc” cũng có thể xuất hiện trong danh sách của hai hoặc hơn hai phần tử hoặc dấu hiệu. Trừ khi được chỉ ra theo cách khác ám chỉ hoặc trực tiếp mâu thuẫn bởi văn cảnh mà nó được sử dụng, cụm từ này là nhằm để có nghĩa là phần tử hoặc dấu hiệu bất kỳ trong số các phần tử hoặc dấu hiệu, theo cách riêng rẽ, hoặc phần tử hoặc dấu hiệu bất kỳ của các phần tử hoặc dấu hiệu được trích dẫn, theo cách kết hợp với phần tử hoặc dấu hiệu khác của các phần tử hoặc các

dấu hiệu được trích dẫn khác. Ví dụ, các cụm từ “ít nhất một trong số A và B;” “một hoặc nhiều thành phần trong số A và B;” và “A và/hoặc B” mỗi thành phần nhằm đề cập tới “A riêng rẽ, B riêng rẽ, hoặc A và B cùng với nhau.” Việc diễn giả tương tự cũng nhằm tới các danh sách chứa ba hoặc hơn ba mục. Ví dụ, các cụm từ “ít nhất một thành phần trong số A, B, và C;” “một hoặc nhiều thành phần của A, B, và C;” và “A, B, và/hoặc C” mỗi cụm từ nhằm nghĩa là “A riêng rẽ, B riêng rẽ, C riêng rẽ, A và B cùng với nhau, A và C cùng với nhau, B và C cùng với nhau, hoặc A và B và C cùng với nhau.” Thêm vào đó, việc sử dụng của thuật ngữ “dựa trên,” bên trên và trong các yêu cầu bảo hộ nhằm có nghĩa là, “dựa ít nhất một phần trên,” sao cho dấu hiệu hoặc phần tử không được trích dẫn là cũng có thể được cho phép.

Đối tượng được mô tả ở đây có thể được áp dụng trong các hệ thống, các thiết bị, các phương pháp, và/hoặc các vật phẩm phụ thuộc vào cấu hình mong muốn. Các áp dụng được nêu ở đây, trong phần mô tả nêu trên không thể hiện tất cả các áp dụng thống nhất với đối tượng được mô tả ở đây. Thay vào đó, chúng chỉ đơn thuần là các ví dụ thống nhất với các khía cạnh liên quan tới đối tượng được mô tả. Mặc dù một số ít biến thể đã được mô tả chi tiết ở trên, nhưng các điều chỉnh hoặc bổ sung khác là cũng khả thi. Cụ thể là, các dấu hiệu và/hoặc các thay đổi khác có thể được tạo ra bên cạnh các dấu hiệu và các thay đổi đã được chỉ ra ở đây. Ví dụ, các áp dụng được mô tả ở trên có thể được hướng tới các tổ hợp khác nhau và các tổ hợp phụ của các dấu hiệu đã được bộc lộ và/hoặc các tổ hợp và các tổ hợp phụ của nhiều dấu hiệu khác đã được bộc lộ ở trên. Thêm vào đó, các luồng logic được mô tả trong các hình vẽ kèm theo và/hoặc được mô tả ở đây không nhất thiết phải yêu cầu thứ tự cụ thể đã được thể hiện, hoặc thứ tự liên tiếp, để đạt được các kết quả mong muốn. Các áp dụng khác có thể nằm trong phạm vi của các yêu cầu bảo hộ kèm theo.

YÊU CẦU BẢO HỘ

1. Bộ giải mã, bộ giải mã này bao gồm hệ mạch được đặt cấu hình để:

nhận luồng bit;

xác định, cho khối hiện tại và sử dụng luồng bit, rằng chế độ hợp nhất được cho phép thực hiện;

xây dựng danh sách ứng viên hợp nhất, trong đó bước xây dựng danh sách ứng viên hợp nhất còn bao gồm việc thêm vectơ chuyển động toàn cục vào danh sách ứng viên vectơ chuyển động; và

xây dựng lại dữ liệu điểm ảnh của khối hiện tại sử dụng danh sách ứng viên vectơ chuyển động.

