



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ
(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN) (11) 
CỤC SỞ HỮU TRÍ TUỆ
1-0051200
(51)^{2020.01} H04N 19/105; H04N 19/573; H04N 19/139 (13) B

(21) 1-2022-02781 (22) 30/09/2020
(86) PCT/RU2020/050260 30/09/2020 (87) WO2020/251417 17/12/2020
(30) 62/908,594 30/09/2019 US; 62/912,049 07/10/2019 US; 62/927,671 29/10/2019 US;
PCT/CN2019/114161 29/10/2019 CN; 62/958,291 07/01/2020 US
(45) 25/09/2025 450 (43) 25/07/2022 412A
(73) HUAWEI TECHNOLOGIES CO., LTD. (CN)
Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong
518129, P. R. China
(72) SOLOVYEV, Timofey Mikhailovich (RU); SYCHEV, Maxim Borisovitch (RU);
CHEN, Huanbang (CN); KARABUTOV, Alexander Alexandrovich (RU);
CHERNYAK, Roman Igorevich (RU); IKONIN, Sergey Yurievich (RU); YANG,
Haitao (CN); ALSHINA, Elena Alexandrovna (RU).
(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP LẬP MÃ, BỘ MÃ HÓA, BỘ GIẢI MÃ, VÀ PHƯƠNG TIỆN
ĐỌC ĐƯỢC BỞI MÁY TÍNH KHÔNG TẠM THỜI

(21) 1-2022-02781

(57) Sáng chế đề cập đến phương pháp lập mã được thực hiện bởi thiết bị mã hóa/giải mã để lập mã dữ liệu video, bộ mã hóa, bộ giải mã, phương tiện lưu trữ đọc được bởi máy tính, và phương tiện đọc được bởi máy tính không tạm thời. Phương pháp bao gồm các bước sau đây: đối với khối được lập mã ở chế độ afin, xác định các vectơ chuyển động điểm điều khiển (control point motion vector, viết tắt là CPMV); xác định vùng tham chiếu trong ảnh tham chiếu tương ứng với khối con của khối được lập mã afin dựa vào các CPMV được xác định; thiết đặt biến clipMVX bằng đúng (TRUE) nếu kích thước của vùng tham chiếu lớn hơn ngưỡng được định rõ trước, theo cách khác thiết đặt biến clipMVX bằng sai (FALSE); dẫn ra trường vectơ chuyển động dựa trên điểm ảnh cho khối được lập mã afin; trong đó, nếu biến clipMVX bằng đúng, bước dẫn ra trường vectơ chuyển động dựa trên điểm ảnh còn bao gồm việc cắt vectơ chuyển động dựa vào khoảng cắt thứ nhất, trong đó khoảng cắt thứ nhất được xác định dựa vào các CPMV được xác định và kích thước của khối được lập mã afin.

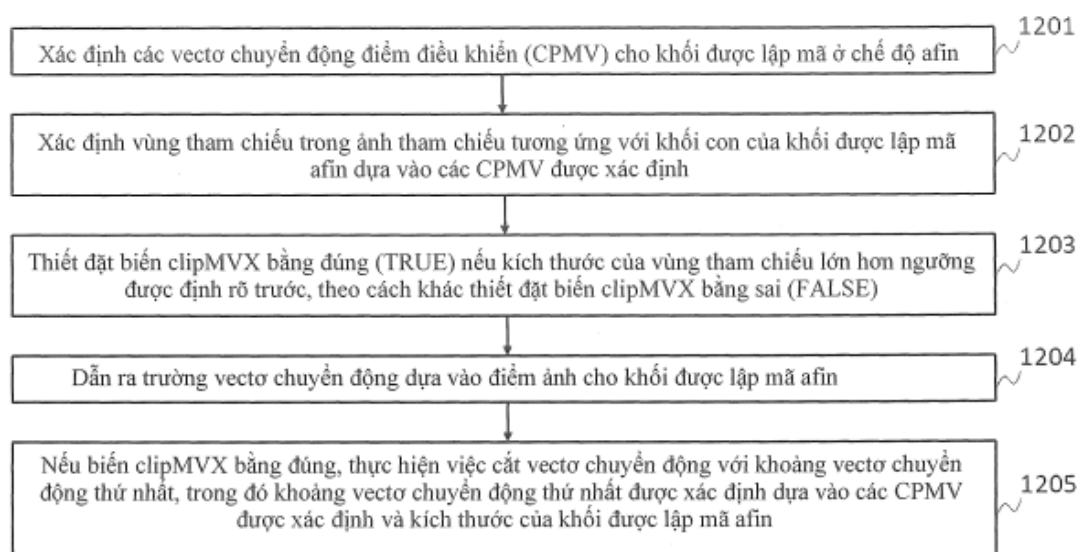


FIG.12

Lĩnh vực kỹ thuật được đề cập

Các phương án của sáng chế nói chung đề cập đến lĩnh vực về việc xử lý ảnh và cụ thể hơn là đến việc liên dự đoán, chẳng hạn như việc sử dụng thích nghi các bộ lọc nội suy trong bù chuyển động afin hoặc bù chuyển động khối con.

Tình trạng kỹ thuật của sáng chế

Lập mã video (mã hóa và giải mã video) được sử dụng trong khoảng rộng của các ứng dụng video số, ví dụ TV số phát rộng, truyền video qua Internet và các mạng di động, các ứng dụng đàm thoại thời gian thực chẳng hạn như trò chuyện video, hội nghị video, các đĩa DVD và Blu-ray, các hệ thống thu nhận và biên tập nội dung video, và các máy quay video của các ứng dụng bảo mật.

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

Bản chất kỹ thuật của sáng chế

Các phương án của sáng chế đề xuất các máy và các phương pháp để mã hóa và giải mã theo các điểm yêu cầu bảo hộ độc lập.

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

Các phương án cụ thể được khái quát trong các điểm yêu cầu bảo hộ độc lập kèm theo, với các phương án khác trong các điểm yêu cầu bảo hộ phụ thuộc.

Sáng chế đề xuất, theo khía cạnh thứ nhất, phương pháp lập mã được thực hiện bởi thiết bị mã hóa/giải mã để lập mã dữ liệu video, trong đó phương pháp này bao gồm các bước sau đây: đối với khối được lập mã ở chế độ afin, xác định các vector chuyển động điểm điều khiển (control point motion vector, viết tắt là CPMV); xác định vùng tham chiếu trong ảnh tham chiếu tương ứng với khối con của khối được lập mã afin dựa vào các CPMV được xác định; thiết đặt biến clipMVX bằng đúng (TRUE) nếu kích thước của vùng tham chiếu lớn hơn ngưỡng được định rõ trước, theo cách khác thiết đặt biến clipMVX bằng sai (FALSE); dẫn ra trường vector chuyển động dựa trên điểm ảnh cho khối được lập mã afin; trong đó, nếu biến clipMVX bằng đúng, bước dẫn ra trường vector chuyển động dựa trên điểm ảnh còn bao gồm việc cắt vector chuyển động với khoảng vector chuyển động thứ nhất, trong đó khoảng vector chuyển động thứ nhất được xác định dựa vào các CPMV được xác định và kích thước của khối được lập mã afin.

Vùng tham chiếu là vùng hình chữ nhật bao gồm tất cả các mẫu tham chiếu mà được cần để thực hiện việc bù chuyển động của khối hiện thời, như được mô tả trên Fig.9, có độ rộng W' và độ cao H' .

Phương pháp này có ưu điểm là các yêu cầu độ rộng dải bộ nhớ có thể được đáp ứng ngay cả khi vùng tham chiếu trong ảnh tham chiếu là lớn.

Theo dạng thực hiện của khía cạnh thứ nhất, việc xác định khoảng vector chuyển động thứ nhất bao gồm các bước: xác định vector chuyển động cho điểm trung tâm của khối được lập mã afin; xác định khoảng vector chuyển động thứ hai dựa vào vector chuyển động cho điểm trung tâm của khối được lập mã afin và các dịch vị được định rõ trước cho kích thước khối tương ứng, kích thước khối bao gồm kích thước của khối được lập mã afin. Điều này đưa ra cách thuận tiện để xác định khoảng cắt và tránh khỏi việc cắt không cần thiết trong trường hợp định tỷ lệ không đối xứng theo các hướng theo chiều ngang và theo chiều dọc được thực hiện. Với việc kiểm tra điều kiện được đề xuất việc định tỷ lệ MV được sử dụng chỉ trong trường hợp độ rộng dải bộ nhớ thực sự lớn hơn ngưỡng được thiết đặt là trị số được định rõ trước dựa vào trường hợp kém nhất độ rộng dải bộ nhớ mong muốn.

Theo dạng thực hiện khác của khía cạnh thứ nhất, bước xác định vector chuyển động cho điểm trung tâm của khối được lập mã afin được thực hiện bởi các phương trình sau đây:

$$\begin{aligned} mv_center[0] &= (mvBaseScaled[0] + dX[0] * (cbWidth >> 1) + dY[0] * (cbHeight >> 1)) \\ mv_center[1] &= (mvBaseScaled[1] + dX[1] * (cbWidth >> 1) + dY[1] * (cbHeight >> 1)), \end{aligned}$$

trong đó `mvBaseScaled` biểu diễn vector chuyển động của góc trên cùng bên trái của khối, `cbWidth` và `cbHeight` lần lượt biểu diễn độ rộng và độ cao của khối, `dX[ 0 ]`, `dX[1]` biểu diễn các độ chênh lệch tương ứng của các phần theo chiều ngang và theo chiều dọc của vector chuyển động trên một mẫu theo hướng chiều ngang, và `dY[ 0 ]`, `dY[1]` biểu diễn các độ chênh lệch tương ứng của các phần theo chiều ngang và theo chiều dọc của vector chuyển động trên một mẫu theo hướng chiều dọc; và `mv_center [ 0 ]` và `mv_center [ 1 ]` lần lượt biểu diễn các phần theo chiều ngang và theo chiều dọc của vector chuyển động cho điểm trung tâm.

Theo dạng thực hiện khác của khía cạnh thứ nhất, bước xác định khoảng vector chuyển động thứ hai dựa vào vector chuyển động cho điểm trung tâm của khối được lập mã affin và các dịch vị được định rõ trước cho kích thước khối tương ứng được thực hiện bởi các phương trình sau đây:

$$mv_hor_min = mv_center [0] - deviationMV [\log2CbWidth - 3]$$

$$mv_ver_min = mv_center [1] - deviationMV [\log2CbHeight - 3]$$

$$mv_hor_max = mv_center [0] + deviationMV [\log2CbWidth - 3]$$

$$mv_ver_max = mv_center [1] + deviationMV [\log2CbHeight - 3]$$

trong đó `deviationMV[]` biểu diễn bảng của các dịch vị được định rõ trước cho kích thước khối tương ứng, và `mv_hor_min`, `mv_ver_min`, `mv_hor_max` và `mv_ver_max` biểu diễn khoảng vector chuyển động thứ hai. Việc tính khoảng MV như vậy dựa vào vector chuyển động cho điểm trung tâm của khối và các dịch vị được định rõ trước hầu như không được yêu cầu độ phức tạp bổ sung bất kỳ và được đảm bảo rằng độ rộng dải bộ nhớ sẽ nhỏ hơn ngưỡng được định rõ trước. Nhưng đối với mô đun chuyển động affin 6 thông số phương pháp như vậy có thể gây ra việc cắt vector chuyển động không cần thiết nếu các hệ số định tỷ lệ không đối xứng được sử dụng theo chiều ngang và theo chiều dọc. Ví dụ nếu khối được định tỷ lệ với hệ số định tỷ lệ X theo chiều dọc và $1/X$ theo chiều ngang, mà không có các điều kiện bổ sung, chỉ thực hiện việc cắt, theo hợp giới hạn, được thu nhận dựa vào vector chuyển động cho điểm trung tâm của khối và các dịch vị được định rõ trước, tương ứng với độ rộng/độ cao của khối, phần quan trọng của khối được biến đổi sẽ bị thiếu do việc cắt MV; kết quả là, chất lượng dự đoán sẽ kém; trong khi theo ví dụ cụ thể này vùng tham chiếu của khối hầu như giống với vùng của chính khối do $(1/X)*X$ bằng 1, vì vậy độ rộng dải bộ nhớ sát với 1 (điều đó hoàn toàn không phải là vấn đề) vì vậy việc cắt vector chuyển động được thực hiện để giới hạn độ rộng dải bộ nhớ không được cần trong trường

hợp này. Không áp dụng việc cắt vector chuyển động bổ sung trong các trường hợp, khi độ rộng dải bộ nhớ nhỏ hơn ngưỡng độ rộng dải bộ nhớ được yêu cầu, đưa ra việc nâng cao hiệu quả lập mã video khi việc cắt vector chuyển động không cần thiết bổ sung gây ra việc suy giảm chất lượng trường chuyển động mà lần lượt gây ra việc suy giảm chất lượng tín hiệu được tái cấu trúc và được dự đoán. Với phương pháp, được đề xuất theo sáng chế, việc cắt MV được thực hiện chỉ khi nó thực sự được cần và bằng cách làm như vậy hiệu quả lập mã so với việc cắt MV được áp dụng vô điều kiện được nâng cao.

Theo dạng thực hiện khác của khía cạnh thứ nhất, các dịch vị được định rõ trước cho các kích thước khối tương ứng tỷ lệ với { 64, 128, 272, 560, 1136 }, trong đó 64 tương ứng với kích thước khối có kích thước tương ứng bằng 8, 128 tương ứng với kích thước khối có kích thước tương ứng bằng 16, 272 tương ứng với kích thước khối có kích thước tương ứng bằng 32, 560 tương ứng với kích thước khối có kích thước tương ứng bằng 64, hoặc 1136 tương ứng với kích thước khối có kích thước tương ứng bằng 128. Ở đây kích thước khối có thể đề cập đến độ rộng/độ cao khối. Quy trình xử lý này có thể được thực hiện một cách riêng biệt đối với các phần theo chiều ngang và theo chiều dọc của vector chuyển động. Nếu khối, ví dụ, có độ rộng bằng 8 và độ cao bằng 16 sau đó đối với phần theo chiều ngang của vector chuyển động dịch vị sẽ tương ứng với 8 và đối với phần theo chiều dọc của vector chuyển động dịch vị sẽ tương ứng với 16. Do đó, điều này cũng áp dụng cho các khối hình chữ nhật, không chỉ cho các khối hình vuông.

Theo dạng thực hiện khác của khía cạnh thứ nhất, sau khi xác định khoảng vector chuyển động thứ nhất, khoảng vector chuyển động thứ nhất được định tỷ lệ theo các thông số lấy mẫu phụ SubWidthC và SubHeightC:

- hor_min = hor_min / SubWidthC
- hor_max = hor_max / SubWidthC
- ver_min = ver_min / SubHeightC
- ver_max = ver_max / SubHeightC

trong đó hor_min, ver_min, hor_max và ver_max biểu diễn khoảng vector chuyển động thứ nhất được định tỷ lệ. Điều quan trọng ở đây là việc định tỷ lệ như vậy được thực hiện sau khi dẫn ra khoảng vector chuyển động thứ nhất. Nó giúp tránh khỏi các trường hợp chroma (sắc độ) và luma (độ chói) phân biệt logic bổ sung bên trong mô đun xác định khoảng vector chuyển động thứ nhất.

Theo dạng thực hiện khác của khía cạnh thứ nhất, phương pháp còn bao gồm các bước: thu nhận các mẫu được nội suy trong ảnh tham chiếu dựa vào trường vector chuyển động

được dẫn ra nhờ sử dụng nội suy song tuyến tính; và áp dụng bộ lọc thông cao cho các mẫu được nội suy. Theo đó phương pháp có thể áp dụng có lợi cho việc bù chuyển động afin nhờ sử dụng bộ lọc nội suy nâng cao (enhanced interpolation filter, viết tắt là EIF).

Việc sử dụng bù chuyển động trên cơ sở điểm ảnh khiến các nỗ lực tính toán bổ sung do số các kết quả thao tác ít hơn có thể được sử dụng lại trong suốt thời gian quy trình nội suy. Đó là lý do tại sao việc bù chuyển động trên cơ sở điểm ảnh hiếm khi được sử dụng với ví dụ, các bộ lọc dựa trên DCT 8 nhánh mà ngược lại được sử dụng rộng rãi cho việc bù chuyển động trên cơ sở khối. Việc sử dụng bù chuyển động trên cơ sở điểm ảnh với nội suy song tuyến tính cho phép giữ lại và trong một số trường hợp thậm chí làm giảm độ phức tạp so với việc bù chuyển động trên cơ sở khối với các bộ lọc nội suy dựa trên DCT 8 nhánh, nhưng bộ lọc ngắn hơn tạo ra các tạo tác làm mờ thấy được mà đã làm cho việc bù chuyển động trên cơ sở điểm ảnh như vậy không thực tế. Đồng thời việc bổ sung bộ lọc thông cao 3 nhánh đơn giản với các hệ số cố định sau khi việc nội suy song tuyến tính cho phép đạt được việc nâng cao chủ quan đáng chú ý của tín hiệu được tái cấu trúc với độ phức tạp có thể quản lý.

Theo dạng thực hiện khác của khía cạnh thứ nhất, khối con của khối được lập mã afin có kích thước 4x4.

Theo dạng thực hiện khác của khía cạnh thứ nhất, ngưỡng được định rõ trước là 72.

Theo dạng thực hiện khác của khía cạnh thứ nhất, trong trường hợp rằng việc liên dự đoán afin bao gồm việc dự đoán đôi, bước thiết đặt biến clipMVX bao gồm:

dẫn ra biến clipMV0 cho danh mục 0 (clipMVX, $X = 0$)

dẫn ra biến clipMV1 cho danh mục 1 (clipMVX, $X = 1$)

dẫn ra biến clipMV là clipMV0 | clipMV1, trong đó “|” nghĩa là HOẶC (OR).

Sáng chế đề xuất, theo khía cạnh thứ hai, bộ mã hóa bao gồm hệ mạch xử lý để tiến hành phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện bất kỳ của nó.

Sáng chế đề xuất, theo khía cạnh thứ ba, bộ giải mã bao gồm hệ mạch xử lý để tiến hành phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện bất kỳ của nó.

Sáng chế đề xuất, theo khía cạnh thứ tư, sản phẩm chương trình máy tính bao gồm các lệnh mà, khi chương trình được thực thi bởi máy tính, khiến máy tính tiến hành phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện bất kỳ của nó.

Sáng chế đề xuất, theo khía cạnh thứ năm, bộ giải mã, bao gồm một hoặc nhiều bộ xử lý; và phương tiện ghi đọc được bởi máy tính không tạm thời được ghép nối với một hoặc

nhiều bộ xử lý và lưu trữ các lệnh cho việc thực thi bởi một hoặc nhiều bộ xử lý, trong đó bước lập trình, khi được thực thi bởi một hoặc nhiều bộ xử lý, tạo cấu hình bộ giải mã để tiến hành phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện bất kỳ của nó.

Sáng chế đề xuất, theo khía cạnh thứ sáu, bộ mã hóa, bao gồm một hoặc nhiều bộ xử lý; và phương tiện ghi đọc được bởi máy tính không tạm thời được ghép nối với một hoặc nhiều bộ xử lý và lưu trữ các lệnh cho việc thực thi bởi một hoặc nhiều bộ xử lý, trong đó bước lập trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ mã hóa để tiến hành phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện bất kỳ của nó.

Sáng chế đề xuất, theo khía cạnh thứ bảy, phương tiện đọc được bởi máy tính không tạm thời mang mã chương trình mà, khi được thực thi bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp theo khía cạnh bất kỳ trong số khía cạnh thứ nhất hoặc dạng thực hiện bất kỳ của nó.

Sáng chế đề xuất, theo khía cạnh thứ tám, bộ giải mã hoặc bộ mã hóa để lập mã chuỗi video, bao gồm bộ phận xác định được tạo cấu hình xác định các vector chuyển động điểm điều khiển (CPMV), cho khối được lập mã afin được lập mã ở chế độ afin và để xác định vùng tham chiếu trong ảnh tham chiếu tương ứng với khối con của khối được lập mã afin dựa vào các CPMV được xác định; và bộ phận dự đoán được tạo cấu hình để thiết đặt biến clipMVX bằng đúng nếu kích thước của vùng tham chiếu lớn hơn ngưỡng được định rõ trước, theo cách khác thiết đặt biến clipMVX bằng sai; dẫn ra trường vector chuyển động dựa trên điểm ảnh cho khối được lập mã afin, trong đó, nếu biến clipMVX bằng đúng, bước dẫn ra trường vector chuyển động dựa trên điểm ảnh còn bao gồm việc cắt vector chuyển động với khoảng vector chuyển động thứ nhất, trong đó khoảng vector chuyển động thứ nhất được xác định dựa vào các CPMV được xác định và kích thước của khối được lập mã afin.

Các ưu điểm của bộ giải mã hoặc bộ mã hóa theo khía cạnh thứ tám hoặc dạng thực hiện bất kỳ trong số các dạng thực hiện sau đây của chúng tương ứng với các ưu điểm tương ứng của phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện của chúng.

Theo dạng thực hiện của khía cạnh thứ tám, bước xác định khoảng vector chuyển động thứ nhất bao gồm việc xác định vector chuyển động cho điểm trung tâm của khối được lập mã afin; xác định khoảng vector chuyển động thứ hai dựa vào vector chuyển động cho điểm trung tâm của khối được lập mã afin và các dịch vị được định rõ trước cho kích thước khối tương ứng, kích thước khối bao gồm kích thước của khối được lập mã afin.

Theo dạng thực hiện khác của khía cạnh thứ tám, bước xác định vector chuyển động cho điểm trung tâm của khối được lập mã affin được thực hiện bởi các phương trình sau đây:

$$\begin{aligned} mv_center[0] &= (mvBaseScaled[0] + dX[0] * (cbWidth >> 1) + dY[0] * (cbHeight >> 1)) \\ mv_center[1] &= (mvBaseScaled[1] + dX[1] * (cbWidth >> 1) + dY[1] * (cbHeight >> 1)), \end{aligned}$$

trong đó $mvBaseScaled$ biểu diễn vector chuyển động của góc trên cùng bên trái của khối, $cbWidth$ và $cbHeight$ lần lượt biểu diễn độ rộng và độ cao của khối, $dX[0]$, $dX[1]$ biểu diễn các độ chênh lệch tương ứng của các phần theo chiều ngang và theo chiều dọc của vector chuyển động trên một mẫu theo hướng chiều ngang, và $dY[0]$, $dY[1]$ biểu diễn các độ chênh lệch tương ứng của các phần theo chiều ngang và theo chiều dọc của vector chuyển động trên một mẫu theo hướng chiều dọc; và $mv_center[0]$ và $mv_center[1]$ lần lượt biểu diễn các phần theo chiều ngang và theo chiều dọc của vector chuyển động cho điểm trung tâm.

Theo dạng thực hiện khác của khía cạnh thứ tám, bước xác định khoảng vector chuyển động thứ hai dựa vào vector chuyển động cho điểm trung tâm của khối được lập mã affin và các dịch vị được định rõ trước cho kích thước khối tương ứng được thực hiện bởi các phương trình sau đây:

$$\begin{aligned} mv_hor_min &= mv_center[0] - deviationMV[\log_2 CbWidth - 3] \\ mv_ver_min &= mv_center[1] - deviationMV[\log_2 CbHeight - 3] \\ mv_hor_max &= mv_center[0] + deviationMV[\log_2 CbWidth - 3] \\ mv_ver_max &= mv_center[1] + deviationMV[\log_2 CbHeight - 3] \end{aligned}$$

trong đó $deviationMV[]$ biểu diễn bảng của các dịch vị được định rõ trước cho kích thước khối tương ứng, và mv_hor_min , mv_ver_min , mv_hor_max và mv_ver_max biểu diễn khoảng vector chuyển động thứ hai.

Theo dạng thực hiện khác của khía cạnh thứ tám, các dịch vị được định rõ trước cho các kích thước khối tương ứng tỷ lệ với $\{64, 128, 272, 560, 1136\}$, trong đó 64 tương ứng với kích thước khối có kích thước tương ứng bằng 8, 128 tương ứng với kích thước khối có kích thước tương ứng bằng 16, 272 tương ứng với kích thước khối có kích thước tương ứng bằng 32, 560 tương ứng với kích thước khối có kích thước tương ứng bằng 64, hoặc 1136 tương ứng với kích thước khối có kích thước tương ứng bằng 128. “Tỷ lệ” này nghĩa là độ chính xác MV có thể khác nhau nhưng sẽ tỷ lệ với các trị số được đưa ra trong bảng. Ví dụ, với hệ số là 2, các dịch vị có thể được định rõ bởi $\{128, 272, 560, 1136, 2272\}$.

Theo dạng thực hiện khác của khía cạnh thứ tám, sau khi xác định khoảng vector chuyển động thứ nhất, khoảng vector chuyển động thứ nhất được định tỷ lệ theo các thông số lấy mẫu phụ SubWidthC và SubHeightC:

- $\text{hor_min} = \text{hor_min} / \text{SubWidthC}$
- $\text{hor_max} = \text{hor_max} / \text{SubWidthC}$
- $\text{ver_min} = \text{ver_min} / \text{SubHeightC}$
- $\text{ver_max} = \text{ver_max} / \text{SubHeightC}$

trong đó hor_min, ver_min, hor_max và ver_max biểu diễn khoảng vector chuyển động thứ nhất được định tỷ lệ.

Theo dạng thực hiện khác của khía cạnh thứ tám, bộ mã hóa hoặc bộ giải mã được tạo cấu hình để thực hiện các bước để thu nhận các mẫu được nội suy trong ảnh tham chiếu dựa vào trường vector chuyển động được dẫn ra nhờ sử dụng nội suy song tuyến tính; và áp dụng bộ lọc thông cao cho các mẫu được nội suy.

Theo dạng thực hiện khác của khía cạnh thứ tám, khối con của khối được lập mã afin có kích thước 4x4.

Theo dạng thực hiện khác của khía cạnh thứ tám, ngưỡng được định rõ trước là 72.

Theo dạng thực hiện khác của khía cạnh thứ tám, trong trường hợp rằng việc liên dự đoán afin bao gồm việc dự đoán đôi, bước thiết đặt biến clipMVX bao gồm:

dẫn ra biến clipMV0 cho danh mục 0

dẫn ra biến clipMV1 cho danh mục 1

dẫn ra biến clipMV là clipMV0 | clipMV1, trong đó “|” nghĩa là HOẶC.

Theo ví dụ, sáng chế đề cập đến phương pháp dùng cho việc liên dự đoán của khối hình ảnh hiện thời trong ảnh hiện thời của video, phương pháp được sử dụng bởi thiết bị mã hóa/giải mã phương pháp bao gồm các bước:

tính kích thước khối con MxN dựa vào các thông số mô hình chuyển động afin hoặc dựa vào thông tin mà từ đó các thông số mô hình chuyển động afin có thể được dẫn ra;

trong trường hợp rằng hoặc độ rộng khối con M hoặc độ cao khối con N nhỏ hơn hoặc bằng trị số được định rõ trước, thực hiện quy trình bù chuyển động bộ lọc nội suy song tuyến tính nâng cao (enhanced bi-linear Interpolation Filter, viết tắt là EIF), trong đó bước thực hiện quy trình bù chuyển động EIF bao gồm:

dẫn ra vector chuyển động của khối con tương ứng của khối hình ảnh (chẳng hạn như khối hình ảnh afin) dựa vào các thông số mô hình chuyển động afin trên cơ sở khối con PxQ (chẳng hạn như 1x1); và

thực hiện việc cắt trên vector chuyển động của khối con, sao cho vector chuyển động được cắt trong khoảng vector chuyển động (chẳng hạn như khoảng vector chuyển động thứ hai).

Theo ví dụ khác, sáng chế đề cập đến máy dùng cho việc liên dự đoán của khối hình ảnh hiện thời trong ảnh hiện thời của máy video bao gồm:

môđun thứ nhất được tạo cấu hình để tính kích thước khối con $M \times N$ dựa vào các thông số mô hình chuyển động afin hoặc dựa vào thông tin mà từ đó các thông số mô hình chuyển động afin có thể được dẫn ra;

môđun thứ hai được tạo cấu hình để thực hiện, trong trường hợp rằng hoặc độ rộng khối con M hoặc độ cao khối con N nhỏ hơn hoặc bằng trị số được định rõ trước, quy trình bù chuyển động bộ lọc nội suy song tuyến tính nâng cao (EIF), trong đó bước thực hiện quy trình bù chuyển động EIF bao gồm: dẫn ra vector chuyển động của khối con tương ứng của khối hình ảnh (chẳng hạn như khối hình ảnh afin) dựa vào các thông số mô hình chuyển động afin trên cơ sở khối con $P \times Q$ (chẳng hạn như 1×1); và thực hiện việc cắt trên vector chuyển động của khối con, sao cho vector chuyển động được cắt trong khoảng vector chuyển động (chẳng hạn như khoảng vector chuyển động thứ hai).