2. Bộ giải mã theo điểm 1, trong đó vectơ chuyển động toàn cục chứa loại mô hình chuyển động toàn cục đã được định nghĩa từ trước.

3. Bộ giải mã theo điểm 1, trong đó vectơ chuyển động toàn cục chứa vectơ chuyển động điểm điều khiển.

4. Bộ giải mã theo điểm 3, trong đó vectơ chuyển động điểm điều khiển là vectơ chuyển động tịnh tiến.

5. Bộ giải mã theo điểm 3, trong đó vectơ chuyển động điểm điều khiển là vectơ của mô hình chuyển động affin bốn thông số.

6. Bộ giải mã theo điểm 3, trong đó vectơ chuyển động điểm điều khiển là vectơ của mô hình chuyển động affin sáu thông số.

7. Bộ giải mã theo điểm 1, còn bao gồm:

bộ xử lý bộ giải mã entropy được đặt cấu hình để nhận luồng bit và giải mã luồng bit thành các hệ số được lượng tử hóa;

bộ xử lý lượng tử hóa ngược và biến đổi ngược được đặt cấu hình để xử lý các hệ số được lượng tử hóa chứa việc thực hiện cosin rời rạc ngược;

bộ lọc khử khối;

bộ đệm khung; và

bộ xử lý dự đoán nội.

8. Bộ giải mã theo điểm 1, trong đó khối hiện tại tạo thành một phần của cây quyết định tứ phân cộng nhị phân.

9. Bộ giải mã theo điểm 1, trong đó khối hiện tại là đơn vị cây mã hóa.

10. Bộ giải mã theo điểm 1, trong đó khối hiện tại là đơn vị mã hóa.

11. Bộ giải mã theo điểm 1, trong đó khối hiện tại là đơn vị dự đoán.

12. Bộ giải mã theo điểm 1, trong đó mô hình chuyển động toàn cục của vectơ chuyển động toàn cục chứa chuyển động tịnh tiến.

13. Bộ giải mã theo điểm 1, trong đó mô hình chuyển động toàn cục của vectơ chuyển động toàn cục chứa chuyển động afin.

14. Bộ giải mã theo điểm 9, trong đó mô hình chuyển động toàn cục được đặc trưng bởi phần đầu của luồng bit, phần đầu chứa tập hợp thông số hình ảnh.

15. Bộ giải mã theo điểm 9, trong đó mô hình chuyển động toàn cục được đặc trưng bởi phần đầu của luồng bit, phần đầu chứa tập hợp thông số chuỗi.

16. Phương pháp giải mã bao gồm các bước:

nhận, bởi bộ giải mã, luồng bit;

xác định, cho khối hiện tại và sử dụng luồng bit, rằng chế độ hợp nhất được cho phép thực hiện;

xây dựng danh sách ứng viên hợp nhất, trong đó bước xây dựng danh sách ứng viên hợp nhất còn bao gồm việc thêm vectơ chuyển động toàn cục vào danh sách ứng viên vectơ chuyển động; và

xây dựng lại dữ liệu điểm ảnh của khối hiện tại và sử dụng danh sách ứng viên vectơ chuyển động.

17. Phương pháp theo điểm 16, trong đó vectơ chuyển động toàn cục chứa loại mô hình chuyển động toàn cục đã được định nghĩa từ trước.

18. Phương pháp theo điểm 16, trong đó vectơ chuyển động toàn cục chứa vectơ chuyển động điểm điều khiển.

19. Phương pháp theo điểm 18, trong đó vectơ chuyển động điểm điều khiển là vectơ chuyển động tịnh tiến.

20. Phương pháp theo điểm 18, trong đó vectơ chuyển động điểm điều khiển là vectơ của mô hình chuyển động afin bốn thông số.

21. Phương pháp theo điểm 18, trong đó vectơ chuyển động điểm điều khiển là vectơ của mô hình chuyển động afin sáu thông số.

22. Phương pháp theo điểm 16, trong đó bộ giải mã còn bao gồm:

bộ xử lý bộ giải mã entropy được đặt cấu hình để nhận luồng bit và giải mã luồng bit thành các hệ số được lượng tử hóa;

bộ xử lý lượng tử hóa ngược và biến đổi ngược được đặt cấu hình để xử lý các hệ số được lượng tử hóa chứa việc thực hiện cosin rời rạc ngược;

bộ lọc khử khối;

bộ đệm khung; và

bộ xử lý dự đoán nội.

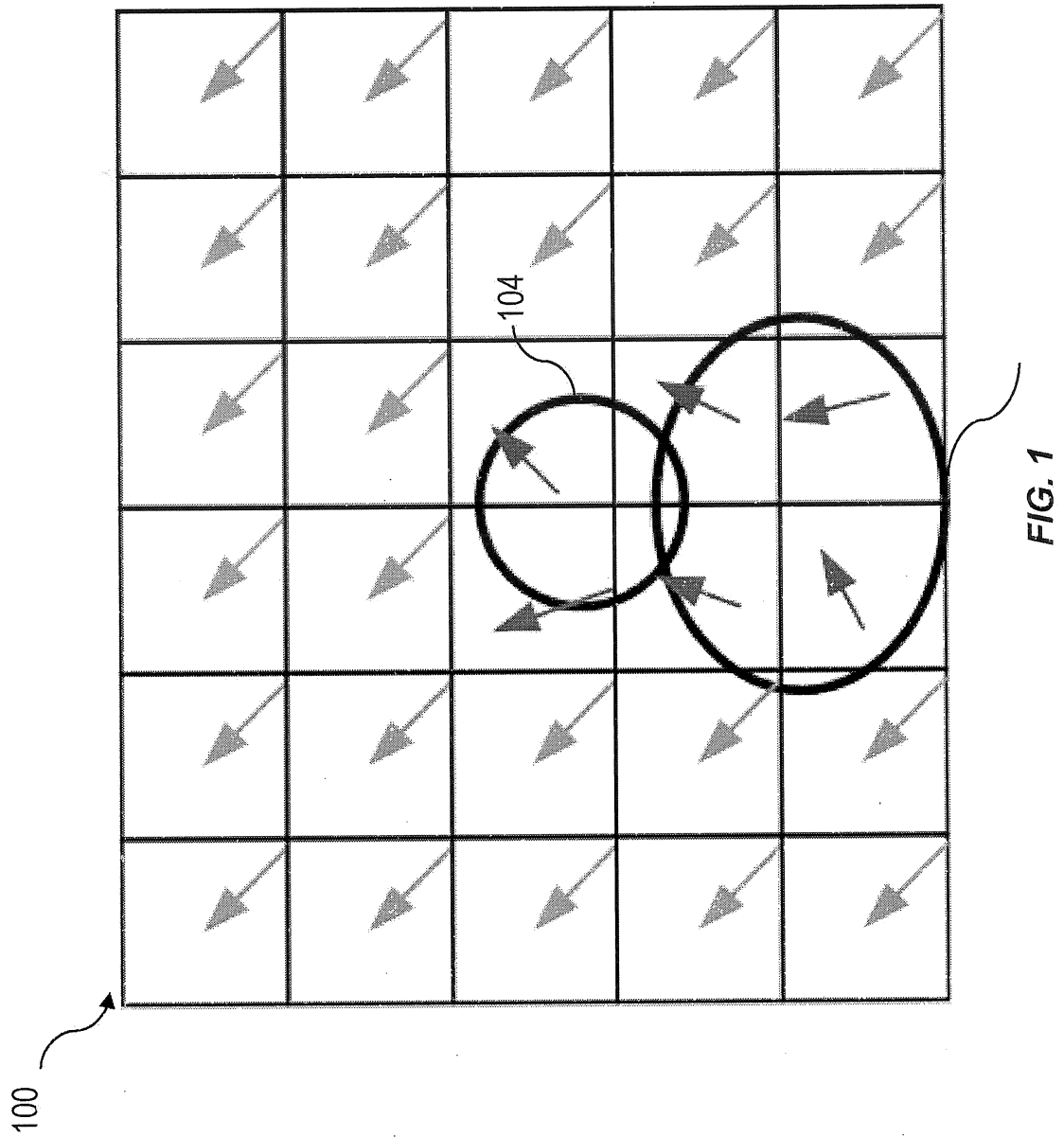
23. Phương pháp theo điểm 16, trong đó khối hiện tại tạo thành một phần của cây quyết định tứ phân cộng nhị phân.