Phương pháp theo ví dụ có thể được thực hiện bởi máy theo ví dụ còn lại.

Chi tiết về một hoặc nhiều phương án được đưa ra trên các hình vẽ và phần mô tả kèm theo dưới đây. Các dấu hiệu, các mục đích, và các ưu điểm khác sẽ rõ ràng từ phần mô tả, các hình vẽ, và các điểm yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

Các phương án sau đây của sáng chế được mô tả chi tiết hơn dựa vào các số liệu và các hình vẽ kèm theo, trong đó:

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

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

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

Fig.4 là sơ đồ khối minh họa ví dụ về máy mã hóa hoặc máy giải mã;

Fig.5 là sơ đồ khối minh họa ví dụ khác về máy mã hóa hoặc máy giải mã;

Fig.6 là ví dụ minh họa về mô hình chuyển động afin trên cơ sở điểm điểm khiên: 4 thông số và 6 thông số

Fig.7 là ví dụ minh họa về trường vector chuyển động khối con afin.

Fig.8 là ví dụ minh họa về các tọa độ của các góc của khối (khối con) được lập mã afin và của khối (khối con) EIF trung gian.

Fig.9 là ví dụ minh họa về vị trí của khối (khối con) được biến đổi trong ảnh tham chiếu và hộp giới hạn tương ứng.

Fig.10 là sơ đồ khối thể hiện cấu trúc ví dụ của hệ thống cung cấp nội dung 3100 mà thực hiện dịch vụ phân phối nội dung.

Fig.11 là sơ đồ khối thể hiện cấu trúc theo ví dụ về thiết bị đầu cuối.

Fig.12 là sơ đồ khối minh họa phương pháp theo khía cạnh thứ nhất của sáng chế.

Fig.13 là sơ đồ khối minh họa bộ giải mã và bộ mã hóa theo khía cạnh thứ năm và thứ sáu của sáng chế.

Fig.14 là sơ đồ khối minh họa bộ giải mã và bộ mã hóa theo khía cạnh thứ tám của sáng chế.

Trong phần sau đây, các ký hiệu chỉ dẫn giống nhau đề cập đến các dấu hiệu giống nhau hoặc ít nhất tương đương về mặt chức năng nếu không được quy định rõ ràng theo cách khác.

Mô tả chi tiết sáng chế

Trong phần mô tả sau đây, tham chiếu được thực hiện tới các hình vẽ kèm theo, mà tạo nên phần của sáng chế, và thể hiện, bằng cách minh họa, các khía cạnh cụ thể của các phương án của sáng chế hoặc các khía cạnh cụ thể trong đó các phương án của sáng chế có thể được sử dụng. Được hiểu rằng các phương án của sáng chế có thể được sử dụng theo các khía cạnh khác và bao gồm các sự thay đổi logic và cấu trúc không được mô tả trên các hình vẽ. Phần mô tả chi tiết sau đây, do đó, không được coi là giới hạn, và phạm vi của sáng chế được định rõ bởi các điểm yêu cầu bảo hộ kèm theo.

Ví dụ, được hiểu rằng sáng chế kết hợp 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. Ví dụ, nếu một hoặc nhiều bước về phương pháp cụ thể được mô tả, thiết bị tương ứng có thể bao gồm một hoặc nhiều bộ phận, ví dụ, các bộ phận chức năng, để thực hiện một hoặc nhiều bước về phương pháp được mô tả (ví dụ, một bộ phận thực hiện một

hoặc nhiều bước, hoặc nhiều bộ phận đều thực hiện một hoặc nhiều trong số các bước), ngay cả khi một hoặc nhiều bộ phận như vậy không được mô tả hoặc được minh họa một cách rõ ràng trên các hình vẽ. Nói cách khác, ví dụ, nếu máy cụ thể được mô tả dựa vào một hoặc nhiều bộ phận, ví dụ, các bộ phận chức năng, phương pháp tương ứng có thể bao gồm một bước để thực hiện tính năng của một hoặc nhiều bộ phận (ví dụ, một bước thực hiện tính năng của một hoặc nhiều bộ phận, hoặc nhiều bước đều thực hiện tính năng của một hoặc nhiều trong số các bộ phận), ngay cả khi một hoặc nhiều bước như vậy không được mô tả hoặc được minh họa một cách rõ ràng trên các hình vẽ. Hơn nữa, được hiểu rằng các dấu hiệu của các phương án và/hoặc các khía cạnh làm ví dụ khác nhau được mô tả ở đây có thể được kết hợp với nhau, trừ khi được lưu ý cụ thể theo cách khác.

Lập mã video thường đề cập đến việc xử lý của chuỗi các ảnh, mà tạo nên video hoặc chuỗi video. Thay vì thuật ngữ “ảnh” thuật ngữ “khung” hoặc “hình ảnh” có thể được sử dụng như các từ đồng nghĩa trong lĩnh vực lập mã video. Lập mã video (hoặc lập 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 việc xử lý (ví dụ, bởi việc nén) các ảnh video ban đầu để làm giảm lượng dữ liệu được yêu cầu để biểu diễn các ảnh video (cho việc truyền và/hoặc lưu trữ hữu hiệu hơn). Giải mã video được thực hiện ở phía đích và thường bao gồm việc xử lý ngược so với bộ mã hóa để tái cấu trúc các ảnh video. Các phương án đề cập đến việc “lập mã” của các ảnh video (hoặc các ảnh nói chung) sẽ được hiểu liên quan đến việc “mã hóa” hoặc “giải mã” của các ảnh video hoặc các chuỗi video tương ứng. Sự kết hợp của phần mã hóa và phần giải mã cũng được đề cập đến là CODEC (lập mã và giải mã).

Trong trường hợp lập mã video mất mát, các ảnh video ban đầu có thể được tái cấu trúc, nghĩa là, các ảnh video được tái cấu trúc có chất lượng giống như các ảnh video ban đầu (giả sử không có sự mất mát truyền hoặc sự mất mát dữ liệu khác trong suốt thời gian lưu trữ hoặc truyền). Trong trường hợp lập mã video mất mát, việc nén thêm, ví dụ, bởi việc lượng tử hóa, được thực hiện, để làm giảm lượng dữ liệu biểu diễn các ảnh video, mà không thể được tái cấu trúc hoàn toàn ở bộ giải mã, nghĩa là, chất lượng của các ảnh video được tái cấu trúc thấp hơn hoặc kém hơn so với chất lượng của các ảnh video ban đầu.

Nhiều chuẩn lập mã video thuộc về nhóm của “các bộ mã hóa/giải mã video lai mất mát” (nghĩa là, kết hợp việc dự đoán theo không gian và theo thời gian trong miền mẫu và việc lập mã biến đổi 2D để áp dụng việc lượng tử hóa trong miền biến đổi). Mỗi ảnh của chuỗi video thường được phân chia thành tập hợp của các khối không chồng lấn và việc

lập mã thường được thực hiện trên mức khối. Nói cách khác, ở bộ mã hóa video thường được xử lý, nghĩa là, được mã hóa, trên mức khối (khối video), ví dụ, nhờ sử dụng việc dự đoán theo không gian (ảnh bên trong) và/hoặc việc dự đoán theo thời gian (ảnh liên kết) để tạo ra khối dự đoán, trừ khối dự đoán khỏi khối hiện thời (khối hiện thời được xử lý/cân được xử lý) để thu nhận khối phần dư, biến đổi khối phần dư và lượng tử hóa khối phần dư trong miền biến đổi để làm giảm lượng dữ liệu được truyền (nén), trong khi ở bộ giải mã việc xử lý ngược so với bộ mã hóa được áp dụng cho khối được mã hóa hoặc được nén để tái cấu trúc khối hiện thời nhằm biểu diễn. Hơn nữa, bộ mã hóa nhân đôi vòng lặp xử lý bộ giải mã sao cho cả hai sẽ tạo ra các việc dự đoán giống nhau (ví dụ, các việc nội hoặc liên dự đoán) và/hoặc các việc tái cấu trúc để xử lý, nghĩa là, lập mã, các khối tiếp theo.

Trong phần sau đây các phương án của hệ thống lập mã video 10, bộ mã hóa video 20 và bộ giải mã video 30 được mô tả dựa vào Fig.1 đến Fig.3.

Fig.1A là sơ đồ khối giản lược minh họa hệ thống lập mã 10 ví dụ, hệ thống lập mã video 10 (hoặc gọi tắt là hệ thống lập mã 10) mà có thể ứng 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 lập mã video 10 biểu diễn các ví dụ về các thiết bị mà có thể được tạo cấu hình để thực hiện các kỹ thuật phù hợp với các ví dụ khác nhau được mô tả theo sáng chế.

Như được thể hiện trên Fig.1A, hệ thống lập mã 10 bao gồm thiết bị nguồn 12 được tạo cấu hình để cung cấp dữ liệu ảnh được mã hóa 21 ví dụ, tới 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 bổ sung, nghĩa là, một cách tùy chọn, bao gồm nguồn ảnh 16, bộ xử lý trước (hoặc bộ phận xử lý trước) 18, ví dụ, bộ xử lý trước ảnh 18, và giao diện truyền thông hoặc bộ phận truyền thông 22.

Nguồn ảnh 16 có thể bao gồm hoặc là loại thiết bị chụp ảnh bất kỳ, ví dụ camera để chụp ảnh thế giới thực, và/hoặc loại thiết bị tạo ảnh bất kỳ, ví dụ bộ xử lý đồ họa máy tính để tạo ra ảnh hoạt hình máy tính, hoặc loại thiết bị khác bất kỳ để thu nhận và/hoặc cung cấp ảnh thế giới thực, ảnh được tạo ra bởi máy tính (ví dụ, nội dung màn hình, ảnh thực tế ảo (virtual reality, viết tắt là VR)) và/hoặc sự kết hợp bất kỳ của chúng (ví dụ, ảnh thực tế tăng cường (augmented reality, viết tắt là AR)). Nguồn ảnh có thể là loại bộ nhớ hoặc thiết bị lưu trữ bất kỳ lưu trữ bất kỳ trong số các ảnh nêu trên.

Phân biệt với bộ xử lý trước 18 và quy trình xử lý được thực hiện bởi bộ phận xử lý trước 18, ảnh hoặc dữ liệu ảnh 17 cũng có thể được gọi là ảnh thô hoặc dữ liệu ảnh thô 17.

Bộ xử lý trước 18 được tạo cấu hình để thu dữ liệu ảnh (thô) 17 và để thực hiện việc xử lý trước trên dữ liệu ảnh 17 để thu nhận ảnh được xử lý trước 19 hoặc dữ liệu ảnh được xử lý trước 19. Việc xử lý trước được thực hiện bởi bộ xử lý trước 18 có thể, ví dụ, bao gồm việc sắp xếp lại, chuyển đổi định dạng màu (ví dụ, từ RGB đến YCbCr), điều chỉnh màu sắc, hoặc khử nhiễu. Có thể hiểu rằng bộ phận xử lý trước 18 có thể là bộ phận tùy chọn.

Bộ mã hóa video 20 được tạo cấu hình để thu dữ liệu ảnh được xử lý trước 19 và cung cấp dữ liệu ảnh được mã hóa 21 (chi tiết thêm sẽ được mô tả dưới đây, ví dụ, dựa vào Fig.2).

Giao diện truyền thông 22 của thiết bị nguồn 12 có thể được tạo cấu hình để thu dữ liệu ảnh được mã hóa 21 và để truyền dữ liệu ảnh được mã hóa 21 (hoặc phiên bản được xử lý thêm bất kỳ của chúng) qua kênh truyền thông 13 tới thiết bị khác, ví dụ, thiết bị đích 14 hoặc thiết bị khác bất kỳ, cho việc lưu trữ và tái cấu trúc trực tiếp.

Thiết bị đích 14 bao gồm bộ giải mã 30 (ví dụ, bộ giải mã video 30), và có thể bổ sung, nghĩa là, một cách tùy chọn, bao gồm giao diện truyền thông hoặc bộ phận truyền thông 28, bộ xử lý sau 32 (hoặc bộ phận xử lý sau 32) và thiết bị hiển thị 34.

Giao diện truyền thông 28 của thiết bị đích 14 được tạo cấu hình thu dữ liệu ảnh được mã hóa 21 (hoặc phiên bản được xử lý thêm bất kỳ của chúng), ví dụ, trực tiếp từ thiết bị nguồn 12 hoặc từ nguồn khác bất kỳ, ví dụ, thiết bị lưu trữ, ví dụ, thiết bị lưu trữ dữ liệu ảnh được mã hóa, và cung cấp dữ liệu ảnh được mã hóa 21 đến bộ giải mã 30.

Giao diện truyền thông 22 và giao diện truyền thông 28 có thể được tạo cấu hình để truyền hoặc thu 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, ví dụ, sự kết nối nối dây hoặc không dây trực tiếp, hoặc qua loại mạng bất kỳ, ví dụ, mạng nối dây hoặc không dây hoặc sự kết hợp bất kỳ của chúng, hoặc loại mạng riêng hoặc công cộng bất kỳ, hoặc loại kết hợp bất kỳ của chúng.

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

Giao diện truyền thông 28, tạo nên bản sao của giao diện truyền thông 22, có thể, ví dụ, được tạo cấu hình để thu dữ liệu được truyền và xử lý dữ liệu truyền nhờ sử dụng loại giải mã hoặc xử lý và/hoặc mở gói truyền tương ứng bất kỳ để thu nhận dữ liệu ảnh được mã hóa 21.

Cả giao diện truyền thông 22 và giao diện truyền thông 28 đều có thể được tạo cấu hình là các giao diện truyền thông một chiều như được chỉ báo bởi mũi tên cho kênh truyền thông 13 trên Fig.1A trở từ thiết bị nguồn 12 đến thiết bị đích 14, hoặc các giao diện truyền thông hai chiều, và có thể được tạo cấu hình, ví dụ, để gửi và thu các tin nhắn, ví dụ, để thiết lập sự kết nối, để báo nhận và trao đổi thông tin khác bất kỳ liên quan đến liên kết truyền thông và/hoặc việc truyền dữ liệu, ví dụ, việc truyền dữ liệu ảnh được mã hóa.

Bộ giải mã 30 được tạo cấu hình để thu dữ liệu ảnh được mã hóa 21 và cung cấp dữ liệu ảnh được giải mã 31 hoặc ảnh được giải mã 31 (chi tiết thêm sẽ được mô tả dưới đây, ví dụ, dựa vào Fig.3 hoặc Fig.5).

Bộ xử lý sau 32 của thiết bị đích 14 được tạo cấu hình để xử lý sau dữ liệu ảnh được giải mã 31 (cũng được gọi là dữ liệu ảnh được tái cấu trúc), ví dụ, ảnh được giải mã 31, để thu nhận dữ liệu ảnh được xử lý sau 33, ví dụ, ảnh được xử lý sau 33. Quy trình xử lý sau được thực hiện bởi bộ phận xử lý sau 32 có thể bao gồm, ví dụ, chuyển đổi định dạng màu (ví dụ, từ YCbCr đến RGB), điều chỉnh màu sắc, sắp xếp lại, hoặc lấy mẫu lại, hoặc quy trình xử lý khác bất kỳ, ví dụ, để chuẩn bị dữ liệu ảnh được giải mã 31 nhằm hiển thị, ví dụ, bởi thiết bị hiển thị 34.

Thiết bị hiển thị 34 của thiết bị đích 14 được tạo cấu hình để thu dữ liệu ảnh được xử lý sau 33 để hiển thị ảnh, ví dụ, tới người dùng hoặc người xem. Thiết bị hiển thị 34 có thể là hoặc bao gồm loại màn hình bất kỳ để biểu diễn ảnh được tái cấu trúc, ví dụ, màn hình hoặc màn hiển thị bên trong hoặc được tích hợp. Các màn hình có thể, ví dụ, bao gồm các màn hình tinh thể lỏng (liquid crystal display, viết tắt là LCD), các màn hình điốt phát sáng hữu cơ (organic light emitting diodes, viết tắt là OLED), các màn hình plasma, các máy chiếu, các màn hình LED nhỏ, tinh thể lỏng trên silicon (liquid crystal on silicon, viết tắt là LCoS), bộ xử lý ánh sáng kỹ thuật số (digital light processor, viết tắt là DLP) hoặc loại màn hình khác bất kỳ.

Mặc dù Fig.1A mô tả thiết bị nguồn 12 và thiết bị đích 14 là các thiết bị riêng biệt, nhưng các phương án về các thiết bị cũng có thể bao gồm cả hai hoặc cả hai tính năng, thiết bị nguồn 12 hoặc tính năng tương ứng và thiết bị đích 14 hoặc tính năng tương ứng.

Theo các phương án như vậy thiết bị nguồn 12 hoặc tính năng tương ứng và thiết bị đích 14 hoặc tính năng tương ứng có thể được thực hiện nhờ sử dụng cùng phần cứng và/hoặc phần mềm hoặc bởi phần cứng và/hoặc phần mềm riêng biệt hoặc sự kết hợp bất kỳ của chúng.

Như sẽ rõ ràng với người có hiểu biết dựa vào phần mô tả, sự tồn tại và sự phân tách (chính xác) của các tính năng của các bộ phận khác nhau hoặc các tính năng nằm 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 tế.

Bộ mã hóa **20** (ví dụ, bộ mã hóa video 20) hoặc bộ giải mã **30** (ví dụ, bộ giải mã video 30) hoặc cả bộ mã hóa 20 và bộ giải mã 30 có thể được thực hiện qua hệ mạch xử lý như được thể hiện trên Fig.1B, chẳng hạn như một hoặc nhiều bộ vi xử lý, các bộ xử lý tín hiệu kỹ thuật số (digital signal processor, viết tắt là DSP), các mạch tích hợp chuyên dụng (application-specific integrated circuit, viết tắt là ASIC), các mảng cổng lập trình được dạng trường (field-programmable gate array, viết tắt là FPGA), logic rời rạc, phần cứng, hoặc các sự kết hợp bất kỳ của chúng. Bộ mã hóa 20 có thể được thực hiện qua hệ mạch xử lý 46 để bao gồm các môđun khác nhau như được thảo luận dựa vào bộ mã hóa 20 trên Fig.2 và/hoặc hệ thống hoặc hệ thống con bộ mã hóa khác bất kỳ được mô tả ở đây. Bộ giải mã 30 có thể được thực hiện qua hệ mạch xử lý 46 để bao gồm các môđun khác nhau như được thảo luận dựa vào bộ giải mã 30 trên Fig.3 và/hoặc hệ thống hoặc hệ thống con bộ giải mã khác bất kỳ được mô tả ở đây. Hệ mạch xử lý có thể được tạo cấu hình để thực hiện các thao tác khác nhau như được thảo luận dưới đây. Như được thể hiện trên Fig.5, nếu các kỹ thuật được thực hiện một phần ở phần mềm, thì thiết bị có thể lưu trữ các lệnh cho phần mềm trong phương tiện ghi đọc được bởi máy tính không tạm thời, thích hợp và có thể thực thi các lệnh ở phần cứng nhờ sử dụng một hoặc nhiều bộ xử lý để thực hiện các kỹ thuật của sáng chế. Một trong số bộ mã hóa video **20** và bộ giải mã video **30** có thể được tích hợp như phần của bộ mã hóa/bộ giải mã (CODEC) được kết hợp trong thiết bị đơn, ví dụ, như được thể hiện trên Fig.1B.

Thiết bị nguồn 12 và thiết bị đích 14 có thể bao gồm bất kỳ trong số khoảng rộng của các thiết bị, bao gồm loại bất kỳ của các thiết bị cầm tay hoặc văn phòng, ví dụ, các máy tính cầm tay hoặc xách tay, các điện thoại di động, các điện thoại thông minh, các máy tính bảng (tablet hoặc tablet computer), các camera, các máy tính để bàn, các thiết bị giải mã tín hiệu (set-top box), các máy thu hình, các thiết bị hiển thị, các trình phát phương tiện kỹ

thuật số, các máy chơi trò chơi video, các thiết bị tạo dòng video (chẳng hạn như các máy chủ dịch vụ nội dung hoặc các máy chủ phân phối nội dung), thiết bị thu phát sóng, thiết bị truyền phát sóng, hoặc tương tự và có thể sử dụng không hoặc 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 cho việc truyền thông không dây. Vì 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 lập 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 sự thiết đặt lập mã video (ví dụ, mã hóa video hoặc giải mã video) mà không cần bao gồm việc truyền thông dữ liệu bất kỳ giữa các thiết bị mã hóa và giải mã. Theo các ví dụ khác, dữ liệu được truy xuất từ bộ nhớ cục bộ, được tạo dòng qua mạng, hoặc tương tự. Thiết bị mã hóa video có thể mã hóa và lưu trữ dữ liệu tới bộ nhớ, và/hoặc thiết bị giải mã video có thể truy xuất và giải mã dữ liệu từ bộ nhớ. Theo một số ví dụ, việc mã hóa và giải mã được thực hiện bởi các thiết bị mà không truyền thông với thiết bị khác, nhưng mã hóa một cách đơn giản dữ liệu tới bộ nhớ và/hoặc truy xuất và giải mã dữ liệu từ bộ nhớ.

Nhằm thuận tiện cho việc mô tả, các phương án của sáng chế được mô tả ở đây, ví dụ, bằng cách tham chiếu tới việc lập mã video hiệu quả cao (High-Efficiency Video Coding, viết tắt là HEVC) hoặc phần mềm tham chiếu của việc lập mã video đa năng (Versatile Video Coding, viết tắt là VVC), chuẩn lập mã video thế hệ tiếp theo được phát triển bởi nhóm hợp tác lập mã video liên kết (Joint Collaboration Team on Video Coding, viết tắt là JCT-VC) của nhóm các chuyên gia lập mã video (Video Coding Experts Group, viết tắt là VCEG) ITU-T và nhóm các chuyên gia ảnh động (Motion Picture Experts Group, viết tắt là MPEG) ISO/IEC. Người có hiểu biết trung bình trong lĩnh vực kỹ thuật tương ứng sẽ hiểu rằng các phương án của sáng chế không giới hạn ở HEVC hoặc VVC.

Bộ mã hóa và phương pháp mã hóa

Fig.2 thể hiện sơ đồ khối giản lược của bộ mã hóa video 20 ví dụ mà được tạo cấu hình để thực hiện các kỹ thuật theo sáng chế. Theo ví dụ trên Fig.2, bộ mã hóa video 20 bao gồm đầu vào 201 (hoặc giao diện đầu vào 201), bộ phận tính toán phân dư 204, bộ phận xử lý biến đổi 206, bộ phận lượng tử hóa 208, bộ phận lượng tử hóa ngược 210, và bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 214, bộ phận bộ lọc vòng lặp 220, bộ đệm ảnh được giải mã (decoded picture buffer, viết tắt là DPB) 230, bộ phận lựa chọn chế độ 260, bộ phận mã hóa entropi 270 và đầu ra 272 (hoặc giao diện đầu ra 272). Bộ phận lựa

chọn chế độ 260 có thể bao gồm bộ phận liên dự đoán 244, bộ phận nội dự đoán 254 và bộ phận phân chia 262. Bộ phận liên dự đoán 244 có thể bao gồm bộ phận đánh giá chuyển động và bộ phận bù chuyển động (không được thể hiện). 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 bộ mã hóa/giải mã video lai.

Bộ phận tính toán phần dư 204, bộ phận xử lý biến đổi 206, bộ phận lượng tử hóa 208, bộ phận lựa chọn chế độ 260 có thể được đề cập đến là tạo nên đường tín hiệu xuôi của bộ mã hóa 20, trong khi bộ phận lượng tử hóa ngược 210, bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 214, bộ đệm 216, bộ lọc vòng lặp 220, bộ đệm ảnh được giải mã (DPB) 230, bộ phận liên dự đoán 244 và bộ phận nội dự đoán 254 có thể được đề cập đến là tạo nên đường tín hiệu ngược của bộ mã hóa video 20, trong đó đường tín hiệu xuôi 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). Bộ phận lượng tử hóa ngược 210, bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 214, bộ lọc vòng lặp 220, bộ đệm ảnh được giải mã (DPB) 230, bộ phận liên dự đoán 244 và bộ phận nội dự đoán 254 cũng được đề cập đến là tạo nên “bộ giải mã tích hợp” của bộ mã hóa video 20.

Các ảnh & việc phân chia ảnh (các ảnh & các khối)

Bộ mã hóa 20 có thể được tạo cấu hình để thu, ví dụ, qua đầu vào 201, ảnh 17 (hoặc dữ liệu ảnh 17), ví dụ, ảnh của chuỗi các ảnh tạo nên video hoặc chuỗi video. Ảnh hoặc dữ liệu ảnh thu được cũng có thể là ảnh được xử lý trước 19 (hoặc dữ liệu ảnh được xử lý trước 19). Nhằm đơn giản hóa phần mô tả sau đây đề cập đến ảnh 17. Ảnh 17 cũng có thể được gọi là ảnh hiện thời hoặc ảnh được lập mã (cụ thể trong việc lập mã video để phân biệt ảnh hiện thời với các ảnh khác, ví dụ, các ảnh được mã hóa và/hoặc được giải mã trước đó của cùng chuỗi video, nghĩa là, chuỗi video mà cũng bao gồm ảnh hiện thời).

Ảnh (kỹ thuật số) là hoặc có thể được coi là mảng hoặc ma trận hai chiều của các mẫu với các trị số cường độ. Mẫu trong mảng cũng có thể được gọi là điểm ảnh (dạng ngắn của thành phần ảnh) hoặc điểm ảnh (pel). Số các mẫu theo hướng chiều ngang và chiều dọc (hoặc trục) của mảng hoặc ảnh định rõ kích thước và/hoặc độ phân giải của ảnh. Nhằm biểu diễn màu sắc, thường ba thành phần màu sắc được sử dụng, nghĩa là, ảnh có thể được biểu diễn hoặc bao gồm ba mảng mẫu. Theo định dạng RGB hoặc không gian màu ảnh bao gồm mảng mẫu đỏ, xanh lá cây và xanh lam tương ứng. Tuy nhiên, trong lập mã video mỗi điểm ảnh thường được biểu diễn theo định dạng độ chói và sắc độ hoặc không gian màu,

ví dụ, YCbCr, mà bao gồm thành phần độ chói được chỉ báo bởi Y (thỉnh thoảng L cũng được sử dụng thay thế) và hai thành phần sắc độ được chỉ báo bởi Cb và Cr. Thành phần độ chói (hoặc viết tắt là luma) Y biểu diễn độ sáng hoặc cường độ mức xám (ví dụ, như trong ảnh thước xám), trong khi hai thành phần sắc độ (hoặc viết tắt là chroma) Cb và Cr biểu diễn sắc độ hoặc các thành phần thông tin màu sắc. Theo đó, ảnh theo định dạng YCbCr bao gồm mảng mẫu độ chói của các trị số mẫu độ chói (Y), và hai mảng mẫu sắc độ của các trị số sắc độ (Cb và Cr). Các ảnh theo định dạng RGB có thể được chuyển đổi hoặc được biến đổi thành định dạng YCbCr và ngược lại, quy trình xử lý cũng được biết đến là biến đổi hoặc chuyển đổi màu sắc. Nếu ảnh là đơn sắc, thì ảnh có thể bao gồm chỉ mảng mẫu độ chói. Theo đó, ảnh có thể là, ví dụ, mảng của các mẫu luma theo định dạng đơn sắc hoặc mảng của các mẫu luma và hai mảng tương ứng của các mẫu chroma theo định dạng màu 4:2:0, 4:2:2, và 4:4:4.

Các phương án về bộ mã hóa video 20 có thể bao gồm bộ phận phân chia ảnh (không được mô tả trên Fig.2) được tạo cấu hình để phân chia ảnh 17 thành các khối ảnh 203 (thường không chồng lấn). Các khối này cũng có thể được gọi là các khối ban đầu, các khối lớn (H.264/AVC) hoặc các khối cây lập mã (coding tree block, viết tắt là CTB) hoặc các bộ phận cây lập mã (coding tree unit, viết tắt là CTU) (H.265/HEVC và VVC). Bộ phận phân chia ảnh có thể được tạo cấu hình để sử dụng cùng kích thước khối cho tất cả các ảnh của chuỗi video và mạng lưới tương ứng định rõ kích thước khối, hoặc để thay đổi kích thước khối giữa các ảnh hoặc các tập hợp con hoặc các nhóm của các ảnh, và phân chia mỗi ảnh thành các khối tương ứng.

Theo các phương án khác, bộ mã hóa video có thể được tạo cấu hình để thu trực tiếp khối 203 của ảnh 17, ví dụ, một, nhiều hoặc tất cả các khối tạo nên ảnh 17. Khối ảnh 203 cũng có thể được gọi là khối ảnh hiện thời hoặc khối ảnh được lập mã.