24. Phương pháp theo điểm 16, trong đó khối hiện tại là đơn vị cây mã hóa.

25. Phương pháp theo điểm 16, trong đó khối hiện tại là đơn vị mã hóa.

26. Phương pháp theo điểm 16, trong đó khối hiện tại là đơn vị dự đoán.

27. Phương pháp theo điểm 16, trong đó mô hình chuyển động toàn cục của vectơ chuyển động toàn cục chứa chuyển động tịnh tiến.
28. Phương pháp theo điểm 16, trong đó mô hình chuyển động toàn cục của vectơ chuyển động toàn cục chứa chuyển động afin.
29. Phương pháp theo điểm 28, trong đó mô hình chuyển động toàn cục được đặc trưng bởi phần đầu của luồng bit, phần đầu chứa tập hợp thông số hình ảnh..
30. Phương pháp theo điểm 28, trong đó mô hình chuyển động toàn cục được đặc trưng bởi phần đầu của luồng bit, phần đầu chứa tập hợp thông số chuỗi.



200



MotionModelIdx[x][y]	Mô hình chuyển động để bù chuyển động
0	Chuyển động tịnh tiến
1	Chuyển động affin 4 thông số
2	Chuyển động affin 6 thông số

FIG. 2

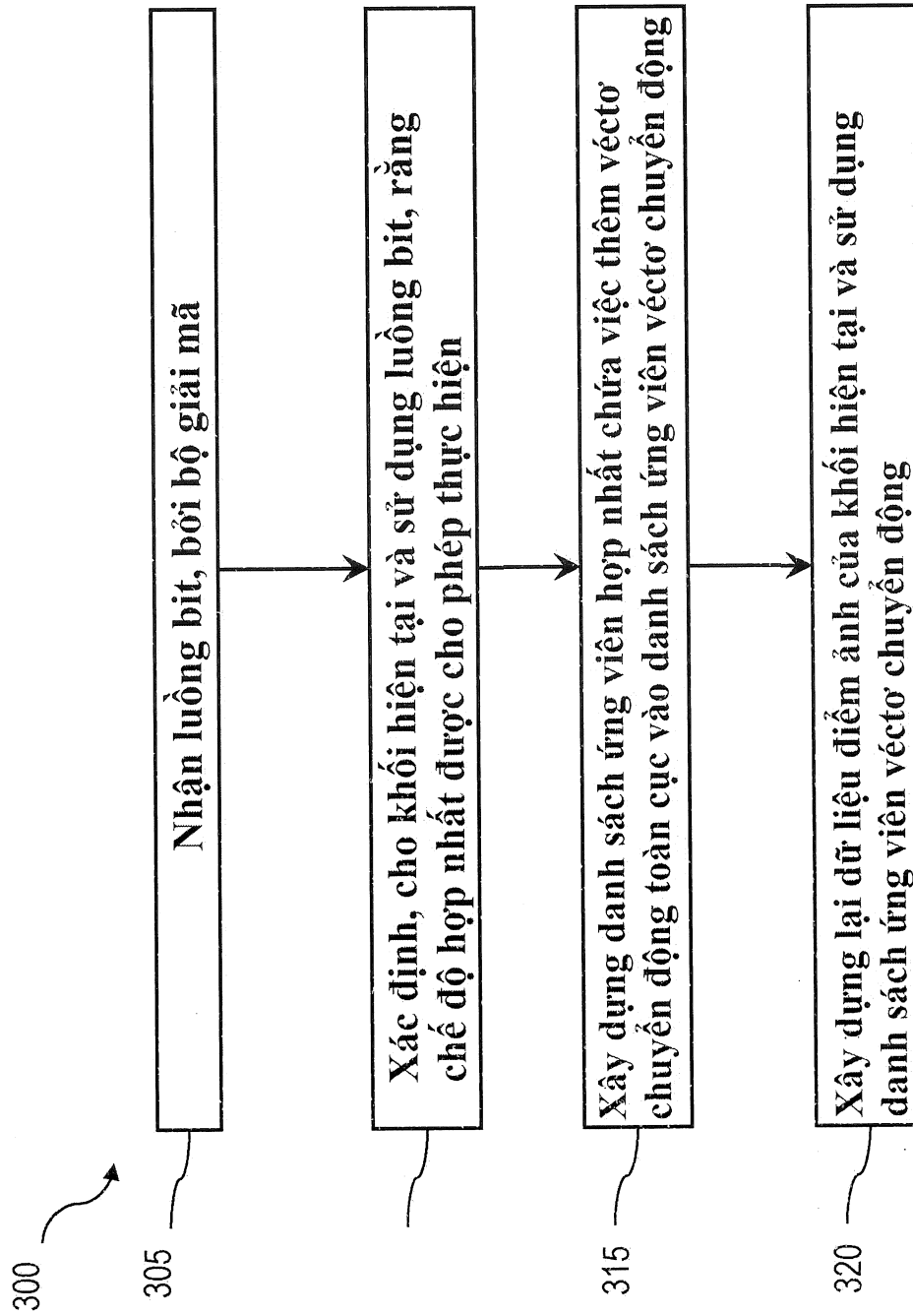
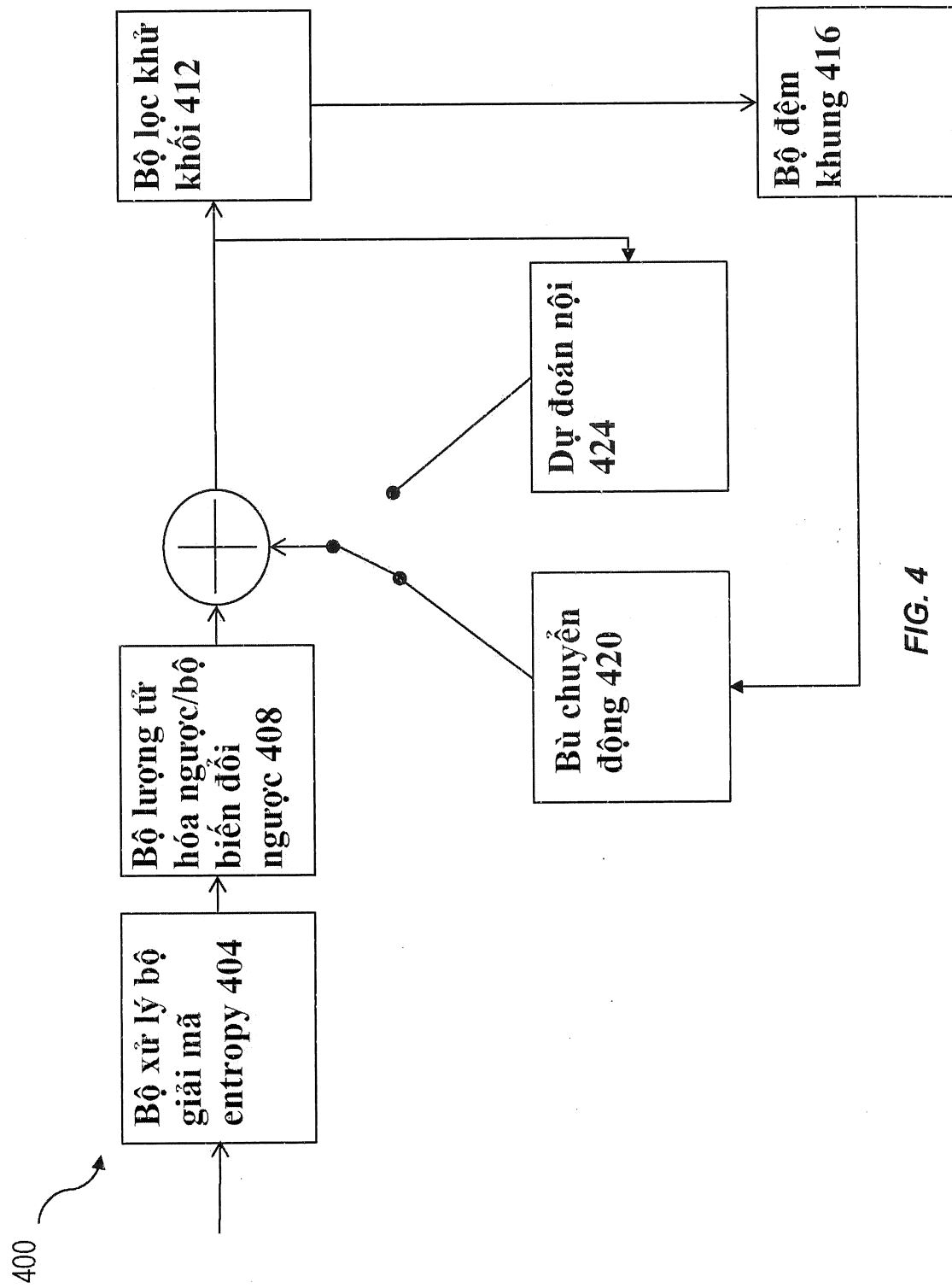


FIG. 3



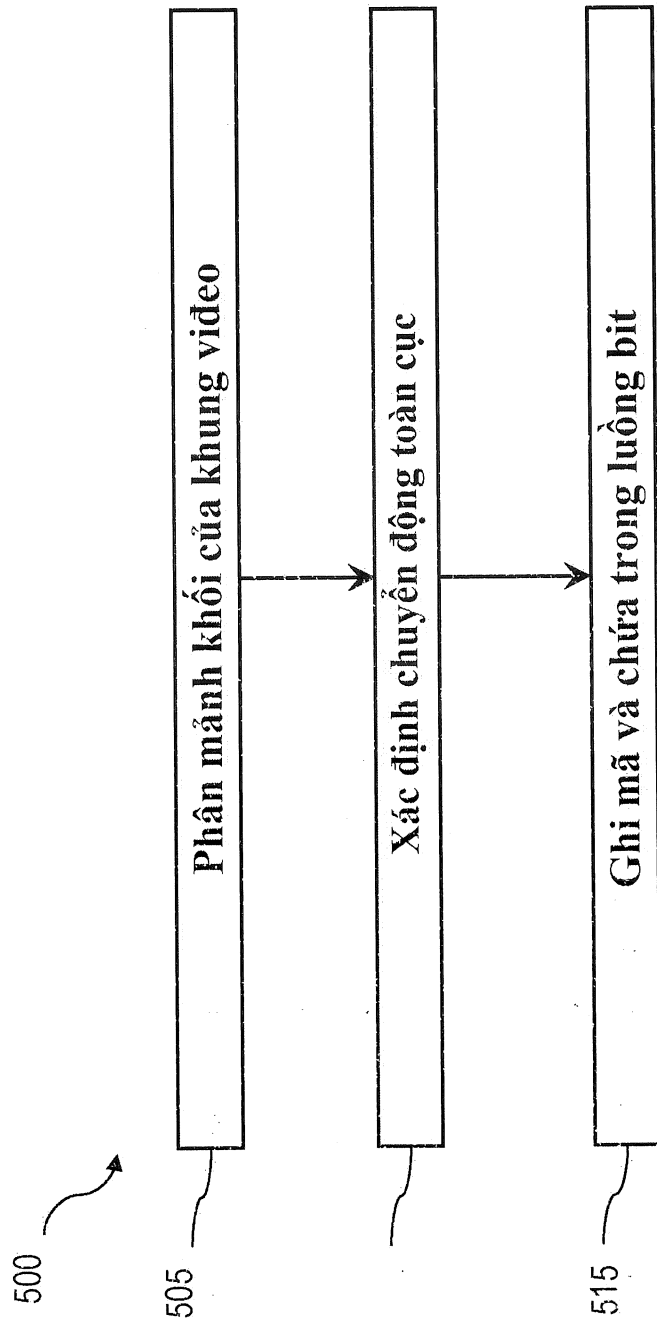