Giống như ảnh 17, khối ảnh 203 lại là hoặc có thể được coi là mảng hoặc ma trận hai chiều của các mẫu với các trị số cường độ (các trị số mẫu), mặc dù kích thước nhỏ hơn ảnh 17. Nói cách khác, khối 203 có thể bao gồm, ví dụ, một mảng mẫu (ví dụ, mảng luma trong trường hợp ảnh đơn sắc 17, hoặc mảng luma hoặc chroma trong trường hợp ảnh màu) hoặc ba mảng mẫu (ví dụ, luma và hai mảng chroma trong trường hợp ảnh màu 17) hoặc số lượng và/hoặc loại khác bất kỳ của các mảng tùy thuộc vào định dạng màu được áp dụng. Số các mẫu theo hướng chiều ngang và chiều dọc (hoặc trục) của khối 203 định rõ kích

thước của khối 203. Theo đó, khối có thể là, ví dụ, mảng $M \times N$ (cột M nhân hàng N) của các mẫu, hoặc mảng $M \times N$ của các hệ số biến đổi.

Các phương án về 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, ví dụ, việc mã hóa và dự đoán được thực hiện trên mỗi khối 203.

Các phương án về bộ mã hóa video 20 như được thể hiện trên Fig.2 còn có thể được tạo cấu hình để phân chia và/hoặc mã hóa ảnh nhờ sử dụng các lát (cũng được đề cập đến là các lát video), trong đó ảnh có thể được phân chia thành hoặc được mã hóa nhờ sử dụng một hoặc nhiều lát (thường không chồng lấn), và mỗi lát có thể bao gồm một hoặc nhiều khối (ví dụ, các CTU) hoặc một hoặc nhiều nhóm của các khối (ví dụ, các miếng (H.265/HEVC và VVC) hoặc các mảnh (VVC)).

Các phương án về bộ mã hóa video 20 như được thể hiện trên Fig.2 còn có thể được tạo cấu hình để phân chia và/hoặc mã hóa ảnh nhờ sử dụng các lát/các nhóm miếng (cũng được đề cập đến là các nhóm miếng video) và/hoặc các miếng (cũng được đề cập đến là các miếng video), trong đó ảnh có thể được phân chia thành hoặc được mã hóa nhờ sử dụng một hoặc nhiều lát/nhiều nhóm miếng (thường không chồng lấn), và mỗi nhóm miếng/lát có thể bao gồm, ví dụ, một hoặc nhiều khối (ví dụ, các CTU) hoặc một hoặc nhiều miếng, trong đó mỗi miếng, ví dụ, có thể có dạng hình chữ nhật và có thể bao gồm một hoặc nhiều khối (ví dụ, các CTU), ví dụ, các khối phân số hoặc hoàn chỉnh.

Tính toán phần dư

Bộ phận tính toán phần dư 204 có thể được tạo cấu hình để tính khối phần dư 205 (cũng được đề cập đến là phần dư 205) dựa vào khối ảnh 203 và khối dự đoán 265 (chi tiết thêm về khối dự đoán 265 được đưa ra dưới đây), ví dụ, bằng cách trừ các trị số mẫu của khối dự đoán 265 khỏi các trị số mẫu của khối ảnh 203, theo từng mẫu (từng điểm ảnh) để thu nhận khối phần dư 205 trong miền mẫu.

Biến đổi

Bộ phận xử lý biến đổi 206 có thể được tạo cấu hình để áp dụng việc biến đổi, ví dụ, biến đổi côsin rời rạc (discrete cosine transform, viết tắt là DCT) hoặc biến đổi sin rời rạc (discrete sine transform, viết tắt là DST), trên các trị số mẫu của khối phần dư 205 để thu nhận 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 phần dư biến đổi và biểu diễn khối phần dư 205 trong miền biến đổi.

Bộ phận 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 như các phép biến đổi được quy định cho H.265/HEVC. So với việc biến đổi DCT trực giao, các phép xấp xỉ số nguyên như vậy thường được định tỷ lệ bởi hệ số nhất định. Để giữ lại định mức của khối phần dư mà được xử lý bởi các sự biến đổi xuôi và ngược, các hệ số định tỷ lệ bổ sung được áp dụng như một phần của quy trình biến đổi. Các hệ số định tỷ lệ thường được chọn dựa vào các sự ràng buộc nhất định như các hệ số định tỷ lệ là lũy thừa của hai đối với các phép toán dịch chuyển, độ sâu bit của các hệ số biến đổi, sự cân bằng giữa độ chính xác và các chi phí thực hiện, v.v. Các hệ số định tỷ lệ cụ thể, ví dụ, được quy định đối với việc biến đổi ngược, ví dụ, bởi bộ phận xử lý biến đổi ngược 212 (và việc biến đổi ngược tương ứng, ví dụ, bởi bộ phận xử lý biến đổi ngược 312 ở bộ giải mã video 30) và các hệ số định tỷ lệ tương ứng đối với việc biến đổi xuôi, ví dụ, bởi bộ phận xử lý biến đổi 206, ở bộ mã hóa 20 có thể được quy định theo đó.

Các phương án về bộ mã hóa video 20 (bộ phận xử lý biến đổi 206 tương ứng) có thể được tạo cấu hình để đưa ra các thông số biến đổi, ví dụ, loại biến đổi hoặc các phép biến đổi, ví dụ, trực tiếp hoặc được mã hóa hoặc được nén qua bộ phận mã hóa entropi 270, sao cho, ví dụ, bộ giải mã video 30 có thể thu và sử dụng các thông số biến đổi để giải mã.

Lượng tử hóa

Bộ phận lượng tử hóa 208 có thể được tạo cấu hình để lượng tử hóa các hệ số biến đổi 207 để thu nhận các hệ số được lượng tử hóa 209, ví dụ, bằng cách áp dụng việc lượng tử hóa vô hướng hoặc việc 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.

Quy trình lượng tử hóa có thể làm giảm độ sâu bit được kết hợp với một số hoặc tất cả các hệ số biến đổi 207. Ví dụ, hệ số biến đổi n bit có thể được làm tròn xuống tới hệ số biến đổi m bit trong suốt thời gian lượng tử hóa, trong đó n lớn hơn m . Mức lượng tử hóa có thể được điều chỉnh bằng cách điều chỉnh thông số lượng tử hóa (quantization parameter, viết tắt là QP). Ví dụ đối với việc lượng tử hóa vô hướng, việc định tỷ lệ khác có thể được áp dụng để đạt được việc lượng tử hóa mịn hơn hoặc thô hơn. Các kích thước bước lượng tử hóa nhỏ hơn tương ứng với việc lượng tử hóa mịn hơn, trong khi các kích thước bước lượng tử hóa lớn hơn tương ứng với việc lượng tử hóa thô hơn. Kích thước bước lượng tử hóa có thể áp dụng được có thể được chỉ báo bởi thông số lượng tử hóa (QP). Thông số

lượng tử hóa có thể ví dụ là chỉ số tới tập hợp được định rõ trước của các kích thước bước lượng tử hóa có thể áp dụng được. Ví dụ, các thông số lượng tử hóa nhỏ có thể tương ứng với việc lượng tử hóa mịn (các kích thước bước lượng tử hóa nhỏ) và các thông số lượng tử hóa lớn có thể tương ứng với việc lượng tử hóa thô (các kích thước bước lượng tử hóa lớn) hoặc ngược lại. Lượng tử hóa có thể bao gồm phép chia cho kích thước bước lượng tử hóa và tương ứng và/hoặc việc giải lượng tử hóa ngược, ví dụ, bởi bộ phận lượng tử hóa ngược 210, có thể bao gồm phép nhân với kích thước bước lượng tử hóa. Các phương án theo một số chuẩn, ví dụ, HEVC, có thể được tạo cấu hình để sử dụng thông số lượng tử hóa để xác định kích thước bước lượng tử hóa. Nói chung, kích thước bước lượng tử hóa có thể được tính toán dựa vào thông số lượng tử hóa nhờ sử dụng phép xấp xỉ điểm cố định của phương trình bao gồm phép chia. Các hệ số định tỷ lệ bổ sung có thể được giới thiệu cho việc lượng tử hóa và giải lượng tử hóa để khôi phục tiêu chuẩn của khối dư, mà có thể được điều chỉnh bởi vì việc định tỷ lệ được sử dụng trong phép xấp xỉ điểm cố định của phương trình đối với kích thước bước lượng tử hóa và thông số lượng tử hóa. Theo một cách thực hiện ví dụ, việc định tỷ lệ của việc biến đổi ngược và giải 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 tới bộ giải mã, ví dụ, ở dòng bit. Lượng tử hóa là phép toán mất mát, trong đó sự mất mát tăng lên với các kích thước bước lượng tử hóa tăng.

Các phương án về bộ mã hóa video 20 (bộ phận lượng tử hóa 208 tương ứng) có thể được tạo cấu hình để đưa ra các thông số lượng tử hóa (QP), ví dụ, trực tiếp hoặc được mã hóa qua bộ phận mã hóa entropi 270, sao cho, ví dụ, bộ giải mã video 30 có thể thu và áp dụng các thông số lượng tử hóa để giải mã.

Lượng tử hóa ngược

Bộ phận lượng tử hóa ngược 210 được tạo cấu hình để áp dụng việc lượng tử hóa ngược của bộ phận lượng tử hóa 208 trên các hệ số được lượng tử hóa để thu nhận các hệ số được giải lượng tử hóa 211, ví dụ, bằng cách áp dụng ngược sơ đồ lượng tử hóa được áp dụng bởi bộ phận lượng tử hóa 208 dựa vào hoặc nhờ sử dụng kích thước bước lượng tử hóa giống như bộ phận lượng tử hóa 208. Các hệ số được giải lượng tử hóa 211 cũng có thể được gọi là các hệ số dư được giải lượng tử hóa 211 và tương ứng - mặc dù thường không giống với các hệ số biến đổi do sự mất mát bởi việc lượng tử hóa – với các hệ số biến đổi 207.

Biến đổi ngược

Bộ phận xử lý biến đổi ngược 212 được tạo cấu hình để áp dụng việc biến đổi ngược của việc biến đổi được áp dụng bởi bộ phận xử lý biến đổi 206, ví dụ, biến đổi cosin rời rạc (DCT) ngược hoặc biến đổi sin rời rạc (DST) ngược hoặc các phép biến đổi ngược khác, để thu nhận khối phần dư được tái cấu trúc 213 (hoặc các hệ số được giải lượng tử hóa 213 tương ứng) trong miền mẫu. Khối phần dư được tái cấu trúc 213 cũng có thể được gọi là khối biến đổi 213.

Tái cấu trúc

Bộ phận tái cấu trúc 214 (ví dụ, bộ bổ sung hoặc bộ cộng 214) được tạo cấu hình để bổ sung khối biến đổi 213 (nghĩa là, khối phần dư được tái cấu trúc 213) vào khối dự đoán 265 để thu nhận khối được tái cấu trúc 215 trong miền mẫu, ví dụ, bằng cách bổ sung – theo từng mẫu - các trị số mẫu của khối phần dư được tái cấu trúc 213 và các trị số mẫu của khối dự đoán 265.

Lọc

Bộ phận bộ lọc vòng lặp 220 (hoặc viết tắt là “bộ lọc vòng lặp” 220), được tạo cấu hình để lọc khối được tái cấu trúc 215 để thu nhận khối được lọc 221, hoặc nói chung, để lọc các mẫu được tái cấu trúc để thu nhận các trị số mẫu được lọc. Bộ phận bộ lọc vòng lặp, ví dụ, được tạo cấu hình để làm mịn các sự chuyển tiếp điểm ảnh, hoặc theo cách khác nâng cao chất lượng video. Bộ phận bộ lọc vòng lặp 220 có thể bao gồm một hoặc nhiều bộ lọc vòng lặp chẳng hạn như bộ lọc tách khối, bộ lọc dịch vị thích nghi mẫu (sample-adaptive offset, viết tắt là SAO) hoặc một hoặc nhiều bộ lọc khác, ví dụ, bộ lọc vòng lặp thích nghi (adaptive loop filter, viết tắt là ALF), bộ lọc triệt nhiễu (noise suppression filter, viết tắt là NSF), hoặc sự kết hợp bất kỳ của chúng. Theo ví dụ, bộ phận bộ lọc vòng lặp 220 có thể bao gồm bộ lọc tách khối, bộ lọc SAO và bộ lọc ALF. Thứ tự của quy trình lọc có thể là bộ lọc tách khối, SAO và ALF. Theo ví dụ khác, quy trình xử lý được gọi là ánh xạ luma với định tỷ lệ chroma (luma mapping with chroma scaling, viết tắt là LMCS) (cụ thể là, bộ tạo lại hình dạng trong vòng lặp thích nghi) được bổ sung. Quy trình xử lý này được thực hiện trước khi tách khối. Theo ví dụ khác, quy trình lọc tách khối cũng có thể được áp dụng cho các mép khối con bên trong, ví dụ, các mép khối con afin, các mép khối con ATMP, các mép biến đổi khối con (sub-block transform, viết tắt là SBT) và các mép phân chia con bên trong (intra sub-partition, viết tắt là ISP). Mặc dù bộ phận bộ lọc vòng lặp 220 được thể hiện trên Fig.2 là bộ lọc trong vòng lặp, theo các cấu hình khác, bộ phận bộ lọc vòng lặp

220 có thể được thực hiện như bộ lọc sau vòng lặp. Khối được lọc 221 cũng có thể được gọi là khối được tái cấu trúc được lọc 221.

Các phương án về bộ mã hóa video 20 (bộ phận bộ lọc vòng lặp 220 tương ứng) có thể được tạo cấu hình để đưa ra các thông số bộ lọc vòng lặp (chẳng hạn như các thông số bộ lọc SAO hoặc các thông số bộ lọc ALF hoặc các thông số LMCS), ví dụ, trực tiếp hoặc được mã hóa qua bộ phận mã hóa entropi 270, sao cho, ví dụ, bộ giải mã 30 có thể thu và áp dụng cùng các thông số bộ lọc vòng lặp hoặc các bộ lọc vòng lặp tương ứng để giải mã.

Bộ đệm ảnh được giải mã

Bộ đệm ảnh được giải mã (DPB) 230 có thể là bộ nhớ mà lưu trữ các ảnh tham chiếu, hoặc dữ liệu ảnh tham chiếu nói chung, để mã hóa dữ liệu video bởi bộ mã hóa video 20. DPB 230 có thể được tạo nên bởi bất kỳ trong số các thiết bị nhớ, chẳng hạn như bộ nhớ truy cập ngẫu nhiên động (dynamic random access memory, viết tắt là DRAM), bao gồm DRAM đồng bộ (synchronous DRAM, viết tắt là SDRAM), RAM điện từ (magnetoresistive RAM, viết tắt là MRAM), RAM điện trở (resistive RAM, viết tắt là RRAM), hoặc các loại khác của các thiết bị nhớ. Bộ đệm ảnh được giải mã (DPB) 230 có thể được tạo cấu hình để lưu trữ một hoặc nhiều khối được lọc 221. Bộ đệm ảnh được giải mã 230 còn có thể được tạo cấu hình để lưu trữ các khối được lọc trước đó khác, ví dụ, các khối được lọc và được tái cấu trúc trước đó 221, của cùng ảnh hiện thời hoặc của các ảnh khác nhau, ví dụ, các ảnh được tái cấu trúc trước đó, và có thể cung cấp các ảnh được tái cấu trúc, nghĩa là, được giải mã trước đó hoàn toàn (và các khối và các mẫu tham chiếu tương ứng) và/hoặc một phần ảnh hiện thời được tái cấu trúc (và các khối và các mẫu tham chiếu tương ứng), ví dụ đối với việc liên dự đoán. Bộ đệm ảnh được giải mã (DPB) 230 cũng có thể được tạo cấu hình để lưu trữ một hoặc nhiều khối được tái cấu trúc không được lọc 215, hoặc các mẫu được tái cấu trúc không được lọc nói chung, ví dụ, nếu khối được tái cấu trúc 215 không được lọc bởi bộ phận bộ lọc vòng lặp 220, hoặc phiên bản được xử lý thêm khác bất kỳ của các khối hoặc các mẫu được tái cấu trúc.

Lựa chọn chế độ (phân chia & dự đoán)

Bộ phận lựa chọn chế độ 260 bao gồm bộ phận phân chia 262, bộ phận liên dự đoán 244 và bộ phận nội dự đoán 254, và được tạo cấu hình để thu hoặc thu nhận dữ liệu ảnh ban đầu, ví dụ, khối ban đầu 203 (khối hiện thời 203 của ảnh hiện thời 17), và dữ liệu ảnh được tái cấu trúc, ví dụ, các mẫu hoặc các khối được tái cấu trúc được lọc và/hoặc không được lọc của cùng ảnh (hiện thời) và/hoặc từ một hoặc nhiều ảnh được giải mã trước đó,

ví dụ, từ bộ đệm ảnh được giải mã 230 hoặc các bộ đệm khác (ví dụ, bộ đệm dòng, không được thể hiện). Dữ liệu ảnh được tái cấu trúc được sử dụng như dữ liệu ảnh tham chiếu cho việc dự đoán, ví dụ, liên dự đoán hoặc nội dự đoán, để thu nhận khối dự đoán 265 hoặc bộ dự đoán 265.

Bộ phận lựa chọn chế độ 260 có thể được tạo cấu hình để xác định hoặc lựa chọn việc phân chia cho chế độ dự đoán khối hiện thời (bao gồm việc không phân chia) và chế độ dự đoán (ví dụ, chế độ nội hoặc liên dự đoán) và tạo ra khối dự đoán 265 tương ứng, mà được sử dụng cho việc tính toán của khối phần dư 205 và cho việc tái cấu trúc của khối được tái cấu trúc 215.

Các phương án về bộ phận lựa chọn chế độ 260 có thể được tạo cấu hình để lựa chọn việc phân chia và chế độ dự đoán (ví dụ, từ việc phân chia và chế độ dự đoán được hỗ trợ bởi hoặc khả dụng đối với bộ phận lựa chọn chế độ 260), mà đưa ra sự trùng khớp nhất hoặc nói cách khác phần dư nhỏ nhất (phần dư nhỏ nhất nghĩa là việc nén tốt hơn đối với việc truyền hoặc lưu trữ), hoặc phí tổn báo hiệu nhỏ nhất (phí tổn báo hiệu nhỏ nhất nghĩa là việc nén tốt hơn đối với việc truyền hoặc lưu trữ), hoặc xem xét hoặc cân bằng cả hai. Bộ phận lựa chọn chế độ 260 có thể được tạo cấu hình để xác định việc phân chia và chế độ dự đoán dựa vào sự tối ưu hóa tốc độ biến dạng (rate distortion optimization, viết tắt là RDO), nghĩa là, lựa chọn chế độ dự đoán mà đưa ra tốc độ biến dạng nhỏ nhất. Các thuật ngữ như “tốt nhất”, “nhỏ nhất”, “tối ưu” v.v. theo ngữ cảnh này không cần thiết đề cập đến tổng thể “tốt nhất”, “nhỏ nhất”, “tối ưu”, v.v. mà cũng có thể đề cập đến việc hoàn tất của tiêu chuẩn chấm dứt hoặc lựa chọn như trị số vượt quá hoặc giảm dưới ngưỡng hoặc các sự giới hạn khác có khả năng dẫn tới “sự lựa chọn dưới điều kiện tốt nhất” nhưng làm giảm độ phức tạp và thời gian xử lý.

Nói cách khác, bộ phận phân chia 262 có thể được tạo cấu hình để phân chia khối 203 thành các phần chia khối nhỏ hơn hoặc các khối con (mà tạo nên lại các khối), ví dụ, nhờ sử dụng lặp lại việc phân chia cây tứ phân (quad-tree, viết tắt là QT), việc phân chia cây nhị phân (binary-tree, viết tắt là BT), việc phân chia cây tam phân (triple-tree, viết tắt là TT) hoặc sự kết hợp bất kỳ của chúng, và để thực hiện, ví dụ, việc dự đoán cho mỗi trong số các phần chia khối hoặc các khối con, trong đó việc lựa chọn chế độ bao gồm việc lựa chọn cấu trúc cây của khối được phân chia 203 và các chế độ dự đoán được áp dụng cho mỗi trong số các phần chia khối hoặc các khối con.

Trong phần sau đây việc phân chia (ví dụ, bởi bộ phận phân chia 260) và việc xử lý dự đoán (bởi bộ phận liên dự đoán 244 và bộ phận nội dự đoán 254) được thực hiện bởi bộ mã hóa video 20 ví dụ sẽ được giải thích chi tiết hơn.

Phân chia

Bộ phận phân chia 262 có thể được tạo cấu hình để phân chia ảnh từ chuỗi video thành chuỗi của các bộ phận cây lập mã (CTU), và bộ phận phân chia 262 có thể phân chia (hoặc phân tách) bộ phận cây lập mã (CTU) 203 thành các phần chia nhỏ hơn, ví dụ, các khối nhỏ hơn có kích thước hình vuông hoặc hình chữ nhật. Đối với ảnh mà có ba mảng mẫu, CTU bao gồm khối $N \times N$ của các mẫu luma cùng với hai khối tương ứng của các mẫu chroma. Kích thước được phép lớn nhất của khối luma trong CTU được quy định là 128×128 trong việc phát triển lập mã video đa năng (VVC), nhưng nó có thể được quy định là trị số khác với 128×128 trong tương lai, ví dụ, 256×256 . Các CTU của ảnh có thể được gộp lại/được nhóm như các lát/các nhóm miếng, các miếng hoặc các mảnh. Miếng bao gồm khu vực hình chữ nhật của ảnh, và miếng có thể được chia thành một hoặc nhiều mảnh. Mảnh bao gồm một số hàng CTU nằm trong miếng. Miếng mà không được phân chia thành nhiều mảnh có thể được gọi là mảnh. Tuy nhiên, mảnh là tập hợp con đúng của miếng và không được gọi là miếng. Có hai chế độ của các nhóm miếng được hỗ trợ trong VVC, cụ thể là chế độ nhóm miếng/lát quét mảnh và chế độ lát hình chữ nhật. Ở chế độ nhóm miếng quét mảnh, nhóm miếng/lát chứa chuỗi của các miếng trong việc quét mảnh miếng của ảnh. Ở chế độ lát hình chữ nhật, lát chứa một số mảnh của ảnh mà tạo nên chung khu vực hình chữ nhật của ảnh. Các mảnh nằm trong lát hình chữ nhật theo thứ tự của việc quét mảnh mảnh của lát. Các khối nhỏ hơn này (mà cũng có thể được gọi là các khối con) còn có thể được phân chia thành thậm chí các phần chia nhỏ hơn. Điều này cũng được đề cập đến là việc phân chia cây hoặc việc phân chia cây phân cấp, trong đó khối ban đầu, ví dụ, ở mức cây gốc 0 (mức phân cấp 0, độ sâu 0), có thể được phân chia đệ quy, ví dụ, được phân chia thành hai hoặc nhiều khối của mức thấp hơn tiếp theo, ví dụ, các nút ở mức cây 1 (mức phân cấp 1, độ sâu 1), trong đó các khối này có thể được phân chia lại thành hai hoặc nhiều khối của mức thấp hơn tiếp theo, ví dụ, mức cây 2 (mức phân cấp 2, độ sâu 2), ví dụ, cho đến khi việc phân chia được chấm dứt, ví dụ, do tiêu chí chấm dứt được hoàn tất, ví dụ, độ sâu cây lớn nhất hoặc kích thước khối nhỏ nhất đạt được. Các khối mà không còn được phân chia cũng được đề cập đến là các khối lá hoặc các nút lá của cây. Cây sử dụng việc phân chia thành hai phần chia được gọi là cây nhị phân (BT), cây sử dụng việc

phân chia thành ba phần chia được gọi là cây tam phân (TT), và cây sử dụng việc phân chia thành bốn phần chia được gọi là cây tứ phân (QT).

Ví dụ, bộ phận cây lập mã (CTU) có thể là hoặc bao gồm CTB của các mẫu luma, hai CTB tương ứng của các mẫu chroma của ảnh mà có ba mảng mẫu, hoặc CTB của các mẫu của ảnh đơn sắc hoặc ảnh mà được lập mã nhờ sử dụng ba mặt phẳng màu sắc riêng biệt và các cấu trúc cú pháp được sử dụng để lập mã các mẫu. Một cách tương ứng, khối cây lập mã (CTB) có thể là khối $N \times N$ của các mẫu cho một số trị số của N sao cho việc chia thành phần thành các CTB là việc phân chia. Bộ phận lập mã (coding unit, viết tắt là CU) có thể là hoặc bao gồm khối lập mã của các mẫu luma, hai khối lập mã tương ứng của các mẫu chroma của ảnh mà có ba mảng mẫu, hoặc khối lập mã của các mẫu của ảnh đơn sắc hoặc ảnh mà được lập mã nhờ sử dụng ba mặt phẳng màu sắc riêng biệt và các cấu trúc cú pháp được sử dụng để lập mã các mẫu. Một cách tương ứng, khối lập mã (CB) có thể là khối $M \times N$ của các mẫu cho một số trị số của M và N sao cho việc chia CTB thành các khối lập mã là việc phân chia.

Theo các phương án, ví dụ, theo HEVC, bộ phận cây lập mã (CTU) có thể được phân tách thành các CU nhờ sử dụng cấu trúc cây tứ phân được biểu thị là cây lập mã. Quyết định xem việc lập mã vùng ảnh nhờ sử dụng việc dự đoán ảnh liên kết (theo thời gian) hay ảnh bên trong (theo không gian) được thực hiện ở mức CU lá hay không. Mỗi CU lá còn có thể được phân tách thành một, hai hoặc bốn PU theo loại phân tách PU. Bên trong một PU, cùng quy trình dự đoán được áp dụng và thông tin thích hợp được truyền tới bộ giải mã trên cơ sở PU. Sau khi thu nhận khối phần dư bằng cách áp dụng quy trình dự đoán dựa vào loại phân tách PU, CU lá có thể được phân chia thành các bộ phận biến đổi (transform unit, viết tắt là TU) theo cấu trúc cây tứ phân khác tương tự cây lập mã cho CU.

Theo các phương án, ví dụ, theo chuẩn lập mã video mới nhất hiện nay đang phát triển, mà được gọi là lập mã video đa năng (VVC), cây đa loại được lồng ghép cây tứ phân được kết hợp nhờ sử dụng cấu trúc phân đoạn phân tách nhị phân và tam phân ví dụ được sử dụng để phân chia bộ phận cây lập mã. Trong cấu trúc cây lập mã nằm trong bộ phận cây lập mã, CU có thể có hoặc dạng hình vuông hoặc hình chữ nhật. Ví dụ, bộ phận cây lập mã (CTU) trước tiên được phân chia bởi cây bậc bốn. Sau đó các nút lá cây bậc bốn còn có thể được phân chia bởi cấu trúc cây đa loại. Có bốn loại phân tách trong cấu trúc cây đa loại, phân tách nhị phân theo chiều dọc (SPLIT_BT_VER), phân tách nhị phân theo chiều ngang (SPLIT_BT_HOR), phân tách tam phân theo chiều dọc (SPLIT_TT_VER), và phân tách

tam phân theo chiều ngang (SPLIT_TT_HOR). Các nút lá cây đa loại được gọi là các bộ phận lập mã (CU), và trừ khi CU quá lớn đối với độ dài biến đổi lớn nhất, phân đoạn này được sử dụng cho việc dự đoán và xử lý biến đổi mà không phân chia thêm bất kỳ nữa. Điều này nghĩa là, trong hầu hết các trường hợp, CU, PU và TU có cùng kích thước khối trong cấu trúc khối lập mã cây đa loại được lồng ghép với cây tứ phân. Ngoại trừ xảy ra khi độ dài biến đổi được hỗ trợ lớn nhất nhỏ hơn độ rộng hoặc độ cao của thành phần màu sắc của CU. VVC phát triển cơ chế báo hiệu duy nhất của thông tin phân tách phân chia trong cấu trúc cây lập mã cây đa loại được lồng ghép với cây tứ phân. Theo cơ chế báo hiệu, bộ phận cây lập mã (CTU) được coi là gốc của cây bậc bốn và trước tiên được phân chia bởi cấu trúc cây bậc bốn. Mỗi nút lá cây bậc bốn (khi đủ lớn để cho phép nó) sau đó còn được phân chia bởi cấu trúc cây đa loại. Trong cấu trúc cây đa loại, cờ thứ nhất (`mtt_split_cu_flag`) được báo hiệu để chỉ báo xem nút còn được phân chia hay không; khi nút còn được phân chia, cờ thứ hai (`mtt_split_cu_vertical_flag`) được báo hiệu để chỉ báo hướng phân tách, và sau đó cờ thứ ba (`mtt_split_cu_binary_flag`) được báo hiệu để chỉ báo xem việc phân tách là phân tách nhị phân hay phân tách tam phân. Dựa vào các trị số của `mtt_split_cu_vertical_flag` và `mtt_split_cu_binary_flag`, chế độ phân tách cây đa loại (`MttSplitMode`) của CU có thể được dẫn ra bởi bộ giải mã dựa vào quy tắc được định rõ trước hoặc bảng. Cần lưu ý, đối với thiết kế nhất định, ví dụ, khối luma 64×64 và thiết kế tạo ống dẫn chroma 32×32 trong các bộ giải mã phần cứng VVC, việc phân tách TT bị cấm khi hoặc độ rộng hoặc độ cao của khối lập mã luma lớn hơn 64, như được thể hiện trên Fig.6. Việc phân tách TT cũng bị cấm khi hoặc độ rộng hoặc độ cao của khối lập mã chroma lớn hơn 32. Thiết kế tạo ống dẫn sẽ chia ảnh thành các bộ phận dữ liệu tạo ống dẫn ảo (Virtual pipelining data unit, viết tắt là VPDU) mà được định rõ là các bộ phận không chồng lấn trong ảnh. Ở các bộ giải mã phần cứng, các VPDU liên tiếp được xử lý đồng thời bởi nhiều giai đoạn tạo ống dẫn. Kích thước VPDU tỷ lệ thuận với kích thước bộ đệm trong hầu hết các giai đoạn tạo ống dẫn, vì vậy điều quan trọng là giữ kích thước VPDU nhỏ. Trong hầu hết các bộ giải mã phần cứng, kích thước VPDU có thể được thiết đặt tới kích thước khối biến đổi (transform block, viết tắt là TB) lớn nhất. Tuy nhiên, trong VVC, việc phân chia cây tam phân (TT) và cây nhị phân (BT) có thể dẫn đến việc làm tăng kích thước các VPDU.