FIG. 5

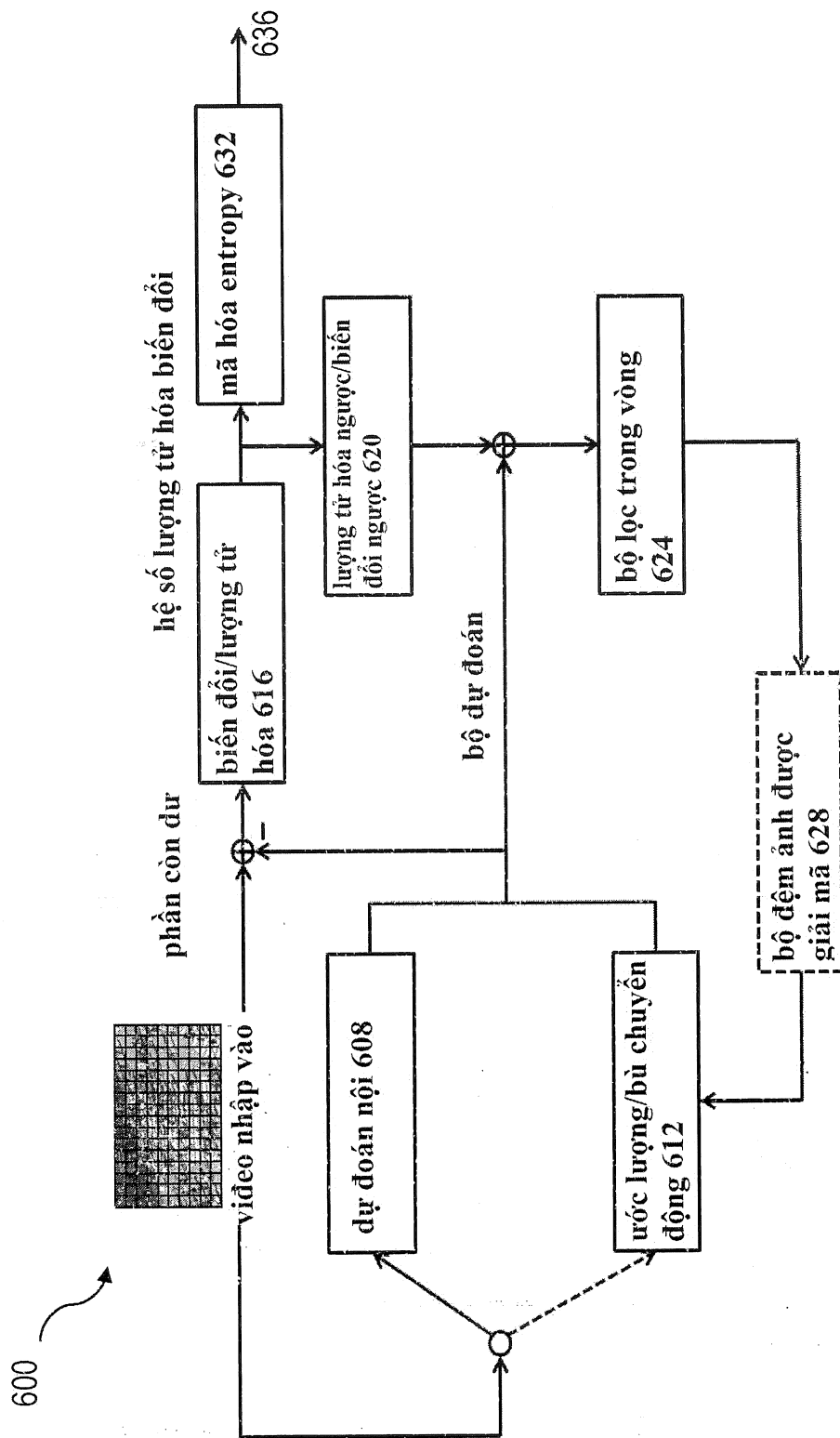


FIG. 6

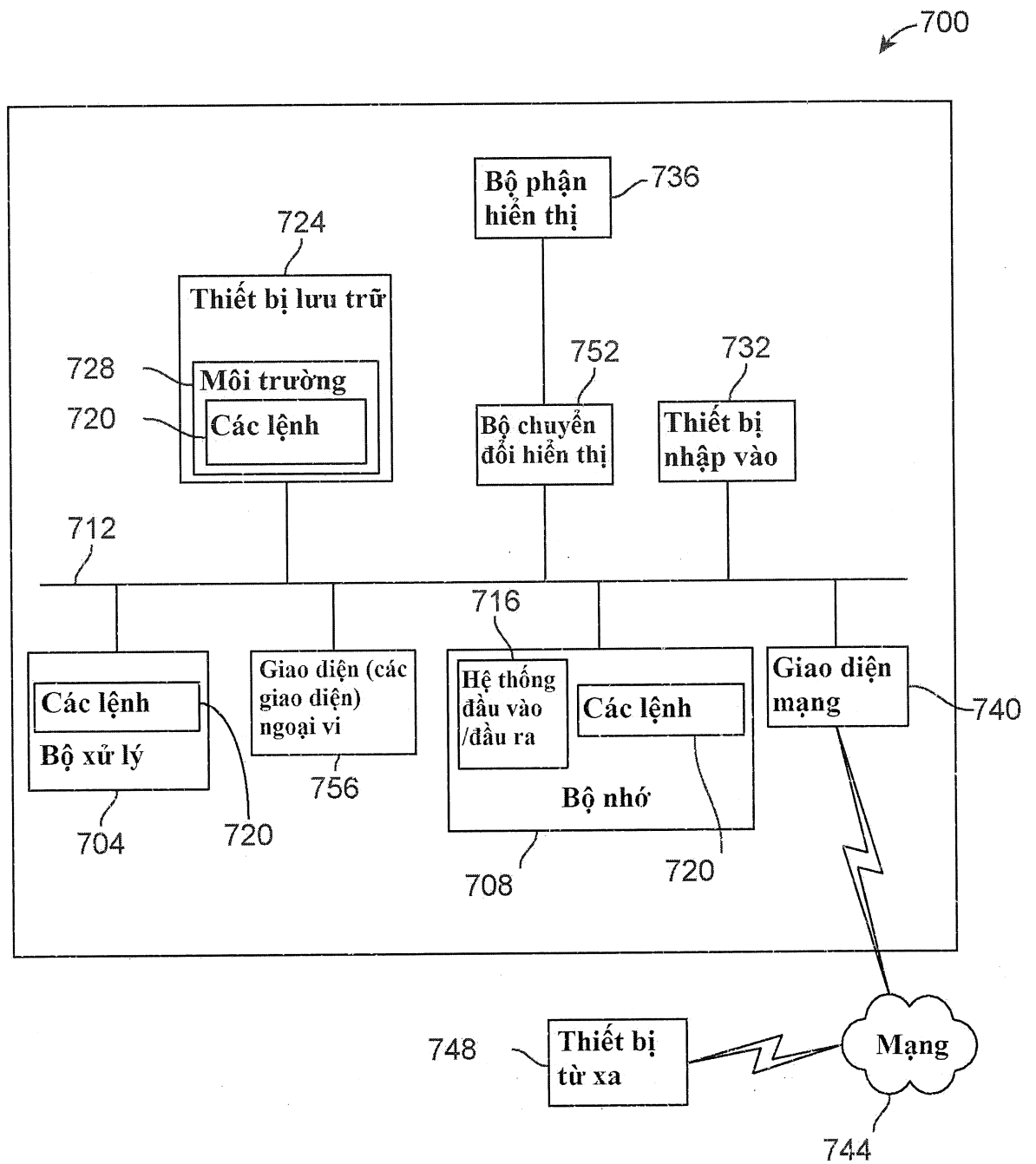


FIG. 7