Ngoài ra, cần lưu ý rằng, khi phần của khối nút cây vượt quá ranh giới ảnh dưới cùng hoặc bên phải, khối nút cây bị buộc được phân tách cho đến khi tất cả các mẫu của mọi CU được lập mã được bố trí bên trong các ranh giới ảnh.

Với ví dụ, công cụ các phần chia con bên trong (Intra Sub-Partition, viết tắt là ISP) có thể chia các khối được nội dự đoán luma theo chiều dọc hoặc theo chiều ngang thành 2 hoặc 4 phần chia con tùy thuộc vào kích thước khối.

Theo một ví dụ, bộ phận lựa chọn chế độ 260 của bộ mã hóa video 20 có thể được tạo cấu hình để thực hiện sự kết hợp bất kỳ của các kỹ thuật phân chia được mô tả ở đây.

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

Nội dự đoán

Tập hợp của các chế độ nội dự đoán có thể bao gồm 25 chế độ nội dự đoán khác nhau, ví dụ, các chế độ vô hướng như chế độ DC (hoặc trung bình) và chế độ hai chiều, hoặc các chế độ có hướng, ví dụ, như được định rõ trong HEVC, hoặc có thể bao gồm 67 chế độ nội dự đoán khác nhau, ví dụ, các chế độ vô hướng như chế độ DC (hoặc trung bình) và chế độ hai chiều, hoặc các chế độ có hướng, ví dụ, như được định rõ cho VVC. Với ví dụ, nhiều chế độ nội dự đoán góc thông thường được thay thế một cách thích ứng với các chế độ nội dự đoán góc rộng cho các khối không vuông, ví dụ, như được định rõ trong VVC. Với ví dụ khác, để tránh khỏi các thao tác phân chia cho việc dự đoán DC, chỉ phía dài hơn được sử dụng để tính giá trị trung bình cho các khối không vuông. Và, các kết quả của việc nội dự đoán của chế độ hai chiều còn có thể được điều chỉnh bởi phương pháp kết hợp nội dự đoán phụ thuộc vị trí (position dependent intra prediction combination, viết tắt là PDCPC).

Bộ phận nội dự đoán 254 được tạo cấu hình để sử dụng các mẫu được tái cấu trúc của các khối lân cận của cùng ảnh hiện thời để tạo ra khối dự đoán bên trong 265 theo chế độ nội dự đoán của tập hợp của các chế độ nội dự đoán.

Bộ phận nội dự đoán 254 (hoặc bộ phận lựa chọn chế độ 260 nói chung) còn được tạo cấu hình để đưa ra các thông số nội dự đoán (hoặc thông tin chỉ báo của chế độ nội dự đoán được lựa chọn cho khối nói chung) tới bộ phận mã hóa entropi 270 ở dạng của các thành phần cú pháp 266 cho việc bao gồm vào dữ liệu ảnh được mã hóa 21, sao cho, ví dụ, bộ giải mã video 30 có thể thu và sử dụng các thông số dự đoán để giải mã.

Liên dự đoán

Tập hợp của các chế độ liên dự đoán (hoặc khả thi) tùy thuộc vào các ảnh tham chiếu khả dụng (nghĩa là, các ảnh được giải mã ít nhất một phần trước đó, ví dụ, được lưu trữ trong DBP 230) và các thông số liên dự đoán khác, ví dụ, xem toàn bộ ảnh tham chiếu hay chỉ một phần, ví dụ, vùng cửa sổ nghiên cứu xung quanh vùng của khối hiện thời, của ảnh tham chiếu được sử dụng để tìm kiếm khối tham chiếu so khớp tốt nhất, và/hoặc ví dụ, xem việc nội suy điểm ảnh được áp dụng hay không, ví dụ, nội suy bán/nửa điểm ảnh, một phần tư điểm ảnh và/hoặc 1/16 điểm ảnh.

Ngoài các chế độ dự đoán nêu trên, chế độ bỏ qua, chế độ trực tiếp và/hoặc chế độ liên dự đoán khác có thể được áp dụng.

Ví dụ, việc dự đoán hợp nhất mở rộng, danh mục ứng viên hợp nhất của chế độ như vậy được cấu trúc bằng cách bao gồm năm loại ứng viên sau đây theo thứ tự: MVP theo không gian từ các CU lân cận theo không gian, MVP theo thời gian từ các CU được sắp xếp vào một chỗ, MVP dựa trên lịch sử từ bảng FIFO, MVP trung bình theo cặp và các MV không. Và việc tinh chỉnh vector chuyển động phía bộ giải mã dựa vào việc so khớp hai chiều (bilateral-matching based decoder side motion vector refinement, viết tắt là DMVR) có thể được áp dụng để làm tăng độ chính xác của các MV của chế độ hợp nhất. Chế độ hợp nhất với MVD (Merge mode with MVD, viết tắt là MMVD), mà đến từ chế độ hợp nhất với các độ chênh lệch vector chuyển động. Cờ MMVD được báo hiệu ngay sau khi gửi cờ bỏ qua và cờ hợp nhất để xác định xem chế độ MMVD được sử dụng cho CU hay không. Và sơ đồ độ phân giải vector chuyển động thích nghi (adaptive motion vector resolution, viết tắt là AMVR) mức CU có thể được áp dụng. AMVR cho phép MVD của CU được lập mã ở độ chính xác khác nhau. Tùy thuộc vào chế độ dự đoán đối với CU hiện thời, các MVD của CU hiện thời có thể được lựa chọn một cách thích ứng. Khi CU được lập mã ở chế độ hợp nhất, chế độ liên/nội dự đoán được kết hợp (combined inter/intra prediction, viết tắt là CIIP) có thể được áp dụng đối với CU hiện thời. Việc tính trung bình trọng số của các tín hiệu nội dự đoán và liên kết được thực hiện để thu nhận việc dự đoán CIIP. Việc dự đoán được bù chuyển động afin, trường chuyển động afin của khối được mô tả bởi thông tin chuyển động của hai điểm điều khiển (4 thông số) hoặc ba vector chuyển động điểm điều khiển (6 thông số). Việc dự đoán vector chuyển động theo thời gian dựa trên khối con (Subblock-based temporal motion vector prediction, viết tắt là STMVP), mà tương tự việc dự đoán vector chuyển động theo thời gian (temporal motion vector

prediction, viết tắt là TMVP) trong HEVC, nhưng dự đoán các vector chuyển động của các CU con nằm trong CU hiện thời. Luồng quang hai chiều (Bi-directional optical flow, viết tắt là BDOF), trước đó được gọi là BIO, là phiên bản đơn giản hơn mà yêu cầu việc tính toán ít hơn nhiều, cụ thể là xét về số các phép nhân và kích thước của bộ nhân. Chế độ phân chia tam giác, ở chế độ như vậy, CU được phân tách đều thành hai phần chia hình tam giác, nhờ sử dụng hoặc việc phân tách theo đường chéo hoặc việc phân tách không theo đường chéo. Ngoài ra, chế độ dự đoán đôi được mở rộng ngoài việc tính trung bình đơn giản để cho phép việc tính trung bình trọng số của hai tín hiệu dự đoán.

Bộ phận liên dự đoán 244 có thể bao gồm bộ phận đánh giá chuyển động (motion estimation, viết tắt là ME) và bộ phận bù chuyển động (motion compensation, viết tắt là MC) (cả hai không được thể hiện trên Fig.2). Bộ phận đánh giá chuyển động có thể được tạo cấu hình để thu hoặc thu nhận khối ảnh 203 (khối ảnh hiện thời 203 của ảnh hiện thời 17) và ảnh được giải mã 231, hoặc ít nhất một hoặc nhiều khối được tái cấu trúc trước đó, ví dụ, các khối được tái cấu trúc của một hoặc nhiều trong số các ảnh được giải mã 231 trước đó khác/khác nhau, cho việc đánh giá chuyển động. Ví dụ, chuỗi video có thể bao gồm ảnh hiện thời và các ảnh được giải mã 231 trước đó, hoặc nói cách khác, ảnh hiện thời và các ảnh được giải mã 231 trước đó có thể là phần của hoặc tạo nên chuỗi các ảnh tạo nên chuỗi video.

Bộ mã hóa 20 có thể, ví dụ, được tạo cấu hình để lựa chọn khối tham chiếu từ nhiều khối tham chiếu của cùng ảnh hoặc các ảnh khác nhau của nhiều ảnh khác và cung cấp ảnh tham chiếu (hoặc chỉ số ảnh tham chiếu) và/hoặc dịch vị (dịch vị theo không gian) giữa vị trí (các tọa độ x, y) của khối tham chiếu và vị trí của khối hiện thời như các thông số liên dự đoán tới bộ phận đánh giá chuyển động. Dịch vị này cũng được gọi là vector chuyển động (motion vector, viết tắt là MV).

Bộ phận bù chuyển động được tạo cấu hình để thu nhận, ví dụ, thu, thông số liên dự đoán và để thực hiện việc liên dự đoán dựa vào hoặc nhờ sử dụng thông số liên dự đoán để thu nhận khối liên dự đoán 265. Việc bù chuyển động, được thực hiện bởi bộ phận bù chuyển động, có thể bao gồm việc nạp hoặc tạo ra khối dự đoán dựa vào vector chuyển động/khối được xác định bởi việc đánh giá chuyển động, có thể thực hiện các phép nội suy tới độ chính xác điểm ảnh con. Việc lọc nội suy có thể tạo ra các mẫu điểm ảnh bổ sung từ các mẫu điểm ảnh đã biết, vì vậy có khả năng làm tăng số các khối dự đoán ứng viên mà có thể được sử dụng để lập mã khối ảnh. Khi thu vector chuyển động cho PU của khối ảnh

hiện thời, bộ phận bù chuyển động có thể bố trí khối dự đoán trong đó vector chuyển động trở tới trong một trong số các danh mục ảnh tham chiếu.

Bộ phận bù chuyển động cũng có thể tạo ra các thành phần cú pháp được kết hợp với các khối và các lát video để sử dụng bởi bộ giải mã video 30 trong việc giải mã các khối ảnh của lát video. Ngoài ra hoặc theo cách khác đối với các lát và các thành phần cú pháp tương ứng, các nhóm miếng và/hoặc các miếng và các thành phần cú pháp tương ứng có thể được tạo ra hoặc được sử dụng.

Lập mã entropi

Bộ phận mã hóa entropi 270 được tạo cấu hình để áp dụng, ví dụ, thuật toán hoặc sơ đồ mã hóa entropi (ví dụ, sơ đồ lập mã độ dài biến đổi (variable length coding, viết tắt là VLC), sơ đồ VLC thích nghi ngữ cảnh (context adaptive VLC, viết tắt là CAVLC), sơ đồ lập mã thuật toán, nhị phân hóa, lập mã số học nhị phân thích nghi ngữ cảnh (context adaptive binary arithmetic coding, viết tắt là CABAC), lập mã số học nhị phân thích nghi ngữ cảnh dựa trên cú pháp (syntax-based context-adaptive binary arithmetic coding, viết tắt là SBAC), lập mã entropi phân chia khoảng xác suất (probability interval partitioning entropy, viết tắt là PIPE) hoặc phương pháp hoặc kỹ thuật mã hóa entropi khác) hoặc tránh qua (không nén) trên các hệ số được lượng tử hóa 209, các thông số liên dự đoán, các thông số nội dự đoán, các thông số bộ lọc vòng lặp và/hoặc các thành phần cú pháp khác để thu nhận dữ liệu ảnh được mã hóa 21 mà có thể được đưa ra qua đầu ra 272, ví dụ, dưới dạng của dòng bit được mã hóa 21, sao cho, ví dụ, bộ giải mã video 30 có thể thu và sử dụng các thông số để giải mã. Dòng bit được mã hóa 21 có thể được truyền tới bộ giải mã video 30, hoặc được lưu trữ trong bộ nhớ dùng cho việc truyền hoặc truy xuất sau đó 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. Ví dụ, bộ mã hóa trên cơ sở không biến đổi 20 có thể lượng tử hóa trực tiếp tín hiệu phân dư mà không có bộ phận xử lý biến đổi 206 cho các khối hoặc các khung nhất định. Theo cách thực hiện khác, bộ mã hóa 20 có thể có bộ phận lượng tử hóa 208 và bộ phận lượng tử hóa ngược 210 được kết hợp thành bộ phận đơn.

Bộ giải mã và phương pháp giải mã

Fig.3 thể hiện ví dụ về bộ giải mã video 30 mà được tạo cấu hình để thực hiện các kỹ thuật của sáng chế. Bộ giải mã video 30 được tạo cấu hình để thu dữ liệu ảnh được mã hóa 21 (ví dụ, dòng bit được mã hóa 21), ví dụ, được mã hóa bởi bộ mã hóa 20, để thu nhận

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, ví dụ, dữ liệu mà biểu diễn các khối ảnh của lát video được mã hóa (và/hoặc các nhóm miếng hoặc các miếng) và các thành phần cú pháp được kết hợp.

Theo ví dụ trên Fig.3, bộ giải mã 30 bao gồm bộ phận giải mã entropi 304, bộ phận lượng tử hóa ngược 310, bộ phận xử lý biến đổi ngược 312, bộ phận tái cấu trúc 314 (ví dụ, bộ cộng 314), bộ lọc vòng lặp 320, bộ đệm ảnh được giải mã (DBP) 330, bộ phận ứng dụng chế độ 360, bộ phận liên dự đoán 344 và bộ phận nội dự đoán 354. Bộ phận liên dự đoán 344 có thể hoặc bao gồm bộ phận bù chuyển động. Bộ giải mã video 30 có thể, theo một số ví dụ, thực hiện quy trình giải mã nói chung ngược với quy trình mã hóa được mô tả dựa vào bộ mã hóa video 100 từ Fig.2.

Như được giải thích liên quan đến bộ mã hóa 20, bộ phận lượng tử hóa ngược 210, bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 214 bộ lọc vòng lặp 220, bộ đệm ảnh được giải mã (DPB) 230, bộ phận liên dự đoán 344 và bộ phận nội dự đoán 354 cũng được đề cập đến là tạo nên “bộ giải mã tích hợp” của bộ mã hóa video 20. Theo đó, bộ phận lượng tử hóa ngược 310 có thể giống nhau về chức năng với bộ phận lượng tử hóa ngược 110, bộ phận xử lý biến đổi ngược 312 có thể giống nhau về chức năng với bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 314 có thể giống nhau về chức năng với bộ phận tái cấu trúc 214, bộ lọc vòng lặp 320 có thể giống nhau về chức năng với bộ lọc vòng lặp 220, và bộ đệm ảnh được giải mã 330 có thể giống nhau về chức năng với bộ đệm ảnh được giải mã 230. Do đó, các phần giải thích được đưa ra đối với các bộ phận và các chức năng tương ứng của bộ mã hóa video 20 áp dụng một cách tương ứng cho các bộ phận và các chức năng tương ứng của bộ giải mã video 30.

Giải mã entropi

Bộ phận giải mã entropi 304 được tạo cấu hình để tách dòng bit 21 (hoặc dữ liệu ảnh được mã hóa 21 nói chung) và thực hiện, ví dụ, việc giải mã entropi tới dữ liệu ảnh được mã hóa 21 để thu nhận, ví dụ, các hệ số được lượng tử hóa 309 và/hoặc các thông số lập mã được giải mã (không được thể hiện trên Fig.3), ví dụ, bất kỳ hoặc tất cả các thông số liên dự đoán (ví dụ, chỉ số ảnh tham chiếu và vector chuyển động), thông số nội dự đoán (ví dụ, chế độ hoặc chỉ số nội dự đoán), các thông số biến đổi, các thông số lượng tử hóa, các thông số bộ lọc vòng lặp, và/hoặc các thành phần cú pháp khác. Bộ phận giải mã entropi 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 liên quan đến bộ phận mã hóa entropi 270 của bộ mã hóa 20. Bộ phận giải mã entropi 304 còn có thể được tạo cấu hình để cung cấp các thông số liên dự đoán, thông số nội dự đoán và/hoặc các thành phần cú pháp khác tới bộ phận ứng dụng chế độ 360 và các thông số khác tới các bộ phận khác của bộ giải mã 30. Bộ giải mã video 30 có thể thu các thành phần cú pháp ở mức lát video và/hoặc mức khối video. Ngoài ra hoặc theo cách khác đối với các lát và các thành phần cú pháp tương ứng, các nhóm miêng và/hoặc các miêng và các thành phần cú pháp tương ứng có thể được thu và/hoặc được sử dụng.

Lượng tử hóa ngược

Bộ phận lượng tử hóa ngược 310 có thể được tạo cấu hình để thu các thông số lượng tử hóa (QP) (hoặc thông tin liên quan đến việc lượng tử hóa ngược nói chung) và các hệ số được lượng tử hóa từ dữ liệu ảnh được mã hóa 21 (ví dụ, bằng cách tách và/hoặc giải mã, ví dụ, bởi bộ phận giải mã entropi 304) và để áp dụng dựa vào các thông số lượng tử hóa việc lượng tử hóa ngược trên các hệ số được lượng tử hóa được giải mã 309 để thu nhận các hệ số được giải lượng tử hóa 311, mà cũng có thể được gọi là các hệ số biến đổi 311. Quy trình lượng tử hóa ngược có thể bao gồm việc sử dụng thông số lượng tử hóa được xác định bởi bộ mã hóa video 20 cho mỗi khối video trong lát video (hoặc miêng hoặc nhóm miêng) để xác định mức lượng tử hóa và, tương tự, mức lượng tử hóa ngược mà sẽ được áp dụng.

Biến đổi ngược

Bộ phận xử lý biến đổi ngược 312 có thể được tạo cấu hình để thu các hệ số được giải lượng tử hóa 311, cũng được gọi là các hệ số biến đổi 311, và để áp dụng việc biến đổi cho các hệ số được giải lượng tử hóa 311 để thu nhận các khối phần dư được tái cấu trúc 213 trong miền mẫu. Các khối phần dư được tái cấu trúc 213 cũng có thể được gọi là các khối biến đổi 313. Việc biến đổi có thể là biến đổi ngược, ví dụ, DCT ngược, DST ngược, biến đổi số nguyên ngược, hoặc quy trình biến đổi ngược tương tự về mặt khái niệm. Bộ phận xử lý biến đổi ngược 312 còn có thể được tạo cấu hình để thu các thông số biến đổi hoặc thông tin tương ứng từ dữ liệu ảnh được mã hóa 21 (ví dụ, bằng cách tách và/hoặc giải mã, ví dụ, bởi bộ phận giải mã entropi 304) để xác định việc biến đổi được áp dụng cho các hệ số được giải lượng tử hóa 311.

Tái cấu trúc

Bộ phận tái cấu trúc 314 (ví dụ, bộ bổ sung hoặc bộ cộng 314) có thể được tạo cấu hình để bổ sung khối phần dư được tái cấu trúc 313, vào khối dự đoán 365 để thu nhận khối được tái cấu trúc 315 trong miền mẫu, ví dụ, bằng cách bổ sung các trị số mẫu của khối phần dư được tái cấu trúc 313 và các trị số mẫu của khối dự đoán 365.

Lọc

Bộ phận bộ lọc vòng lặp 320 (hoặc trong vòng lặp lặp mã hoặc sau vòng lặp lặp mã) được tạo cấu hình để lọc khối được tái cấu trúc 315 để thu nhận khối được lọc 321, ví dụ, để làm mịn các sự chuyển tiếp điểm ảnh, hoặc theo cách khác nâng cao chất lượng video. Bộ phận bộ lọc vòng lặp 320 có thể bao gồm một hoặc nhiều bộ lọc vòng lặp chẳng hạn như bộ lọc tách khối, bộ lọc dịch vị thích nghi mẫu (SAO) hoặc một hoặc nhiều bộ lọc khác, ví dụ, bộ lọc vòng lặp thích nghi (ALF), bộ lọc triệt nhiễu (NSF), hoặc sự kết hợp bất kỳ của chúng. Theo ví dụ, bộ phận bộ lọc vòng lặp 220 có thể bao gồm bộ lọc tách khối, bộ lọc SAO và bộ lọc ALF. Thứ tự của quy trình lọc có thể là bộ lọc tách khối, SAO và ALF. Theo ví dụ khác, quy trình xử lý được gọi là ánh xạ luma với định tỷ lệ chroma (LMCS) (cụ thể là, bộ tạo lại hình dạng trong vòng lặp thích nghi) được bổ sung. Quy trình xử lý này được thực hiện trước khi tách khối. Theo ví dụ khác, quy trình lọc tách khối cũng có thể được áp dụng cho các mép khối con bên trong, ví dụ, các mép khối con afin, các mép khối con ATMP, các mép biến đổi khối con (SBT) và các mép phân chia con bên trong (ISP). Mặc dù bộ phận bộ lọc vòng lặp 320 được thể hiện trên Fig.3 là bộ lọc trong vòng lặp, nhưng theo các cấu hình khác, bộ phận bộ lọc vòng lặp 320 có thể được thực hiện như bộ lọc sau vòng lặp.

Bộ đệm ảnh được giải mã

Các khối video được giải mã 321 của ảnh sau đó được lưu trữ trong bộ đệm ảnh được giải mã 330, mà lưu trữ các ảnh được giải mã 331 như các ảnh tham chiếu cho việc bù chuyển động tiếp theo cho các ảnh khác và/hoặc cho việc đưa ra màn hình tương ứng.

Bộ giải mã 30 được tạo cấu hình để đưa ra ảnh được giải mã 311, ví dụ, qua đầu ra 312, nhằm biểu diễn hoặc xem tới người dùng.

Dự đoán

Bộ phận liên dự đoán 344 có thể giống với bộ phận liên dự đoán 244 (cụ thể là bộ phận bù chuyển động) và bộ phận nội dự đoán 354 có thể giống với bộ phận liên dự đoán 254 về chức năng, và thực hiện các quyết định phân tách hoặc phân chia và dự đoán dựa vào các thông số dự đoán và/hoặc phân chia hoặc thông tin tương ứng thu được từ dữ liệu ảnh

được mã hóa 21 (ví dụ, bằng cách tách và/hoặc giải mã, ví dụ, bởi bộ phận giải mã entropi 304). Bộ phận ứng dụng chế độ 360 có thể được tạo cấu hình để thực hiện việc dự đoán (nội hoặc liên dự đoán) trên mỗi khối dựa vào các ảnh được tái cấu trúc, các khối hoặc các mẫu tương ứng (được lọc hoặc không được lọc) để thu nhận khối dự đoán 365.

Khi lát video được lập mã là lát (I) được lập mã bên trong, bộ phận nội dự đoán 354 của bộ phận ứng dụng chế độ 360 được tạo cấu hình để tạo ra khối dự đoán 365 cho khối ảnh của lát video hiện thời dựa vào chế độ nội dự đoán được báo hiệu và dữ liệu từ các khối được giải mã trước đó của ảnh hiện thời. Khi ảnh video được lập mã là lát (nghĩa là, B hoặc P) được lập mã liên kết, bộ phận liên dự đoán 344 (ví dụ, bộ phận bù chuyển động) của bộ phận ứng dụng chế độ 360 được tạo cấu hình để tạo ra các khối dự đoán 365 cho khối video của lát video hiện thời dựa vào các vector chuyển động và các thành phần cú pháp khác thu được từ bộ phận giải mã entropi 304. Đối với việc liên dự đoán, các khối dự đoán có thể được tạo ra từ một trong số các ảnh tham chiếu nằm trong một trong số các danh mục ảnh tham chiếu. Bộ giải mã video 30 có thể cấu trúc các danh mục khung tham chiếu, danh mục 0 và danh mục 1, nhờ sử dụng các kỹ thuật cấu trúc mặc định dựa vào các ảnh tham chiếu được lưu trữ trong DPB 330. Điều giống hoặc tương tự có thể được áp dụng cho hoặc bởi các phương án nhờ sử dụng các nhóm miếng (ví dụ, các nhóm miếng video) và/hoặc các miếng (ví dụ, các miếng video) ngoài ra hoặc theo cách khác cho các lát (ví dụ, các lát video), ví dụ, video có thể được lập mã nhờ sử dụng các nhóm miếng và/hoặc các miếng I, P hoặc B.

Bộ phận ứng dụng chế độ 360 được tạo cấu hình để xác định thông tin dự đoán cho khối video của lát video hiện thời bằng cách tách các vector chuyển động và các thành phần cú pháp khác, và sử dụng thông tin dự đoán để tạo ra các khối dự đoán cho khối video hiện thời được giải mã. Ví dụ, bộ phận ứng dụng chế độ 360 sử dụng một vài trong số các thành phần cú pháp thu được để xác định chế độ dự đoán (ví dụ, nội hoặc liên dự đoán) được sử dụng để lập mã các khối video của lát video, loại lát liên dự đoán (ví dụ, lát B, lát P, hoặc lát GPB), thông tin cấu trúc cho một hoặc nhiều trong số các danh mục ảnh tham chiếu cho lát, các vector chuyển động cho mỗi khối video được mã hóa liên kết của lát, trạng thái liên dự đoán cho mỗi khối video được lập mã liên kết của lát, và thông tin khác để giải mã các khối video trong lát video hiện thời. Điều giống hoặc tương tự có thể được áp dụng cho hoặc bởi các phương án nhờ sử dụng các nhóm miếng (ví dụ, các nhóm miếng video) và/hoặc các miếng (ví dụ, các miếng video) ngoài ra hoặc theo cách khác cho các lát (ví

dụ, các lát video), ví dụ, video có thể được lập mã nhờ sử dụng các nhóm miếng và/hoặc các miếng I, P hoặc B.

Các phương án về bộ giải mã video 30 như được thể hiện trên Fig.3 có thể được tạo cấu hình để phân chia và/hoặc giải mã ảnh nhờ sử dụng các lát (cũng được đề cập đến là các lát video), trong đó ảnh có thể được phân chia thành hoặc được giải mã nhờ sử dụng một hoặc nhiều lát (thường không chồng lấn), và mỗi lát có thể bao gồm một hoặc nhiều khối (ví dụ, các CTU) hoặc một hoặc nhiều nhóm của các khối (ví dụ, các miếng (H.265/HEVC và VVC) hoặc các mảnh (VVC)).

Các phương án về bộ giải mã video 30 như được thể hiện trên Fig.3 có thể được tạo cấu hình để phân chia và/hoặc giải mã ảnh nhờ sử dụng các lát/các nhóm miếng (cũng được đề cập đến là các nhóm miếng video) và/hoặc các miếng (cũng được đề cập đến là các miếng video), trong đó ảnh có thể được phân chia thành hoặc được giải mã nhờ sử dụng một hoặc nhiều lát/nhiều nhóm miếng (thường không chồng lấn), và mỗi nhóm miếng/lát có thể bao gồm, ví dụ, một hoặc nhiều khối (ví dụ, các CTU) hoặc một hoặc nhiều miếng, trong đó mỗi miếng, ví dụ, có thể có dạng hình chữ nhật và có thể bao gồm một hoặc nhiều khối (ví dụ, các CTU), ví dụ, các khối phân số hoặc hoàn chỉnh.

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. Ví dụ, bộ giải mã 30 có thể tạo ra dòng video đầu ra mà không có bộ phận bộ lọc vòng lặp 320. Ví dụ, bộ giải mã dựa trên cơ sở không biến đổi 30 có thể lượng tử hóa ngược trực tiếp tín hiệu phần dư mà không có bộ phận xử lý biến đổi ngược 312 cho các khối hoặc các khung nhất định. Theo cách thực hiện khác, bộ giải mã video 30 có thể có bộ phận lượng tử hóa ngược 310 và bộ phận xử lý biến đổi ngược 312 được kết hợp thành bộ phận đơn.

Cần hiểu rằng, trong bộ mã hóa 20 và bộ giải mã 30, kết quả xử lý của bước hiện tại còn có thể được xử lý và sau đó được đưa ra tới bước tiếp theo. Ví dụ, sau khi lọc nội suy, việc dẫn ra vector chuyển động hoặc lọc vòng lặp, phép toán khác, chẳng hạn như cắt bớt hoặc dịch chuyển, có thể được thực hiện trên kết quả xử lý của việc lọc nội suy, việc dẫn ra vector chuyển động hoặc lọc vòng lặp.

Cần lưu ý rằng các phép toán khác có thể được áp dụng cho các vector chuyển động được dẫn ra của khối hiện thời (bao gồm nhưng không giới hạn ở các vector chuyển động điểm điều khiển của chế độ afin, các vector chuyển động khối con ở các chế độ afin, hai chiều, ATMVP, các vector chuyển động theo thời gian, và v.v.). Ví dụ, trị số của vector

chuyển động bị ràng buộc ở khoảng được định rõ trước theo bit biểu diễn của nó. Nếu bit biểu diễn của vector chuyển động là bitDepth, thì sau đó khoảng là $-2^{(\text{bitDepth}-1)} \sim 2^{(\text{bitDepth}-1)}-1$, trong đó “^” nghĩa là lũy thừa. Ví dụ, nếu bitDepth được thiết đặt bằng 16, thì khoảng là -32768 ~ 32767; nếu bitDepth được thiết đặt bằng 18, thì khoảng là -131072~131071. Ví dụ, trị số của vector chuyển động được dẫn ra (ví dụ, các MV của bốn khối con 4x4 nằm trong một khối 8x8) bị ràng buộc sao cho độ chênh lệch lớn nhất giữa các phần số nguyên của bốn MV khối con 4x4 không nhiều hơn N điểm ảnh, chẳng hạn như không nhiều hơn 1 điểm ảnh. Ở đây đề xuất hai phương pháp ràng buộc vector chuyển động theo bitDepth.

Fig.4 là hình vẽ giản lược của thiết bị lập mã video 400 theo phương án của sáng chế. Thiết bị lập mã video 400 thích hợp cho việc thực hiện các phương án được bộc lộ như được mô tả ở đây. Theo phương án, thiết bị lập mã video 400 có thể là bộ giải mã chẳng hạn như bộ giải mã video 30 trên Fig.1A hoặc bộ mã hóa chẳng hạn như bộ mã hóa video 20 trên Fig.1A.

Thiết bị lập mã video 400 bao gồm các cổng vào 410 (hoặc các cổng đầu vào 410) và các bộ phận bộ thu (Rx) 420 để thu dữ liệu; bộ xử lý, bộ phận logic, hoặc bộ xử lý trung tâm (central processing unit, viết tắt là CPU) 430 để xử lý dữ liệu; các bộ phận bộ truyền (Tx) 440 và các cổng ra 450 (hoặc các cổng đầu ra 450) để truyền dữ liệu; và bộ nhớ 460 để lưu trữ dữ liệu. Thiết bị lập mã video 400 cũng có thể bao gồm các thành phần quang-điện (optical-to-electrical, viết tắt là OE) và các thành phần điện-quang (electrical-to-optical, viết tắt là EO) được ghép nối với các cổng vào 410, các bộ phận bộ thu 420, các bộ phận bộ truyền 440, và các cổng ra 450 dùng cho việc ra hoặc vào của các tín hiệu quang hoặc điện.

Bộ xử lý 430 được thực hiện bởi phần cứng và phần mềm. Bộ xử lý 430 có thể được thực hiện như một hoặc nhiều chip CPU, các lõi (ví dụ, như bộ xử lý nhiều lõi), các FPGA, các ASIC, và các DSP. Bộ xử lý 430 truyền thông với các cổng vào 410, các bộ phận bộ thu 420, các bộ phận bộ truyền 440, các cổng ra 450, và bộ nhớ 460. Bộ xử lý 430 bao gồm môđun lập mã 470. Môđun lập mã 470 thực hiện các phương án được bộc lộ nêu trên. Chẳng hạn, môđun lập mã 470 thực hiện, xử lý, chuẩn bị, hoặc đưa ra các thao tác lập mã khác nhau. Việc bao gồm môđun lập mã 470 do đó cung cấp sự nâng cao đáng kể tới tính năng của thiết bị lập mã video 400 và tác động việc biến đổi của thiết bị lập mã video 400

tới trạng thái khác. Theo cách khác, môđun lập mã 470 được thực hiện như các lệnh được lưu trữ trong bộ nhớ 460 và được thực hiện bởi bộ xử lý 430.

Bộ nhớ 460 có thể bao gồm một hoặc nhiều đĩa, nhiều ổ đĩa băng, và nhiều ổ đĩa trạng thái rắn và có thể được sử dụng như thiết bị lưu trữ dữ liệu tràn, để lưu trữ các chương trình khi các chương trình như vậy được lựa chọn cho việc thực thi, và để lưu trữ các lệnh và dữ liệu mà được đọc trong suốt thời gian thực thi chương trình. Bộ nhớ 460 có thể, ví dụ, khả biến và/hoặc bất khả biến và có thể là bộ nhớ chỉ đọc (read-only memory, viết tắt là ROM), bộ nhớ truy cập ngẫu nhiên (random access memory, viết tắt là RAM), bộ nhớ có thể định nội dung địa chỉ bậc ba (ternary content-addressable memory, viết tắt là TCAM), và/hoặc bộ nhớ truy cập ngẫu nhiên tĩnh (static random-access memory, viết tắt là SRAM).

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

Bộ xử lý 502 trong máy 500 có thể là bộ xử lý trung tâm. Theo cách khác, bộ xử lý 502 có thể là loại thiết bị khác bất kỳ, hoặc nhiều thiết bị, có khả năng điều khiển hoặc xử lý thông tin tồn tại hiện thời hoặc được phát triển sau này. Mặc dù cách thực hiện được bộc lộ có thể được thực hiện với bộ xử lý đơn như được thể hiện, ví dụ, bộ xử lý 502, nhưng các ưu điểm về tốc độ và hiệu quả có thể đạt được nhờ sử dụng nhiều hơn một bộ xử lý.

Bộ nhớ 504 trong máy 500 có thể là thiết bị bộ nhớ chỉ đọc (ROM) hoặc thiết bị bộ nhớ truy cập ngẫu nhiên (RAM) theo cách thực hiện. Loại thiết bị lưu trữ thích hợp khác bất kỳ có thể được sử dụng như bộ nhớ 504. Bộ nhớ 504 có thể bao gồm mã và dữ liệu 506 mà được truy cập bởi bộ xử lý 502 nhờ sử dụng bus 512. Bộ nhớ 504 còn có thể 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 mà cho phép bộ xử lý 502 thực hiện các phương pháp được mô tả ở đây. Ví dụ, các chương trình ứng dụng 510 có thể bao gồm ứng dụng 1 qua N, mà còn bao gồm ứng dụng lập mã video mà thực hiện các phương pháp được mô tả ở đây.

Máy 500 cũng có thể bao gồm một hoặc nhiều thiết bị đầu ra, chẳng hạn như màn hình 518. Màn hình 518 có thể là, theo một ví dụ, màn hình nhạy chạm mà kết hợp màn hình với thành phần nhạy chạm mà có thể thao tác được để cảm biến các đầu vào chạm. Màn hình 518 có thể được ghép nối với bộ xử lý 502 qua bus 512.

Mặc dù được mô tả ở đây là bus đơn, nhưng bus 512 của máy 500 có thể bao gồm nhiều bus. Hơn nữa, thiết bị lưu trữ thứ cấp 514 có thể được ghép nối trực tiếp với các bộ phận còn lại của máy 500 hoặc có thể được truy cập qua mạng và có thể bao gồm bộ phận

tích hợp đơn giản hạn như thẻ nhớ hoặc nhiều bộ phận chẳng hạn như nhiều thẻ nhớ. Máy 500 vì vậy có thể được thực hiện theo nhiều cấu hình khác nhau.

Dự đoán được bù chuyển động afin

Trong ITU-T H.265, chỉ mô hình chuyển động tịnh tiến được áp dụng cho việc dự đoán bù chuyển động (motion compensation prediction, viết tắt là MCP). Trong khi trong thế giới thực, có nhiều loại chuyển động, ví dụ, phóng to/thu nhỏ, quay, các chuyển động phối cảnh và các chuyển động bất thường khác. Trong VTM6, việc dự đoán bù chuyển động biến đổi afin trên cơ sở khối được áp dụng. Như được thể hiện trên Fig.6, trường chuyển động afin của khối được mô tả bởi thông tin chuyển động của hai điểm điều khiển (4 thông số) hoặc ba vectơ chuyển động điểm điều khiển (CPMV) (6 thông số).

Phương trình chung đối với vectơ chuyển động tính toán ở vị trí mẫu (x, y) là:

$$\begin{cases} mv_x = dHorX * x + dVerX * y + mv_{0x} \\ mv_y = dHorY * x + dVerY * y + mv_{0y} \end{cases} \quad (1-1)$$

Đối với mô hình chuyển động afin 4 thông số, vectơ chuyển động ở vị trí mẫu (x, y) được dẫn ra là:

$$\begin{cases} mv_x = \frac{mv_{1x}-mv_{0x}}{W}x + \frac{mv_{0y}-mv_{1y}}{W}y + mv_{0x} \\ mv_y = \frac{mv_{1y}-mv_{0y}}{W}x + \frac{mv_{1x}-mv_{0x}}{W}y + mv_{0y} \end{cases} \quad (1-2)$$

Đối với mô hình chuyển động afin 6 thông số, vectơ chuyển động ở vị trí mẫu (x, y) được dẫn ra là:

$$\begin{cases} mv_x = \frac{mv_{1x}-mv_{0x}}{W}x + \frac{mv_{2x}-mv_{0x}}{H}y + mv_{0x} \\ mv_y = \frac{mv_{1y}-mv_{0y}}{W}x + \frac{mv_{2y}-mv_{0y}}{H}y + mv_{0y} \end{cases} \quad (1-3)$$

Trong đó (mv_{0x}, mv_{0y}) là vectơ chuyển động của điểm điều khiển góc trên cùng bên trái, (mv_{1x}, mv_{1y}) là vectơ chuyển động của điểm điều khiển góc trên cùng bên phải, và (mv_{2x}, mv_{2y}) là vectơ chuyển động của điểm điều khiển góc dưới cùng bên trái.

Đối với trường hợp sử dụng mô hình chuyển động afin 6 thông số

$$dHorX = \frac{mv_{1x}-mv_{0x}}{W} \quad (1-4)$$

$$dHorY = \frac{mv_{1y}-mv_{0y}}{W}, \quad (1-5)$$

$$dVerX = \frac{mv_{2x}-mv_{0x}}{H}, \quad (1-6)$$

$$dVerY = \frac{mv_{2y}-mv_{0y}}{H} \quad (1-7)$$

Đối với trường hợp sử dụng mô hình chuyển động afin 4 thông số,

$$dHorX = \frac{mv_{1x} - mv_{0x}}{W}, \quad (1-8)$$

$$dHorY = \frac{mv_{1y} - mv_{0y}}{W}, \quad (1-9)$$

$$dVerX = -dHorY, \quad (1-10)$$

$$dVerY = dHorX. \quad (1-11)$$

Như được thực hiện đối với việc liên dự đoán chuyển động tịnh tiến, cũng có hai chế độ liên dự đoán chuyển động afin: chế độ hợp nhất afin và chế độ AMVP afin.

Dự đoán biến đổi afin trên cơ sở khối

Để đơn giản hóa việc dự đoán bù chuyển động, việc dự đoán biến đổi afin trên cơ sở khối được áp dụng. Để dẫn ra vector chuyển động của mỗi khối con luma 8×8 , vector chuyển động của mẫu trung tâm của mỗi khối con, như được thể hiện trên Fig 7, được tính theo các phương trình nêu trên, và được làm tròn tới độ chính xác phân số $1/16$. Sau đó các bộ lọc nội suy bù chuyển động được áp dụng, để tạo ra việc dự đoán của mỗi khối con với vector chuyển động được dẫn ra. Kích thước khối con của các thành phần chroma được thiết đặt là 4×4 .

Bộ lọc nội suy song tuyến tính nâng cao

Bộ lọc nội suy song tuyến tính nâng cao (Enhanced bi-linear Interpolation Filter, viết tắt là EIF) có thể được sử dụng cho khối dự đoán và trên cơ sở khối con. Thủ tục lọc giống nhau đối với các tín hiệu luma và các tín hiệu chroma, thủ tục lọc bao gồm của các bước sau đây:

1. Dẫn ra trường vector chuyển động dựa trên điểm ảnh từ các CPMV, theo phương trình (1-1);
2. Thu nhận các mẫu được nội suy dựa vào các vector chuyển động được dẫn ra, nhờ sử dụng nội suy song tuyến tính cho các dịch vị phân số;
3. Thực hiện việc lọc theo chiều ngang và sau đó theo chiều dọc nhờ sử dụng bộ lọc thông cao 3 nhánh cố định $[-1, 10, -1]$ với hệ số bình thường hóa là 8.

Hai bước thứ nhất được thực hiện đối với vùng $(w + 2) \times (h + 2)$, trong đó w và h lần lượt là độ rộng và độ cao khối dự đoán; một lẻ điểm ảnh từ mỗi biên được bổ sung để áp dụng bộ lọc 3 nhánh ở bước thứ ba. Khối afin ban đầu và khối $(w + 2) \times (h + 2)$ tương ứng mà được sử dụng ở bước trung gian của EIF được mô tả trên Fig.8.

Tính toán độ rộng dải bộ nhớ

Độ rộng dải bộ nhớ được tính là tỷ lệ vùng khối tham chiếu với khối hiện thời. Ví dụ, đối với khối dự đoán đôi 8×8 trong trường hợp bộ lọc nội suy sử dụng với T nhánh, trị số

vùng tham chiếu là S_r bằng $2(8+T-1)(8+T-1)$, vùng khối S_b bằng $8*8$. Vì vậy độ rộng dải bộ nhớ là $MB_{8x8} = \frac{2(8+T-1)(8+T-1)}{64}$. Đối với DCTIF 8 nhánh mà được sử dụng trong ITU-T H.265, VVC và EVC, $MB_{8x8} = \frac{2*15*15}{64} = 7,03125$.

Vùng tham chiếu là vùng hình chữ nhật bao gồm tất cả các mẫu tham chiếu mà được cần để thực hiện việc bù chuyển động của khối hiện thời, như được mô tả trên Fig.9, có độ rộng W' và độ cao H' (hộp giới hạn).

Sử dụng thích nghi EIF và bù chuyển động afin trên cơ sở khối. Định nghĩa vấn đề.

Việc bù chuyển động khối con afin với kích thước khối con nhỏ nhất $8x8$ thân thiện với phần cứng hơn nhiều so với việc bù chuyển động afin với kích thước khối con nhỏ nhất $4x4$. Có ít nhất ba lý do cho điều đó.

1. Độ rộng dải bộ nhớ. Việc bù chuyển động afin với kích thước khối con nhỏ nhất $8x8$ không làm tăng độ rộng dải bộ nhớ so với ITU-T H.265, do các khối dự đoán đôi $8x8$ là trường hợp kém nhất đối với ITU-T H.265 xét về việc tính toán độ rộng dải bộ nhớ. Trong EVC các khối dự đoán đôi $8x8$ cũng không thay đổi trường hợp kém nhất xét về độ rộng dải bộ nhớ (các khối dự đoán đôi $8x4/4x8$ là trường hợp kém nhất trong EVC 3.0 và các khối $4x16/16x4$ sẽ là trường hợp kém nhất trong EVC 4.0). Về cơ bản các khối dự đoán đôi $8x8$ có thể xảy ra trong việc liên dự đoán thông thường cả trong EVC và ITU-T H.265, vì vậy việc bù chuyển động khối con afin với kích thước khối nhỏ nhất như vậy không làm tăng độ phức tạp của việc bù chuyển động.

2. Số các phép nhân. Việc bù chuyển động đối với khối con $8x8$ yêu cầu số các phép nhân ít hơn nhiều so với việc bù chuyển động của bốn khối con $4x4$.

3. Truy cập bộ nhớ. Theo một số cách thực hiện phần cứng không nhỏ hơn 16 mẫu có thể được đọc. Từ các khối $8x8$ phối cảnh này mà trong trường hợp sử dụng DCTIF 8 nhánh lấy các mẫu tham chiếu $(8+8-1)*(8+8-1)$ ứng dụng bộ nhớ hữu hiệu hơn nhiều so với các khối $4x4$.

Tuy nhiên việc bù chuyển động khối con afin với kích thước khối con nhỏ nhất $8x8$ đưa ra nhiều sự suy giảm hiệu suất đáng kể so với bù chuyển động khối con afin với kích thước khối con nhỏ nhất $4x4$. Cụ thể là đối với nội dung với việc quay nhanh. Đối với nội dung như vậy EIF có thể được sử dụng. EIF có ít phép nhân hơn so với việc bù chuyển động khối con afin với các kích thước khối con nhỏ nhất $8x8$ và $4x4$. Tuy nhiên độ rộng dải bộ nhớ đối với EIF mà không có các giới hạn mô hình chuyển động afin có thể là lớn.

Ngoài ra đối với các cách thực hiện phần cứng hữu hiệu của EIF một số yêu cầu bổ sung có thể xuất hiện. Ví dụ, các yêu cầu sau đây có thể xuất hiện đối với EIF xét về phần cứng.

A. Bộ đệm bên trong được giới hạn bởi các dòng N , trong đó N có thể ví dụ là 3, 4, 5 hoặc lớn hơn. Điều này nghĩa là trong suốt thời gian xử lý của một dòng (một hàng) của khối (khối con) hiện thời không nhiều hơn các dòng N từ ảnh tham chiếu có thể được sử dụng.

B. Truy cập bộ nhớ sẽ là tuần tự, mà nghĩa là nếu đối với dòng thứ i của khối hiện thời dòng thứ j của ảnh tham chiếu được nạp, sau đó đối với dòng thứ $(i+1)$ của khối hiện thời chỉ các dòng $j+1, j+2, \dots$ có thể được nạp.

C. Không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất.

Sáng chế là sơ đồ thích nghi sử dụng EIF và việc bù chuyển động khối con afin.

Phương án 1

Sử dụng thích nghi EIF và bù chuyển động afin trên cơ sở khối. Thuật toán cơ bản.

Thuật toán cơ bản của việc sử dụng thích nghi EIF và bù chuyển động khối con afin như sau.

1. Tính kích thước khối con tốt nhất $M \times N$ dựa vào các thông số mô hình chuyển động afin.
2. Nếu cả độ rộng khối con tốt nhất M và độ cao khối con tốt nhất N lớn hơn hoặc bằng 8, thì thực hiện việc bù chuyển động khối con với khối con $M \times N$.
3. Theo cách khác, kiểm tra các điều kiện có khả năng ứng dụng EIF
 - a. Điều kiện có khả năng ứng dụng EIF 1
 - b. Điều kiện có khả năng ứng dụng EIF 2
 - c. ...
 - d. Nếu tất cả các điều kiện có khả năng ứng dụng EIF được đáp ứng, thì thực hiện việc bù chuyển động EIF
 - e. Theo cách khác thiết đặt $M = \max(M, 8)$, $N = \max(N, 8)$ và thực hiện việc bù chuyển động khối con với khối con $M \times N$.

Chi tiết về một số bước của thuật toán cơ bản này được mô tả dưới đây.

Bước 1. Tính toán kích thước khối con tốt nhất

Một phương pháp dẫn ra kích thước khối con afin dựa vào các độ chênh lệch vector chuyển động của các điểm điều khiển afin và độ rộng và độ cao của khối afin. Kích thước

khối con $M \times N$ có thể được dẫn ra bởi phương trình (2-1), trong đó $MvPre$ là độ chính xác vector chuyển động (ví dụ, 1/4 điểm ảnh trong chuẩn HEVC hoặc 1/16 điểm ảnh trong các chuẩn VVC và EVC), và các thông số mô hình chuyển động affin $dHorX$, $dHorY$, $dVerX$, $dVerY$ được tính theo các phương trình (1-4) – (1-7) đối với mô hình 6 thông số và theo các phương trình (1-8) – (1-11) đối với mô hình 4 thông số.

$$\begin{cases} M = clip3(4, w, \frac{w \times MvPre}{\max(abs(dHorX), abs(dHorY))}) \\ N = clip3(4, h, \frac{h \times MvPre}{\max(abs(dVerX), abs(dVerY))}) \end{cases} \quad (2-1)$$

M và N được dẫn ra theo phương trình (2-1) sẽ được điều chỉnh xuống nếu cần để đảm bảo rằng w và h lần lượt có thể phân chia được bởi M và N .

Phương pháp khác là xây dựng bảng tra cứu 3 chiều, sau đó nhận kích thước khối con trực tiếp từ bảng tra cứu theo độ chênh lệch vector chuyển động, kích thước khối, và độ chính xác vector chuyển động. Ví dụ, thiết đặt M tới $Table_M[x][y][z]$, trong đó x bằng $\max(abs(v_{1x} - v_{0x}), abs(v_{1y} - v_{0y}))$, y bằng độ rộng khối affin, z bằng độ chính xác vector chuyển động; thiết đặt N tới $Table_N[x][y][z]$, trong đó x bằng $\max(abs(v_{2x} - v_{0x}), abs(v_{2y} - v_{0y}))$, y bằng độ cao khối affin, z bằng độ chính xác vector chuyển động.

Bước 3. Các điều kiện có khả năng ứng dụng EIF

Các điều kiện có khả năng ứng dụng EIF có thể ví dụ như sau

1. Các giới hạn độ rộng dải bộ nhớ. Loại của các giới hạn này đảm bảo rằng kích thước của vùng trong ảnh tham chiếu tương ứng với khối affin hiện thời (khối EIF) không lớn hơn ngưỡng được định rõ trước T . Ví dụ về định nghĩa của vùng trong ảnh tham chiếu tương ứng với khối affin hiện thời được mô tả trên Fig.9.
2. Bộ đệm bên trong được giới hạn bởi các dòng R , trong đó R là trị số được định rõ trước và có thể ví dụ là 3, 4, 5 hoặc lớn hơn. Điều này nghĩa là trong suốt thời gian xử lý của một dòng (một hàng) của khối (khối con) hiện thời không nhiều hơn các dòng R từ ảnh tham chiếu có thể được sử dụng.
3. Truy cập bộ nhớ sẽ là tuần tự, mà nghĩa là nếu đối với dòng thứ i của khối hiện thời dòng thứ j của ảnh tham chiếu được nạp, sau đó đối với dòng thứ $(i+1)$ của khối hiện thời chỉ các dòng $j+1, j+2, \dots$ có thể được nạp.
4. Không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất.

5. Các bất đẳng thức cho các thông số mô hình chuyển động afin, ví dụ như sau

$$a. \begin{cases} a \leq dHorX \leq b \\ c \leq dHorY \leq d \\ e \leq dVerX \leq f \\ g \leq dVerY \leq h \end{cases}, \text{ trong đó } a, b, c, d, e, f, g, h \text{ là các trị số được định}$$

rõ trước hoặc âm/dương vô cực.

$$b. \begin{cases} dHorX \leq a * dVerX \\ dVerY \leq b * dHorY \end{cases}, \text{ trong đó } a \text{ và } b \text{ là các trị số được định rõ trước}$$

Ví dụ 1 sử dụng phương án 1 trong MPEG-5/EVC

Theo ví dụ này, kích thước khối con EIF bằng 4x4, số các dòng được nạp lớn nhất trong suốt thời gian xử lý dòng thứ nhất R bằng 3, dX[0] tương ứng với dHorX, dX[1] tương ứng với dHorY, dY[0] tương ứng với dVerX và dY[1] tương ứng với dVerY.

Các biến dX[0], dX[1], dY[0], dY[1] theo độ chính xác 1/512 .

...

Biến eifSubblockSize được thiết đặt bằng 4.

Biến eifCanBeApplied được dẫn ra như sau:

- eifCanBeApplied được thiết đặt bằng đúng
- Các mảng X[i], Y[i] được dẫn ra như sau:
 - X[0] = 0
 - X[1] = (eifSubblockSize + 1) * (dX[0] + (1 << 9))
 - X[2] = (eifSubblockSize + 1) * dY[0]
 - X[3] = X[1] + X[2]
 - Y[0] = 0
 - Y[1] = (eifSubblockSize + 1) * dX[1]
 - Y[2] = (eifSubblockSize + 1) * (dY[1] + (1 << 9))
 - Y[3] = Y[1] + Y[2]
- Biến Xmax được thiết đặt bằng giá trị lớn nhất của X[i] với i bằng 0..3
- Biến Xmin được thiết đặt bằng giá trị nhỏ nhất của X[i] với i bằng 0..3
- Biến Ymax được thiết đặt bằng giá trị lớn nhất của Y[i] với i bằng 0..3
- Biến Ymin được thiết đặt bằng giá trị nhỏ nhất của Y[i] với i bằng 0..3

- Biến W được thiết đặt bằng $(X_{\max} - X_{\min} + (1 \ll 9) - 1) \gg 9$
- Biến H được thiết đặt bằng $(Y_{\max} - Y_{\min} + (1 \ll 9) - 1) \gg 9$
- Nếu $(W + 2) * (H + 2)$ lớn hơn 81, biến `elfCanBeApplied` bằng sai
- Theo cách khác,
 - Nếu $dY[1]$ nhỏ hơn $(-1 \ll 9)$, thì biến `elfCanBeApplied` bằng sai
 - Theo cách khác,
 - Nếu $(\max(0, dY[1]) + \text{Abs}(dX[1])) * (1 + \text{elfSubblockSize})$ lớn hơn $(1 \ll 9)$ thì biến `elfCanBeApplied` bằng sai.
- Nếu `elfCanBeApplied` bằng sai so với các biến `sizeSbX` và `sizeSbY` được điều chỉnh như sau:

- $\text{sizeSbX} = \max(8, \text{sizeSbX})$
- $\text{sizeSbY} = \max(8, \text{sizeSbY})$

...

Ví dụ 2 sử dụng phương án 1 trong MPEG-5/EVC

Theo ví dụ này, kích thước khối con EIF bằng 8x8, số các dòng được nạp lớn nhất trong suốt thời gian xử lý dòng thứ nhất R bằng 4, $dX[0]$ tương ứng với $dHorX$, $dX[1]$ tương ứng với $dHorY$, $dY[0]$ tương ứng với $dVerX$ và $dY[1]$ tương ứng với $dVerY$.

Các biến $dX[0]$, $dX[1]$, $dY[0]$, $dY[1]$ theo độ chính xác 1/512 .

...

Biến `elfSubblockSize` được thiết đặt bằng 8.

Biến `elfCanBeApplied` được dẫn ra như sau:

- `elfCanBeApplied` được thiết lập bằng đúng
- Các mảng $X[i]$, $Y[i]$ được dẫn ra như sau:
 - $X[0] = 0$
 - $X[1] = (\text{elfSubblockSize} + 1) * (dX[0] + (1 \ll 9))$
 - $X[2] = (\text{elfSubblockSize} + 1) * dY[0]$
 - $X[3] = X[1] + X[2]$
 - $Y[0] = 0$
 - $Y[1] = (\text{elfSubblockSize} + 1) * dX[1]$
 - $Y[2] = (\text{elfSubblockSize} + 1) * (dY[1] + (1 \ll 9))$
 - $Y[3] = Y[1] + Y[2]$
- Biến $X_{\max}$ được thiết đặt bằng giá trị lớn nhất của $X[i]$ với i bằng 0..3

- Biến X_{min} được thiết đặt bằng giá trị nhỏ nhất của $X[i]$ với i bằng 0..3
- Biến Y_{max} được thiết đặt bằng giá trị lớn nhất của $Y[i]$ với i bằng 0..3
- Biến Y_{min} được thiết đặt bằng giá trị nhỏ nhất của $Y[i]$ với i bằng 0..3
- Biến W được thiết đặt bằng $(X_{max} - X_{min} + (1 \ll 9) - 1) \gg 9$
- Biến H được thiết đặt bằng $(Y_{max} - Y_{min} + (1 \ll 9) - 1) \gg 9$
- Nếu $(W + 2) * (H + 2)$ lớn hơn 225, biến `elfCanBeApplied` bằng sai
- Theo cách khác,
 - Nếu $dY[1]$ nhỏ hơn $(-1 \ll 9)$, thì biến `elfCanBeApplied` bằng sai
 - Theo cách khác,
 - Nếu $(\max(0, dY[1]) + \text{Abs}(dX[1])) * (1 + \text{elfSubblockSize})$ lớn hơn $2 * (1 \ll 9)$ thì biến `elfCanBeApplied` bằng sai.
- Nếu `elfCanBeApplied` bằng sai so với các biến `sizeSbX` và `sizeSbY` được điều chỉnh như sau:

- $\text{sizeSbX} = \max(8, \text{sizeSbX})$
- $\text{sizeSbY} = \max(8, \text{sizeSbY})$

...

Ví dụ 3 sử dụng phương án 1 trong MPEG-5/EVC

Theo ví dụ này, kích thước khối con EIF bằng 8x8, số các dòng được nạp lớn nhất trong suốt thời gian xử lý dòng thứ nhất R bằng 5, $dX[0]$ tương ứng với $dHorX$, $dX[1]$ tương ứng với $dHorY$, $dY[0]$ tương ứng với $dVerX$ và $dY[1]$ tương ứng với $dVerY$.

Các biến $dX[0]$, $dX[1]$, $dY[0]$, $dY[1]$ theo độ chính xác 1/512.

...

Biến `elfSubblockSize` được thiết đặt bằng 8.

Biến `elfCanBeApplied` được dẫn ra như sau:

- `elfCanBeApplied` được thiết đặt là đúng
- Các mảng $X[i]$, $Y[i]$ được dẫn ra như sau:
- $X[0] = 0$
- $X[1] = (\text{elfSubblockSize} + 1) * (dX[0] + (1 \ll 9))$
- $X[2] = (\text{elfSubblockSize} + 1) * dY[0]$
- $X[3] = X[1] + X[2]$
- $Y[0] = 0$
- $Y[1] = (\text{elfSubblockSize} + 1) * dX[1]$

- $Y[2] = (\text{eifSubblockSize} + 1) * (dY[1] + (1 \ll 9))$
- $Y[3] = Y[1] + Y[2]$
- Biến $X_{\max}$ được thiết đặt bằng giá trị lớn nhất của $X[i]$ với i bằng 0..3
- Biến $X_{\min}$ được thiết đặt bằng giá trị nhỏ nhất của $X[i]$ với i bằng 0..3
- Biến $Y_{\max}$ được thiết đặt bằng giá trị lớn nhất của $Y[i]$ với i bằng 0..3
- Biến $Y_{\min}$ được thiết đặt bằng giá trị nhỏ nhất của $Y[i]$ với i bằng 0..3
- Biến W được thiết đặt bằng $(X_{\max} - X_{\min} + (1 \ll 9) - 1) \gg 9$
- Biến H được thiết đặt bằng $(Y_{\max} - Y_{\min} + (1 \ll 9) - 1) \gg 9$
- Nếu $(W + 2) * (H + 2)$ lớn hơn 225, biến eifCanBeApplied bằng sai
- Theo cách khác,
 - Nếu $dY[1]$ nhỏ hơn $(-1 \ll 9)$, thì biến eifCanBeApplied bằng sai
 - Theo cách khác,
 - Nếu $(\max(0, dY[1]) + \text{Abs}(dX[1])) * (1 + \text{eifSubblockSize})$ lớn hơn $3 * (1 \ll 9)$ thì biến eifCanBeApplied bằng sai.
- Nếu eifCanBeApplied bằng sai so với các biến sizeSbX và sizeSbY được điều chỉnh như sau:

- $\text{sizeSbX} = \max(8, \text{sizeSbX})$
- $\text{sizeSbY} = \max(8, \text{sizeSbY})$

...

Ví dụ 4 sử dụng phương án 1 trong MPEG-5/EVC

Theo ví dụ này, kích thước khối con EIF bằng 8x8, chỉ giới hạn độ rộng dải bộ nhớ được kiểm tra đối với EIF, $dX[0]$ tương ứng với $dHorX$, $dX[1]$ tương ứng với $dHorY$, $dY[0]$ tương ứng với $dVerX$ và $dY[1]$ tương ứng với $dVerY$.

Các biến $dX[0]$, $dX[1]$, $dY[0]$, $dY[1]$ theo độ chính xác 1/512 .

...

Biến eifSubblockSize được thiết đặt bằng 8.

Biến eifCanBeApplied được dẫn ra như sau:

- eifCanBeApplied được thiết lập là đúng
- Các mảng $X[i]$, $Y[i]$ được dẫn ra như sau:
- $X[0] = 0$
- $X[1] = (\text{eifSubblockSize} + 1) * (dX[0] + (1 \ll 9))$
- $X[2] = (\text{eifSubblockSize} + 1) * dY[0]$

- $X[3] = X[1] + X[2]$
- $Y[0] = 0$
- $Y[1] = (\text{eifSubblockSize} + 1) * dX[1]$
- $Y[2] = (\text{eifSubblockSize} + 1) * (dY[1] + (1 \ll 9))$
- $Y[3] = Y[1] + Y[2]$
- Biến $X_{\max}$ được thiết đặt bằng giá trị lớn nhất của $X[i]$ với i bằng 0..3
- Biến $X_{\min}$ được thiết đặt bằng giá trị nhỏ nhất của $X[i]$ với i bằng 0..3
- Biến $Y_{\max}$ được thiết đặt bằng giá trị lớn nhất của $Y[i]$ với i bằng 0..3
- Biến $Y_{\min}$ được thiết đặt bằng giá trị nhỏ nhất của $Y[i]$ với i bằng 0..3
- Biến W được thiết đặt bằng $(X_{\max} - X_{\min} + (1 \ll 9) - 1) \gg 9$
- Biến H được thiết đặt bằng $(Y_{\max} - Y_{\min} + (1 \ll 9) - 1) \gg 9$
- Nếu $(W + 2) * (H + 2)$ lớn hơn 225, biến eifCanBeApplied bằng sai
- Nếu eifCanBeApplied bằng sai so với các biến sizeSbX và sizeSbY được điều

chỉnh như sau:

- $\text{sizeSbX} = \max(8, \text{sizeSbX})$
- $\text{sizeSbY} = \max(8, \text{sizeSbY})$

...

Ví dụ 5 sử dụng phương án 1 trong MPEG-5/EVC

Theo ví dụ này, kích thước khối con EIF bằng 4x4, chỉ giới hạn độ rộng dải bộ nhớ được kiểm tra đối với EIF, $dX[0]$ tương ứng với $dHorX$, $dX[1]$ tương ứng với $dHorY$, $dY[0]$ tương ứng với $dVerX$ và $dY[1]$ tương ứng với $dVerY$.

Các biến $dX[0]$, $dX[1]$, $dY[0]$, $dY[1]$ theo độ chính xác 1/512 .

...

Biến eifSubblockSize được thiết đặt bằng 8.

Biến eifCanBeApplied được dẫn ra như sau:

- eifCanBeApplied được thiết lập là đúng
- Các mảng $X[i]$, $Y[i]$ được dẫn ra như sau:
- $X[0] = 0$
- $X[1] = (\text{eifSubblockSize} + 1) * (dX[0] + (1 \ll 9))$
- $X[2] = (\text{eifSubblockSize} + 1) * dY[0]$
- $X[3] = X[1] + X[2]$
- $Y[0] = 0$

- $Y[1] = (\text{eifSubblockSize} + 1) * dX[1]$
- $Y[2] = (\text{eifSubblockSize} + 1) * (dY[1] + (1 \ll 9))$
- $Y[3] = Y[1] + Y[2]$
- Biến Xmax được thiết đặt bằng giá trị lớn nhất của $X[i]$ với i bằng 0..3
- Biến Xmin được thiết đặt bằng giá trị nhỏ nhất của $X[i]$ với i bằng 0..3
- Biến Ymax được thiết đặt bằng giá trị lớn nhất của $Y[i]$ với i bằng 0..3
- Biến Ymin được thiết đặt bằng giá trị nhỏ nhất của $Y[i]$ với i bằng 0..3
- Biến W được thiết đặt bằng $(Xmax - Xmin + (1 \ll 9) - 1) \gg 9$
- Biến H được thiết đặt bằng $(Ymax - Ymin + (1 \ll 9) - 1) \gg 9$
- Nếu $(W + 2) * (H + 2)$ lớn hơn 81, biến eifCanBeApplied bằng sai
- Nếu eifCanBeApplied bằng sai so với các biến sizeSbX và sizeSbY được điều

chỉnh như sau:

- $\text{sizeSbX} = \max(8, \text{sizeSbX})$
- $\text{sizeSbY} = \max(8, \text{sizeSbY})$

...

Phương án 2

Sử dụng thích nghi EIF và bù chuyển động affin trên cơ sở khối. Thuật toán cơ bản.

Thuật toán cơ bản của việc sử dụng thích nghi EIF và bù chuyển động khối con affin như sau.

1. Tính kích thước khối con tốt nhất $M \times N$ dựa vào các thông số mô hình chuyển động affin.
2. Nếu cả độ rộng khối con tốt nhất M và độ cao khối con tốt nhất N lớn hơn hoặc bằng 8, thì thực hiện việc bù chuyển động khối con với khối con $M \times N$.
3. Theo cách khác, kiểm tra các điều kiện có khả năng ứng dụng EIF
 - a. Điều kiện có khả năng ứng dụng EIF 1
 - b. Điều kiện có khả năng ứng dụng EIF 2
 - c. ...
 - d. Điều kiện có khả năng ứng dụng EIF P
 - e. Nếu tất cả các điều kiện có khả năng ứng dụng EIF được đáp ứng, thì thực hiện việc bù chuyển động EIF. Quy trình bù chuyển động EIF bao gồm các bước:
 - i. Kiểm tra điều kiện có khả năng ứng dụng EIF P+1

- ii. Kiểm tra điều kiện có khả năng ứng dụng EIF P+2
- iii. ...
- iv. Kiểm tra điều kiện có khả năng ứng dụng EIF P+K
- v. Nếu một trong số các điều kiện có khả năng ứng dụng EIF P+1 .. P+K không được đáp ứng, thì
 - tính khoảng vector chuyển động thứ nhất mà đưa ra việc hoàn tất của các điều kiện mà đã không được đáp ứng.
 - cắt các vector chuyển động được tính theo chế độ afin để đảm bảo, rằng các vector này trong khoảng vector chuyển động thứ nhất

f. Theo cách khác thiết đặt $M' = \max(M, 8)$, $N' = \max(N, 8)$ và thực hiện việc bù chuyển động khối con với khối con $M \times N$.

Theo một số ví dụ P và K có thể bằng không mà nghĩa là thuật toán nêu trên sẽ không có các bước 3.a – 3.d nếu P bằng không và sẽ không có các bước 3.e.i – 3.e.v nếu K bằng không.

Chi tiết về một số bước của thuật toán cơ bản này được mô tả dưới đây.

Bước 1. Tính toán kích thước khối con tốt nhất

Một phương pháp dẫn ra kích thước khối con afin dựa vào các độ chênh lệch vector chuyển động của các điểm điều khiển afin và độ rộng và độ cao của khối afin. Kích thước khối con $M \times N$ có thể được dẫn ra bởi phương trình (2-1), trong đó $MvPre$ là độ chính xác vector chuyển động (ví dụ, 1/4 điểm ảnh trong chuẩn HEVC hoặc 1/16 điểm ảnh trong các chuẩn VVC và EVC), và các thông số mô hình chuyển động afin $dHorX$, $dHorY$, $dVerX$, $dVerY$ được tính theo các phương trình (1-4) – (1-7) đối với mô hình 6 thông số và theo các phương trình (1-8) – (1-11) đối với mô hình 4 thông số.

$$\begin{cases} M = clip3(4, w, \frac{w \times MvPre}{\max(abs(dHorX), abs(dHorY))}) \\ N = clip3(4, h, \frac{h \times MvPre}{\max(abs(dVerX), abs(dVerY))}) \end{cases} \quad (2-1)$$

M và N được dẫn ra theo phương trình (2-1) sẽ được điều chỉnh xuống nếu cần để đảm bảo rằng w và h lần lượt có thể phân chia được bởi M và N.

Phương pháp khác là xây dựng bảng tra cứu 3 chiều, sau đó nhận kích thước khối con trực tiếp từ bảng tra cứu theo độ chênh lệch vector chuyển động, kích thước khối, và độ chính xác vector chuyển động. Ví dụ, thiết đặt M tới $Table_M[x][y][z]$, trong đó x bằng

$\max(abs(v_{1x} - v_{0x}), abs(v_{1y} - v_{0y}))$, y bằng độ rộng khối afin, z bằng độ chính xác vector chuyển động; thiết đặt N tới Table_N[x][y][z], trong đó x bằng $\max(abs(v_{2x} - v_{0x}), abs(v_{2y} - v_{0y}))$, y bằng độ cao khối afin, z bằng độ chính xác vector chuyển động.

Bước 3. Các điều kiện có khả năng ứng dụng EIF

Các điều kiện có khả năng ứng dụng EIF có thể ví dụ như sau

1. Các giới hạn độ rộng dải bộ nhớ. Loại của các giới hạn này đảm bảo rằng kích thước của vùng trong ảnh tham chiếu tương ứng với khối afin hiện thời (khối EIF) không lớn hơn ngưỡng được định rõ trước T. Ví dụ về định nghĩa của vùng trong ảnh tham chiếu tương ứng với khối afin hiện thời được mô tả trên Fig.9.
2. Bộ đệm bên trong được giới hạn bởi các dòng R, trong đó R là trị số được định rõ trước và có thể ví dụ là 3, 4, 5 hoặc lớn hơn. Điều này nghĩa là trong suốt thời gian xử lý của một dòng (một hàng) của khối (khối con) hiện thời không nhiều hơn các dòng R từ ảnh tham chiếu có thể được sử dụng.
3. Truy cập bộ nhớ sẽ là tuần tự, mà nghĩa là nếu đối với dòng thứ i của khối hiện thời dòng thứ j của ảnh tham chiếu được nạp, sau đó đối với dòng thứ (i+1) của khối hiện thời chỉ các dòng j+1, j+2, ... có thể được nạp.
4. Không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất.
5. Các bất đẳng thức cho các thông số mô hình chuyển động afin, ví dụ như sau

$$a. \begin{cases} a \leq dHorX \leq b \\ c \leq dHorY \leq d \\ e \leq dVerX \leq f \\ g \leq dVerY \leq h \end{cases}, \text{ trong đó } a, b, c, d, e, f, g, h \text{ là các trị số được định}$$

rõ trước hoặc âm/dương vô cực.

$$b. \begin{cases} dHorX \leq a * dVerX \\ dVerY \leq b * dHorY \end{cases}, \text{ trong đó } a \text{ và } b \text{ là các trị số được định rõ trước}$$

Theo một ví dụ thuật toán cơ bản như sau:

1. Tính kích thước khối con tốt nhất MxN dựa vào các thông số mô hình chuyển động afin.
2. Nếu cả độ rộng khối con tốt nhất M hoặc độ cao khối con tốt nhất N lớn hơn hoặc bằng 8, thì thực hiện việc bù chuyển động khối con với khối con MxN.

3. Theo cách khác, kiểm tra các điều kiện có khả năng ứng dụng EIF:

a. Điều kiện có khả năng ứng dụng EIF 1: Bộ đệm bên trong được giới hạn bởi các dòng R, trong đó R là trị số được định rõ trước và có thể ví dụ là 3, 4, 5 hoặc lớn hơn. Điều này nghĩa là trong suốt thời gian xử lý của một dòng (một hàng) của khối (khối con) hiện thời không nhiều hơn các dòng R từ ảnh tham chiếu có thể được sử dụng.

b. Điều kiện có khả năng ứng dụng EIF 2: Truy cập bộ nhớ sẽ là tuần tự, mà nghĩa là nếu đối với dòng thứ i của khối hiện thời dòng thứ j của ảnh tham chiếu được nạp, sau đó đối với dòng thứ (i+1) của khối hiện thời chỉ các dòng j+1, j+2, ... có thể được nạp.

c. Điều kiện có khả năng ứng dụng EIF 3: Không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất.

d. ...

e. Điều kiện có khả năng ứng dụng EIF P

f. Nếu tất cả các điều kiện có khả năng ứng dụng EIF được đáp ứng, thì thực hiện việc bù chuyển động EIF. Quy trình bù chuyển động EIF bao gồm các bước:

i. Kiểm tra điều kiện có khả năng ứng dụng EIF P+ 1: Kiểm tra các giới hạn độ rộng dải bộ nhớ. Loại của các giới hạn này đảm bảo rằng kích thước của vùng trong ảnh tham chiếu tương ứng với khối afin hiện thời (khối EIF) không lớn hơn ngưỡng được định rõ trước T. Ví dụ về định nghĩa của vùng trong ảnh tham chiếu tương ứng với khối afin hiện thời được mô tả trên Fig.9.

ii. ...

iii. Kiểm tra điều kiện có khả năng ứng dụng EIF P+K

iv. Nếu một trong số các điều kiện có khả năng ứng dụng EIF P+1 .. P+K không được đáp ứng, thì

1. tính khoảng vector chuyển động thứ nhất mà đưa ra việc hoàn tất của các điều kiện mà đã không được đáp ứng.

2. cắt các vector chuyển động được tính theo chế độ affin để đảm bảo, rằng các vector này trong khoảng vector chuyển động thứ nhất

g. Theo cách khác thiết đặt $M' = \max(M, 8)$, $N' = \max(N, 8)$ và thực hiện việc bù chuyển động khối con với khối con $M \times N$.

Theo một số ví dụ N có thể bằng 3 mà nghĩa là thuật toán nêu trên sẽ không có các bước 3.d – 3.e. Theo một số ví dụ K có thể bằng 1 mà nghĩa là thuật toán nêu trên sẽ không có các bước 3.f.ii – 3.ii.

Phương án 3

Sử dụng thích nghi EIF và bù chuyển động affin trên cơ sở khối. Thuật toán cơ bản.

Thuật toán cơ bản của việc sử dụng thích nghi EIF và bù chuyển động khối con affin như sau.

1. Tính kích thước khối con tốt nhất $M \times N$ dựa vào các thông số mô hình chuyển động affin.

2. Nếu cả độ rộng khối con tốt nhất M và độ cao khối con tốt nhất N lớn hơn hoặc bằng 8, thì thực hiện việc bù chuyển động khối con với khối con $M \times N$.

3. Theo cách khác, kiểm tra các điều kiện có khả năng ứng dụng EIF

a. Điều kiện có khả năng ứng dụng EIF 1

b. Điều kiện có khả năng ứng dụng EIF 2

c. ...

d. Điều kiện có khả năng ứng dụng EIF P

e. Nếu tất cả các điều kiện có khả năng ứng dụng EIF được đáp ứng, thì thực hiện việc bù chuyển động EIF. Quy trình bù chuyển động EIF bao gồm các bước:

i. Kiểm tra điều kiện có khả năng ứng dụng EIF $P+1$

ii. Kiểm tra điều kiện có khả năng ứng dụng EIF $P+2$

iii. ...

iv. Kiểm tra điều kiện có khả năng ứng dụng EIF $P+K$

v. Nếu một trong số các điều kiện có khả năng ứng dụng EIF $P+1$

.. $P+K$ không được đáp ứng, thì

– tính khoảng vector chuyển động thứ nhất mà đưa ra việc hoàn tất của các điều kiện mà đã không được đáp ứng.

– thiết đặt khoảng vector chuyển động thứ hai bằng khoảng vector chuyển động thứ nhất

vi. Theo cách khác

- tính khoảng vector chuyển động thứ ba
- thiết đặt khoảng vector chuyển động thứ hai bằng khoảng vector chuyển động thứ ba

vii. Cắt các vector chuyển động được tính theo chế độ affin để đảm bảo, rằng các vector này trong khoảng vector chuyển động thứ hai.

f. Theo cách khác thiết đặt $M' = \max(M, 8)$, $N' = \max(N, 8)$ và thực hiện việc bù chuyển động khối con với khối con $M \times N$.

Theo một số ví dụ P và K có thể bằng không mà nghĩa là thuật toán nêu trên sẽ không có các bước 3.a – 3.d nếu P bằng không và sẽ không có các bước 3.e.i – 3.e.v nếu K bằng không.

Chi tiết về một số bước của thuật toán cơ bản này được mô tả dưới đây.

Bước 1. Tính toán kích thước khối con tốt nhất

Một phương pháp dẫn ra kích thước khối con affin dựa vào các độ chênh lệch vector chuyển động của các điểm điều khiển affin và độ rộng và độ cao của khối affin. Kích thước khối con $M \times N$ có thể được dẫn ra bởi phương trình (2-1), trong đó $MvPre$ là độ chính xác vector chuyển động (ví dụ, 1/4 điểm ảnh trong chuẩn HEVC hoặc 1/16 điểm ảnh trong các chuẩn VVC và EVC), và các thông số mô hình chuyển động affin $dHorX$, $dHorY$, $dVerX$, $dVerY$ được tính theo các phương trình (1-4) – (1-7) đối với mô hình 6 thông số và theo các phương trình (1-8) – (1-11) đối với mô hình 4 thông số.

$$\begin{cases} M = clip3(4, w, \frac{w \times MvPre}{\max(abs(dHorX), abs(dHorY))}) \\ N = clip3(4, h, \frac{h \times MvPre}{\max(abs(dVerX), abs(dVerY))}) \end{cases} \quad (2-1)$$

M và N được dẫn ra theo phương trình (2-1) sẽ được điều chỉnh xuống nếu cần để đảm bảo rằng w và h lần lượt có thể phân chia được bởi M và N.

Phương pháp khác là xây dựng bảng tra cứu 3 chiều, sau đó nhận kích thước khối con trực tiếp từ bảng tra cứu theo độ chênh lệch vector chuyển động, kích thước khối, và độ chính xác vector chuyển động. Ví dụ, thiết đặt M tới $Table_M[x][y][z]$, trong đó x bằng $\max(abs(v_{1x} - v_{0x}), abs(v_{1y} - v_{0y}))$, y bằng độ rộng khối affin, z bằng độ chính xác vector chuyển động; thiết đặt N tới $Table_N[x][y][z]$, trong đó x bằng $\max(abs(v_{2x} -$

$v_{0x}), abs(v_{2y} - v_{0y}))$, y bằng độ cao khối afin, z bằng độ chính xác vector chuyển động.

Bước 3. Các điều kiện có khả năng ứng dụng EIF

Các điều kiện có khả năng ứng dụng EIF có thể ví dụ như sau

1. Các giới hạn độ rộng dải bộ nhớ. Loại của các giới hạn này đảm bảo rằng kích thước của vùng trong ảnh tham chiếu tương ứng với khối afin hiện thời (khối EIF) không lớn hơn ngưỡng được định rõ trước T. Ví dụ về định nghĩa của vùng trong ảnh tham chiếu tương ứng với khối afin hiện thời được mô tả trên Fig.9.
2. Bộ đệm bên trong được giới hạn bởi các dòng R, trong đó R là trị số được định rõ trước và có thể ví dụ là 3, 4, 5 hoặc lớn hơn. Điều này nghĩa là trong suốt thời gian xử lý của một dòng (một hàng) của khối (khối con) hiện thời không nhiều hơn các dòng R từ ảnh tham chiếu có thể được sử dụng.
3. Truy cập bộ nhớ sẽ là tuần tự, mà nghĩa là nếu đối với dòng thứ i của khối hiện thời dòng thứ j của ảnh tham chiếu được nạp, sau đó đối với dòng thứ (i+1) của khối hiện thời chỉ các dòng j+1, j+2, ... có thể được nạp.
4. Không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất.
5. Các bất đẳng thức cho các thông số mô hình chuyển động afin, ví dụ như sau

$$a. \begin{cases} a \leq dHorX \leq b \\ c \leq dHorY \leq d \\ e \leq dVerX \leq f \\ g \leq dVerY \leq h \end{cases}, \text{ trong đó } a, b, c, d, e, f, g, h \text{ là các trị số được định}$$

rõ trước hoặc âm/dương vô cực.

$$b. \begin{cases} dHorX \leq a * dVerX \\ dVerY \leq b * dHorY \end{cases}, \text{ trong đó } a \text{ và } b \text{ là các trị số được định rõ trước}$$

Theo một ví dụ thuật toán cơ bản như sau:

1. Tính kích thước khối con tốt nhất MxN dựa vào các thông số mô hình chuyển động afin.
2. Nếu cả độ rộng khối con tốt nhất M hoặc độ cao khối con tốt nhất N lớn hơn hoặc bằng 8, thì thực hiện việc bù chuyển động khối con với khối con MxN.
3. Theo cách khác, kiểm tra các điều kiện có khả năng ứng dụng EIF:

a. Điều kiện có khả năng ứng dụng EIF 1: Bộ đệm bên trong được giới hạn bởi các dòng R, trong đó R là trị số được định rõ trước và có thể ví dụ là 3, 4, 5 hoặc lớn hơn. Điều này nghĩa là trong suốt thời gian xử lý của một dòng (một hàng) của khối (khối con) hiện thời không nhiều hơn các dòng R từ ảnh tham chiếu có thể được sử dụng.

b. Điều kiện có khả năng ứng dụng EIF 2: Truy cập bộ nhớ sẽ là tuần tự, mà nghĩa là nếu đối với dòng thứ i của khối hiện thời dòng thứ j của ảnh tham chiếu được nạp, sau đó đối với dòng thứ (i+1) của khối hiện thời chỉ các dòng j+1, j+2, ... có thể được nạp.

c. Điều kiện có khả năng ứng dụng EIF 3: Không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất.

d. ...

e. Điều kiện có khả năng ứng dụng EIF P

f. Nếu tất cả các điều kiện có khả năng ứng dụng EIF được đáp ứng, thì thực hiện việc bù chuyển động EIF. Quy trình bù chuyển động EIF bao gồm các bước:

i. Kiểm tra điều kiện có khả năng ứng dụng EIF P + 1: Kiểm tra các giới hạn độ rộng dải bộ nhớ. Loại của các giới hạn này đảm bảo rằng kích thước của vùng trong ảnh tham chiếu tương ứng với khối afin hiện thời (khối EIF) không lớn hơn ngưỡng được định rõ trước T. Ví dụ về định nghĩa của vùng trong ảnh tham chiếu tương ứng với khối afin hiện thời được mô tả trên Fig.9. (Phần này sẽ được thay đổi theo phương án 4)

ii. ...

iii. Kiểm tra điều kiện có khả năng ứng dụng EIF P+K

iv. Nếu một trong số các điều kiện có khả năng ứng dụng EIF P+1 .. P+K không được đáp ứng, thì

1. tính khoảng vector chuyển động thứ nhất mà đưa ra việc hoàn tất của các điều kiện mà đã không được đáp ứng. (Phần này sẽ được thay đổi theo phương án 4)

2. thiết đặt khoảng vector chuyển động thứ hai bằng khoảng vector chuyển động thứ nhất

v. Theo cách khác

1. tính khoảng vector chuyển động thứ ba

2. thiết đặt khoảng vector chuyển động thứ hai bằng khoảng vector chuyển động thứ ba

vi. Cắt các vector chuyển động được tính theo chế độ affin để đảm bảo, rằng các vector này trong khoảng vector chuyển động thứ hai.

g. Theo cách khác thiết đặt $M = \max(M, 8)$, $N = \max(N, 8)$ và thực hiện việc bù chuyển động khối con với khối con $M \times N$.

Theo một số ví dụ N có thể bằng 3 mà nghĩa là thuật toán nêu trên sẽ không có các bước 3.d – 3.e. Theo một số ví dụ K có thể bằng 1 mà nghĩa là thuật toán nêu trên sẽ không có các bước 3.f.ii – 3.f.iii.

Theo một ví dụ các bước 3.f.i và 3.f.iv.1 được thực hiện như sau:

Tính toán mức tiêu thụ truy cập bộ nhớ cho khối affin trong trường hợp sử dụng EIF

Khi EIF được sử dụng cho việc bù chuyển động của khối affin với kích thước $W \times H$, các bước sau đây được thực hiện cho việc tính toán mức tiêu thụ truy cập bộ nhớ.

1. Dẫn ra vị trí của mỗi mẫu góc của khối affin.

2. Dẫn ra vị trí của mỗi mẫu góc của khối mà được sử dụng trên bước 3 của EIF (biểu thị nó là khối trung gian EIF).

3. Dẫn ra các vector chuyển động cho mỗi mẫu góc của khối trung gian EIF.

4. Dẫn ra vị trí của khối được biến đổi trong ảnh tham chiếu.

5. Dẫn ra kích thước hộp giới hạn cho khối được biến đổi.

6. Nhận mức tiêu thụ truy cập bộ nhớ dựa vào kích thước khối được biến đổi và độ dài bộ lọc (EIF sử dụng nội suy song tuyến tính, vì vậy độ dài bộ lọc bằng 2).

Chi tiết về cách thực hiện của các bước này được mô tả dưới đây.

Bước 1. Dẫn ra vị trí của mỗi mẫu góc của khối affin

Biểu thị (x_0, y_0) là tọa độ của mẫu trên cùng bên trái của khối affin. Sau đó vị trí của khối affin có thể được mô tả bởi các tọa độ của các mẫu góc của nó (trên cùng bên trái, trên cùng bên phải, dưới cùng bên trái, dưới cùng bên phải):

$$\begin{cases} (x_0, y_0) = (x_0, y_0) \\ (x_1, y_1) = (x_0 + W - 1, y_0) \\ (x_2, y_2) = (x_0, y_0 + H - 1) \\ (x_3, y_3) = (x_0 + W - 1, y_0 + H - 1) \end{cases} \quad (2-1)$$

Bước 2. Dẫn ra vị trí của mỗi mẫu góc của khối trung gian EIF

Nội suy song tuyến tính ở bước 2 của EIF được thực hiện cho khối $(W+2) \times (H+2)$ (một lẻ điểm ảnh từ mỗi biên được bổ sung). Khối $(W+2) \times (H+2)$ này được biểu thị là khối EIF trung gian. Các tọa độ của các mẫu góc khối EIF trung gian (trên cùng bên trái, trên cùng bên phải, dưới cùng bên trái, dưới cùng bên phải) là:

$$\begin{cases} (X_0, Y_0) = (x_0 - 1, y_0 - 1) \\ (X_1, Y_1) = (x_0 + W, y_0 - 1) \\ (X_2, Y_2) = (x_0 - 1, y_0 + H) \\ (X_3, Y_3) = (x_0 + W, y_0 + H) \end{cases} \quad (2-2)$$

Các tọa độ của các góc của khối afin và của khối EIF trung gian được mô tả trên Fig 8.

Bước 3. Dẫn ra các vector chuyển động cho mỗi mẫu góc của khối trung gian EIF

Các vector chuyển động được dẫn ra theo phương trình (1-1). Theo một số ví dụ, việc cắt vector chuyển động có thể được sử dụng để đảm bảo rằng các mẫu bên ngoài của ảnh hiện thời với lẻ của một kích thước CTU không được sử dụng.

$$\begin{cases} V_0 = V_{x_0, y_0} = (mv_{0x} + (-1)dHorX + (-1)dVerX, & mv_{0y} + (-1)dHorY + (-1)dVerY) \\ V_1 = V_{x_1, y_1} = (mv_{0x} + (W)dHorX + (-1)dVerX, & mv_{0y} + (W)dHorY + (-1)dVerY) \\ V_2 = V_{x_2, y_2} = (mv_{0x} + (-1)dHorX + (H)dVerX, & mv_{0y} + (-1)dHorY + (H)dVerY) \\ V_3 = V_{x_3, y_3} = (mv_{0x} + (W)dHorX + (H)dVerX, & mv_{0y} + (W)dHorY + (H)dVerY) \end{cases} \quad (2-3)$$

Bước 4. Dẫn ra vị trí của khối được biến đổi trong ảnh tham chiếu

Lần lượt biểu thị các phần theo chiều ngang và theo chiều dọc của vector V_i là V_{ix} và V_{iy} .

Vị trí của khối được biến đổi trong ảnh tham chiếu có thể được mô tả bởi các tọa độ của các mẫu góc của nó (trên cùng bên trái, trên cùng bên phải, dưới cùng bên trái, dưới cùng bên phải):

$$\begin{cases} (X'_0, Y'_0) = (X_0 + V_{0x}, Y_0 + V_{0y}) \\ (X'_1, Y'_1) = (X_1 + V_{1x}, Y_1 + V_{1y}) \\ (X'_2, Y'_2) = (X_2 + V_{2x}, Y_2 + V_{2y}) \\ (X'_3, Y'_3) = (X_3 + V_{3x}, Y_3 + V_{3y}) \end{cases} \quad (2-4)$$

Bước 5. Dẫn ra kích thước hộp giới hạn cho khối được biến đổi

Kích thước hộp giới hạn cho khối được biến đổi trong ảnh tham chiếu có thể được tính

bởi phương trình sau đây, trong đó hàm lớn nhất quay lại trị số lớn nhất của các đối số, hàm nhỏ nhất quay lại trị số nhỏ nhất của các đối số:

$$\begin{cases} W' = \max(X'_0, X'_1, X'_2, X'_3) - \min(X'_0, X'_1, X'_2, X'_3) + 1 \\ H' = \max(Y'_0, Y'_1, Y'_2, Y'_3) - \min(Y'_0, Y'_1, Y'_2, Y'_3) + 1 \end{cases} \quad (2-5)$$

Vị trí của khối được biến đổi trong ảnh tham chiếu và hộp giới hạn tương ứng được mô tả trên Fig.9.

Theo một ví dụ, $W' = \text{Ceil}(W')$, $H' = \text{Ceil}(H')$ được thực hiện sau phương trình (2-5).

Theo ví dụ khác, $W' = \text{Floor}(W')$, $H' = \text{Floor}(H')$ được thực hiện sau phương trình (2-5).

Bước 6. Nhận mức tiêu thụ truy cập bộ nhớ

Mức tiêu thụ truy cập bộ nhớ của khối affin trong một ảnh tham chiếu có thể được quyết định, bởi kích thước hộp giới hạn cho khối được biến đổi và độ dài của bộ lọc nội suy MC cho khối chuyển động affin T' , ví dụ, 2, 4, 6, 8....:

$$Mem = (W' + T' - 1) * (H' + T' - 1) \quad (2-6)$$

Đối với EIF, nội suy song tuyến tính được sử dụng, do đó độ dài bộ lọc là 2 và mức tiêu thụ truy cập bộ nhớ bằng

$$Mem = (W' + 1) * (H' + 1) \quad (2-7)$$

Các giới hạn mô hình chuyển động affin đối với EIF

Biểu thị độ rộng dải bộ nhớ trường hợp kém nhất đích là $T = \frac{S_{wc}}{W * H}$, trong đó W và H lần lượt là độ rộng và các độ cao khối hiện thời, và S_{wc} là mức tiêu thụ truy cập bộ nhớ được phép lớn nhất cho khối hiện thời, theo độ rộng dải bộ nhớ trường hợp đích. Để đảm bảo rằng độ rộng dải bộ nhớ EIF không lớn hơn độ rộng dải bộ nhớ trường hợp đích, mức tiêu thụ truy cập bộ nhớ của khối affin sẽ bị ràng buộc theo điều kiện sau đây:

$$\begin{aligned} \frac{(W' + 1) * (H' + 1)}{W * H} &\leq T \\ \text{Hoặc} \\ (W' + 1) * (H' + 1) &\leq T * W * H \\ \text{Hoặc} \\ (W' + 1) * (H' + 1) &\leq S_{wc} \end{aligned} \quad (2-8)$$

Trị số T có thể được định rõ trước trong cả bộ mã hóa và bộ giải mã, hoặc được quy định trong tập hợp thông số của chuỗi video bộ mã hóa/giải mã, ví dụ, các mức chuỗi, mức ảnh, tập hợp thông số mức lát chẳng hạn.

Theo một ví dụ, nếu mức tiêu thụ truy cập bộ nhớ được phép lớn nhất trên mỗi mẫu được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 4x4, sau đó T có thể được dẫn ra như sau, trong đó T là độ dài của bộ lọc nội suy:

$$T = \frac{(4+T'-1)*(4+T'-1)}{4*4} \quad (2-9)$$

Đối với T' bằng 6 giới hạn như sau:

$$(W' + 1) * (H' + 1) \leq (4 + 6 - 1) * (4 + 6 - 1) \quad (2-10)$$

Theo ví dụ khác, nếu mức tiêu thụ truy cập bộ nhớ được phép lớn nhất trên mỗi mẫu được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x8, sau đó T có thể được dẫn ra như sau, trong đó T là độ dài của bộ lọc nội suy:

$$T = \frac{(8+T'-1)*(8+T'-1)}{8*8} \quad (2-11)$$

Theo ví dụ khác, mức tiêu thụ truy cập bộ nhớ được phép lớn nhất của mỗi mẫu có thể khác nhau theo hướng dự đoán của khối hiện thời, nghĩa là: khi khối hiện thời là dự đoán một chiều, sử dụng ngưỡng T_{UNI} , khi khối hiện thời là dự đoán đôi, sử dụng ngưỡng T_{BI} .

Ví dụ, T_{UNI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 4x4, T_{BI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x4, thì:

$$T_{UNI} = \frac{(4+T'-1)*(4+T'-1)}{4*4} \quad (2-12)$$

$$T_{BI} = \frac{(8+T'-1)*(4+T'-1)}{8*4} \quad (2-13)$$

Theo ví dụ khác, T_{UNI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 4x4, T_{BI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x8, thì:

$$T_{UNI} = \frac{(4+T'-1)*(4+T'-1)}{4*4} \quad (2-14)$$

$$T_{BI} = \frac{(8+T'-1)*(8+T'-1)}{8*8} \quad (2-15)$$

T' theo các ví dụ nêu trên là độ dài của bộ lọc nội suy bù chuyển động (MC) cho khối chuyển động tịnh tiến, ví dụ, 2, 4, 6, 8....

Trị số của T , T_{UNI} và T_{BI} có thể tùy thuộc vào độ rộng và độ cao của khối hiện thời.

Đối với khối afin dự đoán đôi, ràng buộc nêu trên được áp dụng riêng cho cả list0 và list1.

Theo một ví dụ khác mức tiêu thụ truy cập bộ nhớ được tính cho list0 và cho list1 là Mem_0 và Mem_1 , và tổng của các thành phần này bị giới hạn. Ví dụ, nếu T_{BI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x8, giới hạn sau đây được sử dụng:

$$Mem_0 + Mem_1 \leq 2 * (8 + T' - 1) * (8 + T' - 1) \quad (2-16)$$

Nếu các vector chuyển động của khối afin không thể theo sau các điều kiện ràng buộc (2-8), khối không thể sử dụng bù chuyển động EIF.

Theo một ví dụ, nếu khối afin không thể sử dụng bù chuyển động EIF, bù chuyển động khối con với kích thước khối con nhỏ nhất 8x8 được sử dụng thay cho EIF.

Theo ví dụ khác, nếu khối afin không thể sử dụng bù chuyển động EIF, bù chuyển động

tính tiến với vectơ chuyển động được tính cho trung tâm của khối afin được sử dụng thay cho EIF.

Theo ví dụ khác, nếu khối afin không thể sử dụng bù chuyển động EIF, hộp giới hạn được dẫn ra dựa vào W' và H' cho các bất đẳng thức (2-8) này được thỏa mãn. Ví dụ W' và H' có thể được tính là:

$$W' = \sqrt{T * W * H} - 1$$

$$H' = \sqrt{T * W * H} - 1$$

Sau đó trong suốt thời gian tính toán MV trong EIF theo phương trình (1-1), phần theo chiều ngang của các vectơ chuyển động được cắt tới khoảng $[mv_{0x}, mv_{0x} + W' - 1]$, và phần theo chiều dọc của các vectơ chuyển động được cắt tới khoảng $[mv_{0y}, mv_{0y} + H' - 1]$. Theo một ví dụ, nếu tỷ lệ khía cạnh khối là k ($W=kH$), H' được tính bằng cách giải phương trình

$$k(H')^2 + (k + 1)H' + 1 - TWH = 0$$

và thực hiện các bước sau đây: $H' = \text{Floor}(H')$, $W' = kH'$.

Theo một ví dụ khác các bước 3.f.i và 3.f.iv.1 được thực hiện như sau:

Tính toán mức tiêu thụ truy cập bộ nhớ cho khối afin trong trường hợp sử dụng EIF

Các bước sau đây được thực hiện cho việc tính toán mức tiêu thụ truy cập bộ nhớ.

1. Dẫn ra vị trí của mỗi mẫu góc của khối con $W \times H$.
2. Dẫn ra vị trí của mỗi mẫu góc của khối con được sử dụng trên bước 3 của EIF (biểu thị nó là khối con trung gian EIF).
3. Dẫn ra các vectơ chuyển động cho mỗi mẫu góc của khối con trung gian EIF.
4. Dẫn ra vị trí của khối con được biến đổi trong ảnh tham chiếu.
5. Dẫn ra kích thước hộp giới hạn cho khối con được biến đổi.
6. Nhận mức tiêu thụ truy cập bộ nhớ dựa vào kích thước hộp giới hạn đối với kích thước khối con được biến đổi và độ dài bộ lọc (EIF sử dụng nội suy song tuyến tính, vì vậy độ dài bộ lọc bằng 2).

Chi tiết về cách thực hiện của các bước này được mô tả dưới đây.

Bước 1. Dẫn ra vị trí của mỗi mẫu góc của khối con afin

Biểu thị (x_0, y_0) là tọa độ của mẫu trên cùng bên trái của khối afin. Theo phương án này đối với việc tính toán mức tiêu thụ truy cập bộ nhớ, giả sử rằng tọa độ của mẫu trên cùng bên trái của khối afin bằng $(1, 1)$. Vị trí (x_0, y_0) không có nghĩa cho việc tính toán mức tiêu thụ truy cập bộ nhớ và với $(x_0, y_0) = (1, 1)$ các phương trình đơn giản hơn.

Sau đó vị trí của khối afin có thể được mô tả bởi các tọa độ của các mẫu góc của nó (trên cùng bên trái, trên cùng bên phải, dưới cùng bên trái, dưới cùng bên phải):

$$\begin{cases} (x_0, y_0) = (1, 1) \\ (x_1, y_1) = (W, 1) \\ (x_2, y_2) = (1, H) \\ (x_3, y_3) = (W, H) \end{cases} \quad (3-1)$$

Bước 2. Dẫn ra vị trí của mỗi mẫu góc của khối con trung gian EIF

Do EIF sử dụng bộ lọc 3 nhánh ở bước 3, nên nội suy song tuyến tính ở bước 2 của EIF được thực hiện cho khối con $(W+2) \times (H+2)$ (một lẻ điểm ảnh từ mỗi biên được bỏ sung). Khối con $(W+2) \times (H+2)$ này được biểu thị là khối con EIF trung gian. Các tọa độ của các mẫu góc khối EIF trung gian (trên cùng bên trái, trên cùng bên phải, dưới cùng bên trái, dưới cùng bên phải) là:

$$\begin{cases} (X_0, Y_0) = (0, 0) \\ (X_1, Y_1) = (1 + W, 0) \\ (X_2, Y_2) = (0, 1 + H) \\ (X_3, Y_3) = (1 + W, 1 + H) \end{cases} \quad (3-2)$$

Các tọa độ của các góc của khối con afin và của khối con EIF trung gian được mô tả trên Fig 8.

Bước 3. Dẫn ra các vector chuyển động cho mỗi mẫu góc của khối con trung gian EIF

Vector chuyển động ban đầu (mv_{0x}, mv_{0y}) không có nghĩa cho việc tính toán mức tiêu thụ truy cập bộ nhớ và với $(mv_{0x}, mv_{0y}) = (dHorX + dVerX, dHorY + dVerY)$ các phương trình đơn giản hơn.

Các vector chuyển động được dẫn ra theo phương trình (1-1).

$$\begin{cases} V_0 = V_{X_0, Y_0} = (0, 0) \\ V_1 = V_{X_1, Y_1} = ((W + 1)dHorX, (W + 1)dHorY) \\ V_2 = V_{X_2, Y_2} = ((H + 1)dVerX, (H + 1)dVerY) \\ V_3 = V_{X_3, Y_3} = ((W + 1)dHorX + (H + 1)dVerX, (W + 1)dHorY + (H + 1)dVerY) \end{cases} \quad (3-3)$$

Bước 4. Dẫn ra vị trí của khối được biến đổi trong ảnh tham chiếu

Vị trí của khối được biến đổi trong ảnh tham chiếu có thể được mô tả bởi các tọa độ của các mẫu góc của nó (trên cùng bên trái, trên cùng bên phải, dưới cùng bên trái, dưới cùng bên phải):

$$\left\{ \begin{array}{l} (X'_0, Y'_0) = (0,0) \\ (X'_1, Y'_1) = (1 + W + (W + 1)dHorX, (W + 1)dHorY) \\ (X'_2, Y'_2) = ((H + 1)dVerX, 1 + H + (H + 1)dVerY) \\ (X'_3, Y'_3) = (1 + W + (W + 1)dHorX + (H + 1)dVerX, 1 + H + (W + 1)dHorY + (H + 1)dVerY) \end{array} \right. \quad (3-4)$$

Bước 5. Dẫn ra kích thước hộp giới hạn cho khối con được biến đổi

Kích thước hộp giới hạn cho khối con được biến đổi trong ảnh tham chiếu có thể được tính bởi phương trình sau đây, trong đó hàm lớn nhất quay lại trị số lớn nhất của các đối số, hàm nhỏ nhất quay lại trị số nhỏ nhất của các đối số:

$$\begin{cases} W' = \max(X'_0, X'_1, X'_2, X'_3) - \min(X'_0, X'_1, X'_2, X'_3) + 1 \\ H' = \max(Y'_0, Y'_1, Y'_2, Y'_3) - \min(Y'_0, Y'_1, Y'_2, Y'_3) + 1 \end{cases} \quad (3-5)$$

Vị trí của khối con được biến đổi trong ảnh tham chiếu và hộp giới hạn tương ứng được mô tả trên Fig.9.

Theo một ví dụ $W' = \text{Ceil}(W')$, $H' = \text{Ceil}(H')$ được thực hiện sau phương trình (3-5).

Theo một ví dụ khác $W' = \text{Floor}(W')$, $H' = \text{Floor}(H')$ được thực hiện sau phương trình (3-5).

Bước 6. Nhận mức tiêu thụ truy cập bộ nhớ

Mức tiêu thụ truy cập bộ nhớ của khối con affin trong một ảnh tham chiếu có thể được quyết định, bởi kích thước hộp giới hạn cho kích thước khối con được biến đổi và độ dài của bộ lọc nội suy MC cho khối chuyển động affin T' , ví dụ, 2, 4, 6, 8....:

$$Mem = (W' + T' - 1) * (H' + T' - 1) \quad (3-6)$$

Đối với EIF, nội suy song tuyến tính được sử dụng, do đó độ dài bộ lọc là 2 và mức tiêu thụ truy cập bộ nhớ bằng

$$Mem = (W' + 1) * (H' + 1) \quad (3-7)$$

Các giới hạn mô hình chuyển động affin đối với EIF

Biểu thị độ rộng dải bộ nhớ trường hợp kém nhất đích là $T = \frac{S_{wc}}{W * H}$, trong đó W và H lần lượt là độ rộng và các độ cao khối con hiện thời, và S_{wc} là mức tiêu thụ truy cập bộ nhớ được phép lớn nhất cho khối con hiện thời, theo độ rộng dải bộ nhớ trường hợp đích. Để đảm bảo rằng độ rộng dải bộ nhớ EIF không lớn hơn độ rộng dải bộ nhớ trường hợp đích, mức tiêu thụ truy cập bộ nhớ của khối con EIF sẽ bị ràng buộc theo điều kiện sau đây:

$$\frac{(W' + 1) * (H' + 1)}{W * H} \leq T$$

Hoặc

$$(W' + 1) * (H' + 1) \leq T * W * H \quad (3-8)$$

Hoặc

$$(W' + 1) * (H' + 1) \leq S_{wc}$$

Trị số T có thể được định rõ trước trong cả bộ mã hóa và bộ giải mã, hoặc được quy định trong tập hợp thông số của chuỗi video bộ mã hóa/giải mã, ví dụ, các mức chuỗi, mức ảnh, tập hợp thông số mức lát chẳng hạn.

Theo một ví dụ, nếu mức tiêu thụ truy cập bộ nhớ được phép lớn nhất trên mỗi mẫu được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 4x4, sau đó T có thể được dẫn ra như sau, trong đó T là độ dài của bộ lọc nội suy:

$$T = \frac{(4+T'-1)*(4+T'-1)}{4*4} \quad (3-9)$$

Đối với T' bằng 6 giới hạn như sau:

$$(W' + 1) * (H' + 1) \leq (4 + 6 - 1) * (4 + 6 - 1) \quad (3-10)$$

Theo ví dụ khác, nếu mức tiêu thụ truy cập bộ nhớ được phép lớn nhất trên mỗi mẫu được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x8, sau đó T có thể được dẫn ra như sau, trong đó T là độ dài của bộ lọc nội suy:

$$T = \frac{(8+T'-1)*(8+T'-1)}{8*8} \quad (3-11)$$

Theo ví dụ khác, mức tiêu thụ truy cập bộ nhớ được phép lớn nhất của mỗi mẫu có thể khác nhau theo hướng dự đoán của khối hiện thời, nghĩa là: khi khối hiện thời là dự đoán một chiều, sử dụng ngưỡng T_{UNI} , khi khối hiện thời là dự đoán đôi, sử dụng ngưỡng T_{BI} .

Ví dụ, T_{UNI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 4x4, T_{BI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x4, thì:

Theo ví dụ khác, T_{UNI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 4x4, T_{BI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x8, thì:

$$T_{UNI} = \frac{(4+T'-1)*(4+T'-1)}{4*4} \quad (3-12)$$

$$T_{BI} = \frac{(8+T'-1)*(4+T'-1)}{8*4} \quad (3-13)$$

T' theo các ví dụ nêu trên là độ dài của bộ lọc nội suy bù chuyển động (MC) cho khối chuyển động tịnh tiến, ví dụ, 2, 4, 6, 8....

Trị số của T , T_{UNI} và T_{BI} có thể tùy thuộc vào độ rộng và độ cao của khối hiện thời.

Đối với khối afin dự đoán đôi, ràng buộc nêu trên được áp dụng riêng cho cả list0 và list1.

Theo một ví dụ khác mức tiêu thụ truy cập bộ nhớ được tính cho list0 và cho list1 là Mem_0 và Mem_1 và tổng của các thành phần này bị giới hạn. Ví dụ nếu T_{BI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x8 giới hạn sau đây được sử dụng:

$$Mem_0 + Mem_1 \leq 2 * (8 + T' - 1) * (8 + T' - 1) \quad (3-16)$$

Nếu các vector chuyển động của khối afin không thể theo sau các điều kiện ràng buộc (3-8), khối không thể sử dụng bù chuyển động EIF.

Theo một ví dụ nếu khối afin không thể sử dụng bù chuyển động EIF, bù chuyển động khối con với kích thước khối con nhỏ nhất 8x8 được sử dụng thay cho EIF.

Theo một ví dụ khác nếu khối afin không thể sử dụng bù chuyển động EIF, bù chuyển động tịnh tiến với vector chuyển động được tính cho trung tâm của khối afin được sử dụng thay cho EIF.

Theo một ví dụ khác nếu khối afin không thể sử dụng bù chuyển động EIF, hộp giới hạn được dẫn ra dựa vào W' và H' cho các bất đẳng thức (3-8) này được thỏa mãn. Ví dụ W' và H' có thể được tính là:

$$W' = \sqrt{T * W * H} - 1$$

$$H' = \sqrt{T * W * H} - 1$$

Sau đó trong suốt thời gian tính toán MV trong EIF theo phương trình (1-1), phần theo chiều ngang của các vector chuyển động được cắt tới khoảng $[mv_{0x}, mv_{0x} + W' - 1]$ và phần theo chiều dọc của các vector chuyển động được cắt tới khoảng $[mv_{0y}, mv_{0y} + H' - 1]$. Theo một ví dụ nếu tỷ lệ khía cạnh khối là k ($W=kH$), H' được tính bằng cách giải phương trình

$$k(H')^2 + (k + 1)H' + 1 - TWH = 0$$

và thực hiện các bước sau đây: $H' = \text{Floor}(H')$, $W' = kH'$.

Theo một ví dụ bước 3.f.iv.1 được thực hiện như sau:

8.5.4.5 Dẫn ra các thông số cắt cho vector chuyển động afin

Các đầu vào với quy trình xử lý này là:

- vị trí (x_{Cb} , y_{Cb}) trong các bộ phận mẫu đầy đủ,
- hai biến $cbWidth$ và $cbHeight$ xác định độ rộng và độ cao của khối lập mã hiện thời,
- sự thay đổi theo chiều ngang của vector chuyển động dX ,

- sự thay đổi theo chiều dọc của vector chuyển động dY,
- vector chuyển động mvBaseScaled,
- độ rộng của ảnh trong các mẫu pic_width,
- độ cao của ảnh trong các mẫu pic_height,
- cờ mvClipMode xác định loại của các ràng buộc hộp giới hạn.

Các đầu ra của quy trình xử lý này là:

- hor_max, ver_max, hor_min và ver_min mà biểu thị các thành phần theo chiều ngang và theo chiều dọc vector chuyển động được phép lớn nhất và nhỏ nhất.

Vector chuyển động trung tâm mv_center được dẫn ra như sau:

$$\text{mv_center}[0] = (\text{mvBaseScaled}[0] + \text{dX}[0] * (\text{cbWidth} \gg 1) + \text{dY}[0] * (\text{cbHeight} \gg 1)) \quad (8\ 743)$$

$$\text{mv_center}[1] = (\text{mvBaseScaled}[1] + \text{dX}[1] * (\text{cbWidth} \gg 1) + \text{dY}[1] * (\text{cbHeight} \gg 1)) \quad (8\ 743)$$

Quy trình làm tròn đối với các vector chuyển động như được quy định trong khoản 8.5.3.10 được truy xuất với mv_center, rightShift được thiết đặt bằng 5, và leftShift được thiết đặt bằng 0 là các đầu vào và vector chuyển động được làm tròn quay lại là mv_center.

Vector chuyển động mv_center được cắt như sau:

$$\text{mv_center}[0] = \text{Clip3}(-217, 217 - 1, \text{mv_center}[0]) \quad (8\ 686)$$

$$\text{mv_center}[1] = \text{Clip3}(-217, 217 - 1, \text{mv_center}[1]) \quad (8\ 686)$$

Biến smv_hor_min, mv_ver_min, mv_hor_max và mv_ver_max được dẫn ra như sau:

$$\text{mv_hor_min} = \text{mv_center}[0] - \text{deviationA}[\log_2 \text{CbWidth} - 3] \quad (8\ 743)$$

$$\text{mv_ver_min} = \text{mv_center}[1] - \text{deviationA}[\log_2 \text{CbHeight} - 3] \quad (8\ 743)$$

$$\text{mv_hor_max} = \text{mv_center}[0] + \text{deviationB}[\log_2 \text{CbWidth} - 3] \quad (8\ 743)$$

$$\text{mv_ver_max} = \text{mv_center}[1] + \text{deviationB}[\log_2 \text{CbHeight} - 3] \quad (8\ 743)$$

với deviationA và deviationB được quy định với k=0,4 là:

Nếu mvClipMode được thiết đặt bằng 0, các biến deviationA và deviationB được định rõ như sau:

$$\text{deviationA}[k] = \{ \text{Ta1}, \text{Ta2}, \text{Ta3}, \text{Ta4}, \text{Ta5} \},$$

$$\text{deviationB}[k] = \{ \text{Tb1}, \text{Tb2}, \text{Tb3}, \text{Tb4}, \text{Tb5} \}.$$

(Theo một ví dụ, các giá trị Ta1-5, Tb1-5 có thể là như sau

$$\text{deviationA}[k] = \{ 16, 80, 224, 512, 1088 \},$$

$$\text{deviationB}[k] \{ 16, 96, 240, 528, 1104 \}.)$$

Theo cách khác, các biến *deviationA* và *deviationB* được dẫn ra bằng cách truy xuất quy trình xử lý được quy định trong khoản XXX.

Các biến *hor_max_pic*, *ver_max_pic*, *hor_min_pic* và *ver_min_pic* được dẫn ra như sau:

$$\text{hor_max_pic} = (\text{pic_width} + \text{CtbSizeY} - \text{xCb} - \text{cbWidth} + 1) \ll 4 \quad (8\ 743)$$

$$\text{ver_max_pic} = (\text{pic_height} + \text{CtbSizeY} - \text{yCb} - \text{cbHeight} + 1) \ll 4 \quad (8\ 743)$$

$$\text{hor_min_pic} = (-\text{CtbSizeY} - \text{xCb}) \ll 4 \quad (8\ 743)$$

$$\text{ver_min_pic} = (-\text{CtbSizeY} - \text{yCb}) \ll 4 \quad (8\ 743)$$

Đầu ra *hor_max*, *ver_max*, *hor_min* và *ver_min* mà biểu thị các thành phần theo chiều ngang và theo chiều dọc vector chuyển động được phép lớn nhất và nhỏ nhất được dẫn ra như sau:

$$\text{hor_max} = \min(\text{hor_max_pic}, \text{mv_hor_max}) \ll 5 \quad (8\ 743)$$

$$\text{ver_max} = \min(\text{ver_max_pic}, \text{mv_ver_max}) \ll 5 \quad (8\ 743)$$

$$\text{hor_min} = \max(\text{hor_min_pic}, \text{mv_hor_min}) \ll 5 \quad (8\ 743)$$

$$\text{ver_min} = \max(\text{ver_min_pic}, \text{mv_ver_min}) \ll 5 \quad (8\ 743)$$

Ví dụ 1 sử dụng phương án 3 trong MPEG-5/EVC

Theo ví dụ này, kích thước khối con EIF bằng 4x4, số các dòng được nạp lớn nhất trong suốt thời gian xử lý dòng thứ nhất R bằng 3, *dX[0]* tương ứng với *dHorX*, *dX[1]* tương ứng với *dHorY*, *dY[0]* tương ứng với *dVerX* và *dY[1]* tương ứng với *dVerY*.

Các biến *dX[0]*, *dX[1]*, *dY[0]*, *dY[1]* theo độ chính xác 1/512 .

8.5.3.7 Quy trình dẫn ra đối với các mảng vector chuyển động từ các vector chuyển động điểm điều khiển afin

Các đầu vào với quy trình xử lý này là:

- vị trí luma (*xCb*, *yCb*) của mẫu trên cùng bên trái của khối lập mã luma hiện thời liên quan đến mẫu luma trên cùng bên trái của ảnh hiện thời,
- hai biến *cbWidth* và *cbHeight* xác định độ rộng và độ cao của khối lập mã luma,
- số các vector chuyển động điểm điều khiển *numCpMv*,
- các vector chuyển động điểm điều khiển *cpMvLX[cpIdx]*, với *cpIdx* = 0..*numCpMv* – 1 và *X* bằng 0 hoặc 1,
- chỉ số tham chiếu *refIdxLX* và *X* bằng 0 hoặc 1.

Các đầu ra của quy trình xử lý này là:

- số các khối con lập mã luma theo hướng chiều ngang *numSbXLX* và theo hướng

chiều dọc numSbY_{LX} với X bằng 0 hoặc 1,

- kích thước của các khối con lập mã luma theo hướng chiều ngang sizeSbX_{LX} và theo hướng chiều dọc sizeSbY_{LX} với X bằng 0 hoặc 1,
- mảng vector chuyển động khối con luma mvLX[xSbIdx][ySbIdx] với xSbIdx = 0..numSbX_{LX} - 1, ySbIdx = 0 .. NumSbY_{LX} - 1 và X bằng 0 hoặc 1,
- mảng vector chuyển động khối con chroma mvCLX[xSbIdx][ySbIdx] với xSbIdx = 0..numSbX_{LX} - 1, ySbIdx = 0 .. NumSbY_{LX} - 1 và X bằng 0 hoặc 1.

Các biến sizeSbX_{LX}, sizeSbY_{LX}, numSbX_{LX}, numSbY_{LX}, và cờ clipMV được dẫn ra theo 8.5.3.8.

...

8.5.3.8 Quy trình dẫn ra đối với kích thước khối con afin

Các đầu vào với quy trình xử lý này là:

- hai biến cbWidth và cbHeight xác định độ rộng và độ cao của khối lập mã luma,
- số các vector chuyển động điểm điều khiển numCpMv,
- các vector chuyển động điểm điều khiển cpMvLX[cpIdx], với cpIdx = 0..numCpMv - 1 và X bằng 0 hoặc 1.

Các đầu ra của quy trình xử lý này là:

- kích thước của các khối con lập mã luma theo hướng chiều ngang sizeSbX và theo hướng chiều dọc sizeSbY,
- số các khối con lập mã luma theo hướng chiều ngang numSbX và theo hướng chiều dọc numSbY.
- cờ clipMV chỉ báo khối hiện thời được xử lý với EIF với việc cắt MV.

Biến sizeSbX được dẫn ra như sau:

- Biến eifSubblockSize được thiết đặt bằng 4.
- Biến eifCanBeApplied được dẫn ra như sau:
- eifCanBeApplied được thiết đặt bằng đúng
- clipMV được thiết đặt bằng sai
- Các mảng X[i], Y[i] được dẫn ra như sau:
 - $X[0] = 0$
 - $X[1] = (\text{eifSubblockSize} + 1) * (dX[0] + (1 \ll 9))$
 - $X[2] = (\text{eifSubblockSize} + 1) * dY[0]$
 - $X[3] = X[1] + X[2]$

- $Y[0] = 0$
- $Y[1] = (\text{eifSubblockSize} + 1) * dX[1]$
- $Y[2] = (\text{eifSubblockSize} + 1) * (dY[1] + (1 \ll 9))$
- $Y[3] = Y[1] + Y[2]$
- Biến Xmax được thiết đặt bằng giá trị lớn nhất của $X[i]$ với i bằng 0..3
- Biến Xmin được thiết đặt bằng giá trị nhỏ nhất của $X[i]$ với i bằng 0..3
- Biến Ymax được thiết đặt bằng giá trị lớn nhất của $Y[i]$ với i bằng 0..3
- Biến Ymin được thiết đặt bằng giá trị nhỏ nhất của $Y[i]$ với i bằng 0..3
- Biến W được thiết đặt bằng $(X_{\text{max}} - X_{\text{min}} + (1 \ll 9) - 1) \gg 9$
- Biến H được thiết đặt bằng $(Y_{\text{max}} - Y_{\text{min}} + (1 \ll 9) - 1) \gg 9$
- Nếu $(W + 2) * (H + 2)$ lớn hơn 81, biến clipMV được thiết đặt bằng đúng
- Nếu $dY[1]$ nhỏ hơn $(-1) \ll 9$, thì biến eifCanBeApplied bằng sai
- Nếu $(\max(0, dY[1]) + \text{Abs}(dX[1])) * (1 + \text{eifSubblockSize})$ lớn hơn $(1 \ll 9)$

thì biến eifCanBeApplied bằng sai.

– Nếu eifCanBeApplied bằng sai so với các biến sizeSbX và sizeSbY được điều chỉnh như sau:

- $\text{sizeSbX} = \max(8, \text{sizeSbX})$
- $\text{sizeSbY} = \max(8, \text{sizeSbY})$

...

8.5.4 Quy trình giải mã đối với các mẫu liên dự đoán

Các đầu vào với quy trình xử lý này là:

- vị trí luma (xCb, yCb) xác định mẫu trên cùng bên trái của khối lập mã luma hiện thời liên quan đến mẫu luma trên cùng bên trái của ảnh hiện thời,
- hai biến nCbW và nCbH xác định độ rộng và độ cao của khối lập mã luma hiện thời,
- các biến numSbXL0, numSbXL1 và numSbYL0, numSbYL1 xác định số các khối con lập mã luma theo hướng theo chiều ngang và theo chiều dọc,
- các vector chuyển động luma theo độ chính xác mẫu phân số 1/16 $\text{mvL0}[\text{xSbIdx}][\text{ySbIdx}]$ với $\text{xSbIdx} = 0 \dots \text{NumSbXL0} - 1$, và $\text{ySbIdx} = 0 \dots \text{NumSbYL0} - 1$, và $\text{mvL1}[\text{xSbIdx}][\text{ySbIdx}]$ với $\text{xSbIdx} = 0 \dots \text{NumSbXL1} - 1$, và $\text{ySbIdx} = 0 \dots \text{NumSbYL1} - 1$
- các vector chuyển động được tinh chỉnh $\text{refMvL0}[\text{xSbIdx}][\text{ySbIdx}]$ và

$\text{refMvL1}[x\text{SbIdx}][y\text{SbIdx}]$ với $x\text{SbIdx} = 0 \dots \text{NumSbX} - 1$, và $y\text{SbIdx} = 0 \dots \text{NumSbY} - 1$,

- các vector chuyển động chroma theo độ chính xác mẫu phân số 1/32 $\text{mvCL0}[x\text{SbIdx}][y\text{SbIdx}]$ với $x\text{SbIdx} = 0 \dots \text{numSbXL0} - 1$, $y\text{SbIdx} = 0 \dots \text{numSbYL0} - 1$, và $\text{mvCL1}[x\text{SbIdx}][y\text{SbIdx}]$ với $x\text{SbIdx} = 0 \dots \text{numSbXL1} - 1$, $y\text{SbIdx} = 0 \dots \text{numSbYL1} - 1$,
- các chỉ số tham chiếu refIdxL0 và refIdxL1 ,
- các cờ ứng dụng danh mục dự đoán, predFlagL0 , và predFlagL1 ,
- biến dmvrAppliedFlag chỉ báo việc sử dụng DMVR,
- số các vector chuyển động điểm điều khiển numCpMv ,
- các vector chuyển động điểm điều khiển $\text{cpMvL0}[cpIdx]$ với $cpIdx = 0 \dots \text{numCpMv} - 1$ và $\text{cpMvL1}[cpIdx]$ với $cpIdx = 0 \dots \text{numCpMv} - 1$.

Các đầu ra của quy trình xử lý này là:

- predSamplesL mảng $(nCbWL) \times (nCbHL)$ của các mẫu dự đoán luma, trong đó $nCbWL$ và $nCbHL$ được dẫn ra như được quy định dưới đây,
- khi ChromaArrayType không bằng 0, preSamplesCb mảng $(nCbWC) \times (nCbHC)$ của các mẫu dự đoán chroma cho thành phần Cb, trong đó $nCbWC$ và $nCbHC$ được dẫn ra như được quy định dưới đây,
- khi ChromaArrayType không bằng 0, predSamplesCr mảng $(nCbWC) \times (nCbHC)$ của các mẫu phần dư chroma cho thành phần Cr, trong đó $nCbWC$ và $nCbHC$ được dẫn ra như được quy định dưới đây.

Các biến $nCbWL$ và $nCbHL$ lần lượt được thiết đặt bằng $nCbW$ và $nCbH$, và các biến $nCbWC$ và $nCbHC$ lần lượt được thiết đặt bằng $nCbW / \text{SubWidthC}$ và $nCbH / \text{SubHeightC}$.

Để predSamplesL0L và predSamplesL1L là các mảng $(nCbW) \times (nCbH)$ của các trị số mẫu luma được dự đoán và, khi ChromaArrayType không bằng 0, predSampleL0Cb , predSampleL1Cb , predSampleL0Cr , và predSampleL1Cr là các mảng $(nCbW / \text{SubWidthC}) \times (nCbH / \text{SubHeightC})$ của các trị số mẫu chroma được dự đoán.

Đối với X đều bằng 0 và 1, khi predFlagLX bằng 1, phần sau đây áp dụng:

- Ảnh tham chiếu bao gồm mảng hai chiều theo thứ tự refPicLXL của các mẫu luma và, khi ChromaArrayType không bằng 0, hai mảng hai chiều theo thứ tự refPicLXCb và refPicLXCr của các mẫu chroma được dẫn ra bằng cách truy xuất quy trình xử lý được quy

định trong khoản 8.5.4.1 với `refIdxLX` là đầu vào.

– Độ rộng và độ cao của khối con lập mã luma hiện thời `sbWidth`, `sbHeight` được dẫn ra như sau:

$$\text{sbWidth} = \text{nCbW} / \text{numSbXLX} \quad (8\ 701)$$

$$\text{sbHeight} = \text{nCbH} / \text{numSbYLX} \quad (8\ 702)$$

– Nếu `affine_flag` bằng 1 và một trong số các biến `sbWidth`, `sbHeight` nhỏ hơn 8, phần sau đây áp dụng:

– Sự thay đổi theo chiều ngang của vector chuyển động `dX`, sự thay đổi theo chiều dọc của vector chuyển động `dY` và vector chuyển động cơ sở `mvBaseScaled` được dẫn ra bằng cách truy xuất quy trình xử lý được quy định trong khoản 8.5.3.9 với độ rộng khối lập mã luma `nCbW`, độ cao khối lập mã luma `nCbH`, số các vector chuyển động điểm điều khiển `numCpMv` và các vector chuyển động điểm điều khiển `cpMvLX[ cpIdx ]` với `cpIdx = 0..numCpMv - 1` là các đầu vào.

– Mảng `predSamplesLXL` được dẫn ra bằng cách truy xuất quy trình nội suy cho bộ lọc nội suy nâng cao được quy định trong khoản 8.5.4.3 với các vị trí luma (`xSb`, `ySb`), độ rộng khối lập mã luma `nCbW`, độ cao khối lập mã luma `nCbH`, sự thay đổi theo chiều ngang của vector chuyển động `dX`, sự thay đổi theo chiều dọc của vector chuyển động `dY`, vector chuyển động cơ sở `mvBaseScaled`, mảng tham chiếu `refPicLXL`, mẫu `bitDepth` `bitDepthY`, độ rộng ảnh `pic_width_in_luma_samples` và độ cao `pic_height_in_luma_samples` và cờ `clipMV` là các đầu vào.

– Nếu `ChromaArrayType` không bằng 0, phần sau đây áp dụng:

$$\text{mvBaseScaled}[0] = \text{mvBaseScaled}[0] / \text{SubWidthC}$$

(8 703)

$$\text{mvBaseScaled}[1] = \text{mvBaseScaled}[1] / \text{SubHeightC}$$

(8 704)

– Các mảng `predSamplesLXCb`, được dẫn ra bằng cách truy xuất quy trình nội suy cho bộ lọc nội suy nâng cao được quy định trong khoản 8.5.4.3 với các vị trí chroma (`xSb / SubWidthC`, `ySb / SubHeightC`), độ rộng khối lập mã chroma `nCbW / SubWidthC`, độ cao khối lập mã luma `nCbH / SubHeightC`, sự thay đổi theo chiều ngang của vector chuyển động `dX`, sự thay đổi theo chiều dọc của vector chuyển động `dY`, vector chuyển động cơ sở `mvBaseScaled`, mảng tham chiếu `refPicLXCb`, mẫu `bitDepth` `bitDepthC`, độ rộng ảnh `pic_width_in_luma_samples /`

SubWidthC, độ cao ảnh $\text{pic_height_in_luma_samples} / \text{SubHeightC}$ và cờ clipMV là các đầu vào.

– Các mảng predSamplesLXCb , được dẫn ra bằng cách truy xuất quy trình nội suy cho bộ lọc nội suy nâng cao được quy định trong khoản 8.5.4.3 với các vị trí chroma ($x_{Sb} / \text{SubWidthC}$, $y_{Sb} / \text{SubHeightC}$), độ rộng khối lập mã chroma $n_{CbW} / \text{SubWidthC}$, độ cao khối lập mã chroma $n_{CbH} / \text{SubHeightC}$, sự thay đổi theo chiều ngang của vector chuyển động dX , sự thay đổi theo chiều dọc của vector chuyển động dY , vector chuyển động cơ sở $mvBaseScaled$, mảng tham chiếu $refPicLXCb$, mẫu bitDepth bitDepthC , độ rộng ảnh $\text{pic_width_in_luma_samples} / \text{SubWidthC}$, độ cao ảnh $\text{pic_height_in_luma_samples} / \text{SubHeightC}$ và cờ clipMV là các đầu vào.

...

8.5.4.3 Quy trình nội suy đối với bộ lọc nội suy nâng cao

Các đầu vào với quy trình xử lý này là:

- vị trí (x_{Cb} , y_{Cb}) trong các bộ phận mẫu đầy đủ,
- hai biến $cbWidth$ và $cbHeight$ xác định độ rộng và độ cao của khối lập mã hiện thời,
- sự thay đổi theo chiều ngang của vector chuyển động dX ,
- sự thay đổi theo chiều dọc của vector chuyển động dY ,
- vector chuyển động $mvBaseScaled$,
- các mảng mẫu ảnh tham chiếu được lựa chọn $refPicLX$,
- độ sâu bit mẫu bitDepth
- độ rộng của ảnh trong các mẫu pic_width ,
- độ cao của ảnh trong các mẫu pic_height .
- cờ clipMV xác định xem việc cắt MV được áp dụng hay không.

Các đầu ra của quy trình xử lý này là:

- predSamplesLX mảng $(cbWidth) \times (cbHeight)$ của các trị số mẫu dự đoán.

Các hệ số bộ lọc nội suy $T[p]$ cho mỗi vị trí mẫu phân số $1/32$ p bằng x_{Frac} hoặc y_{Frac} được quy định trong bảng 8 15.

Các biến shift0 , shift1 , offset0 , offset1 được dẫn ra như sau:

shift0 được thiết đặt bằng $\text{bitDepth} - 8$, offset0 bằng 0,

shift1 được thiết đặt bằng $12 - \text{shift0}$, offset1 bằng $2^{\text{shift1}} - 1$

Các biến hor_max, ver_max, hor_min và ver_min được dẫn ra bằng cách truy xuất quy trình xử lý được quy định trong 8.5.4.5 với vị trí (xCb, yCb) trong các bộ phận mẫu đầy đủ, hai biến cbWidth và cbHeight xác định độ rộng và độ cao của khối lập mã hiện thời, sự thay đổi theo chiều ngang của vectơ chuyển động dX, sự thay đổi theo chiều dọc của vectơ chuyển động dY, vectơ chuyển động mvBaseScaled, độ rộng của ảnh trong các mẫu pic_width, độ cao của ảnh trong các mẫu pic_height và cờ clipMV là đầu vào, và hor_max, ver_max, hor_min và ver_min là đầu ra.

Đối với $x = -1..CbWidth$ và $y = -1..CbHeight$, phần sau đây áp dụng:

- Vectơ chuyển động mvX được dẫn ra như sau:

$$mvX[0] = (mvBaseScaled[0] + dX[0] * x + dY[0] * y) \quad (8\ 728)$$

$$mvX[1] = (mvBaseScaled[1] + dX[1] * x + dY[1] * y) \quad (8\ 729)$$

$$mvX[0] = Clip3(hor_min, hor_max, mvX[0]) \quad (8\ 730)$$

$$mvX[1] = Clip3(ver_min, ver_max, mvX[1]) \quad (8\ 731)$$

- Các biến xInt, yInt, xFrac và yFrac được dẫn ra như sau:

$$xInt = xCb + (mvX[0] \gg 9) + x \quad (8\ 730)$$

$$yInt = yCb + (mvX[1] \gg 9) + y \quad (8\ 731)$$

$$xFrac = (mvX[0] \gg 4) \& 31 \quad (8\ 732)$$

$$yFrac = (mvX[1] \gg 4) \& 31 \quad (8\ 733)$$

- Các biến A và B được dẫn ra như sau:

$$A = (refPicLX[xInt][yInt] * T[xFrac][0] + \\ + refPicLX[xInt + 1][yInt] * T[xFrac][1] + offset0) \gg shift0 \quad (8\ 736)$$

$$B = (refPicLX[xInt][yInt + 1] * T[xFrac][0] + \\ + refPicLX[xInt + 1][yInt + 1] * T[xFrac][1] + offset0) \gg shift0 \quad (8\ 737)$$

- Trị số mẫu bx,y tương ứng với vị trí (x, y) được dẫn ra như sau:

$$bx,y = (A * T[yFrac][0] + B * T[yFrac][1] + offset1) \gg shift1 \quad (8\ 740)$$

Các hệ số bộ lọc nội suy nâng cao eF[] được quy định là $\{-1, 10, -1\}$.

Các biến shift2, shift3, offset2 và offset3 được dẫn ra như sau:

shift2 được thiết đặt bằng $\max(\text{bit_depth} - 11, 0)$, offset2 bằng $2^{\text{shift2}} - 1$,

shift3 được thiết đặt bằng $(6 - \max(\text{bit_depth} - 11, 0))$, offset3 bằng $2^{\text{shift3}} - 1$,

Đối với $x = 0..CbWidth - 1$ và $y = -1..CbHeight$, phần sau đây áp dụng:

$$hx,y = (eF[0] * bx-1,y + eF[1] * bx,y + eF[2] * bx+1,y + offset2) \gg \\ shift2 \quad (8\ 741)$$

Đối với $x = 0..CbWidth - 1$ và $y = 0..CbHeight - 1$, phần sau đây áp dụng:

- $predSamplesLXL[x][y] = Clip3(0, (1 \ll bitDepth) - 1, (eF[0] * h_{x,y-1} + eF[1] * h_{x,y} + eF[2] * b_{x,y+1} + offset3) \gg shift3)$
- (8 742)

Bảng 1 – Bản mô tả của các hệ số bộ lọc nội suy $T[p]$ cho mỗi vị trí mẫu phân số $1/32$

p

Vị trí mẫu phân số p	Các hệ số bộ lọc nội suy	
	$T[p][0]$	$T[p][1]$
0	64	0
1	62	2
2	60	4
3	58	6
4	56	8
5	54	10
6	52	12
7	50	14
8	48	16
9	46	18
10	44	20
11	42	22
12	40	24
13	38	26
14	36	28
15	34	30
16	32	32
17	30	34
18	28	36
19	26	38
20	24	40
21	22	42
22	20	44
23	18	46
24	16	48
25	14	50
26	12	52
27	10	54
28	8	56
29	6	58

30	4	60
31	2	62

...

8.5.4.5 Dẫn ra các thông số cắt cho vector chuyển động afin

Các đầu vào với quy trình xử lý này là:

- vị trí (xCb, yCb) trong các bộ phận mẫu đầy đủ,
- hai biến cbWidth và cbHeight xác định độ rộng và độ cao của khối lập mã hiện

thời,

- sự thay đổi theo chiều ngang của vector chuyển động dX,
- sự thay đổi theo chiều dọc của vector chuyển động dY,
- vector chuyển động mvBaseScaled,
- độ rộng của ảnh trong các mẫu pic_width,
- độ cao của ảnh trong các mẫu pic_height,
- cờ mvClipMode xác định loại của các ràng buộc hộp giới hạn.

Các đầu ra của quy trình xử lý này là:

- hor_max, ver_max, hor_min và ver_min mà biểu thị các thành phần theo chiều ngang và theo chiều dọc vector chuyển động được phép lớn nhất và nhỏ nhất.

Vector chuyển động trung tâm mv_center được dẫn ra như sau:

$$\text{mv_center}[0] = (\text{mvBaseScaled}[0] + \text{dX}[0] * (\text{cbWidth} \gg 1) + \text{dY}[0] * (\text{cbHeight} \gg 1)) \quad (8\ 743)$$

$$\text{mv_center}[1] = (\text{mvBaseScaled}[1] + \text{dX}[1] * (\text{cbWidth} \gg 1) + \text{dY}[1] * (\text{cbHeight} \gg 1)) \quad (8\ 743)$$

Quy trình làm tròn đối với các vector chuyển động như được quy định trong khoản 8.5.3.10 được truy xuất với mv_center, rightShift được thiết đặt bằng 5, và leftShift được thiết đặt bằng 0 là các đầu vào và vector chuyển động được làm tròn quay lại là mv_center.

Vector chuyển động mv_center được cắt như sau:

$$\text{mv_center}[0] = \text{Clip3}(-217, 217 - 1, \text{mv_center}[0]) \quad (8\ 686)$$

$$\text{mv_center}[1] = \text{Clip3}(-217, 217 - 1, \text{mv_center}[1]) \quad (8\ 686)$$

Biến smv_hor_min, mv_ver_min, mv_hor_max và mv_ver_max được dẫn ra như sau:

$$\text{mv_hor_min} = \text{mv_center}[0] - \text{deviationA}[\log_2 \text{CbWidth} - 3] \quad (8\ 743)$$

$$\text{mv_ver_min} = \text{mv_center}[1] - \text{deviationA}[\log_2 \text{CbHeight} - 3] \quad (8\ 743)$$

$$\text{mv_hor_max} = \text{mv_center}[0] + \text{deviationB}[\log_2 \text{CbWidth} - 3] \quad (8\ 743)$$

$$\text{mv_ver_max} = \text{mv_center}[1] + \text{deviationB}[\log_2 \text{CbHeight} - 3] \quad (8\ 743)$$

với deviationA và deviationB được quy định với $k=0,4$ là:

Nếu mvClipMode được thiết đặt bằng 0, các biến deviationA và deviationB được định rõ như sau:

$\text{deviationA}[k] = \{ 16, 80, 224, 512, 1088 \},$

$\text{deviationB}[k] = \{ 16, 96, 240, 528, 1104 \}.$

Theo cách khác, các biến deviationA và deviationB được dẫn ra bằng cách truy xuất quy trình xử lý được quy định trong khoản XXX.

Các biến hor_max_pic, ver_max_pic, hor_min_pic và ver_min_pic được dẫn ra như sau:

$\text{hor_max_pic} = (\text{pic_width} + \text{CtbSizeY} - \text{xCb} - \text{cbWidth} + 1) \ll 4 \quad (8\ 743)$

$\text{ver_max_pic} = (\text{pic_height} + \text{CtbSizeY} - \text{yCb} - \text{cbHeight} + 1) \ll 4 \quad (8\ 743)$

$\text{hor_min_pic} = (-\text{CtbSizeY} - \text{xCb}) \ll 4 \quad (8\ 743)$

$\text{ver_min_pic} = (-\text{CtbSizeY} - \text{yCb}) \ll 4 \quad (8\ 743)$

Đầu ra hor_max, ver_max, hor_min và ver_min mà biểu thị các thành phần theo chiều ngang và theo chiều dọc vectơ chuyển động được phép lớn nhất và nhỏ nhất được dẫn ra như sau:

$\text{hor_max} = \min(\text{hor_max_pic}, \text{mv_hor_max}) \ll 5 \quad (8\ 743)$

$\text{ver_max} = \min(\text{ver_max_pic}, \text{mv_ver_max}) \ll 5 \quad (8\ 743)$

$\text{hor_min} = \max(\text{hor_min_pic}, \text{mv_hor_min}) \ll 5 \quad (8\ 743)$

$\text{ver_min} = \max(\text{ver_min_pic}, \text{mv_ver_min}) \ll 5 \quad (8\ 743)$

Phương án 4

Thuật toán 1. Sử dụng thích nghi EIF và bù chuyển động affin trên cơ sở khối (thuật toán cơ bản)

Thuật toán cơ bản của việc sử dụng thích nghi EIF và bù chuyển động khối con affin như sau.

1. Tính kích thước khối con tốt nhất $M \times N$ dựa vào các thông số mô hình chuyển động affin.
2. Nếu cả độ rộng khối con tốt nhất M và độ cao khối con tốt nhất N lớn hơn hoặc bằng 8, thì thực hiện việc bù chuyển động khối con với khối con $M \times N$.
3. Theo cách khác, kiểm tra các điều kiện có khả năng ứng dụng EIF
 - a. Điều kiện có khả năng ứng dụng EIF 1
 - b. Điều kiện có khả năng ứng dụng EIF 2

c. ...

d. Điều kiện có khả năng ứng dụng EIF P

e. Nếu tất cả các điều kiện có khả năng ứng dụng EIF được đáp ứng, thì thực hiện việc bù chuyển động EIF. Quy trình bù chuyển động EIF bao gồm các bước:

- i. Kiểm tra điều kiện có khả năng ứng dụng EIF P+1
 - ii. Kiểm tra điều kiện có khả năng ứng dụng EIF P+2
 - iii. ...
 - iv. Kiểm tra điều kiện có khả năng ứng dụng EIF P+K
 - v. Nếu một trong số các điều kiện có khả năng ứng dụng EIF P+1 .. P+K không được đáp ứng, thì
 - tính khoảng vector chuyển động thứ nhất mà đưa ra việc hoàn tất của các điều kiện mà đã không được đáp ứng.
 - thiết đặt khoảng vector chuyển động thứ hai bằng khoảng vector chuyển động thứ nhất
 - vi. Theo cách khác
 - tính khoảng vector chuyển động thứ ba
 - thiết đặt khoảng vector chuyển động thứ hai bằng khoảng vector chuyển động thứ ba
 - vii. Cắt các vector chuyển động được tính theo chế độ afin để đảm bảo, rằng các vector này trong khoảng vector chuyển động thứ hai.
- f. Theo cách khác thiết đặt $M = \max(M, 8)$, $N = \max(N, 8)$ và thực hiện việc bù chuyển động khối con với khối con $M \times N$.

Theo một số ví dụ P và K có thể bằng không mà nghĩa là thuật toán nêu trên sẽ không có các bước 3.a – 3.d nếu P bằng không và sẽ không có các bước 3.e.i – 3.e.v nếu K bằng không.

Chi tiết về một số bước của thuật toán cơ bản này được mô tả dưới đây.

Bước 1. Tính toán kích thước khối con tốt nhất

Một phương pháp dẫn ra kích thước khối con afin dựa vào các độ chênh lệch vector chuyển động của các điểm điều khiển afin và độ rộng và độ cao của khối afin. Kích thước khối con $M \times N$ có thể được dẫn ra bởi phương trình (2-1), trong đó Mv_{Pre} là độ chính xác vector chuyển động (ví dụ, 1/4 điểm ảnh trong chuẩn HEVC hoặc 1/16 điểm ảnh trong

các chuẩn VVC và EVC), và các thông số mô hình chuyển động afin dHorX, dHorY, dVerX, dVerY được tính theo các phương trình (1-4) – (1-7) đối với mô hình 6 thông số và theo các phương trình (1-8) – (1-11) đối với mô hình 4 thông số.

$$\begin{cases} M = \text{clip3}(4, w, \frac{w \times MvPre}{\max(\text{abs}(dHorX), \text{abs}(dHorY))}) \\ N = \text{clip3}(4, h, \frac{h \times MvPre}{\max(\text{abs}(dVerX), \text{abs}(dVerY))}) \end{cases} \quad (2-1)$$

M và N được dẫn ra theo phương trình (2-1) sẽ được điều chỉnh xuống nếu cần để đảm bảo rằng w và h lần lượt có thể phân chia được bởi M và N.

Phương pháp khác là xây dựng bảng tra cứu 3 chiều, sau đó nhận kích thước khối con trực tiếp từ bảng tra cứu theo độ chênh lệch vector chuyển động, kích thước khối, và độ chính xác vector chuyển động. Ví dụ, thiết đặt M tới Table_M[x][y][z], trong đó x bằng $\max(\text{abs}(v_{1x} - v_{0x}), \text{abs}(v_{1y} - v_{0y}))$, y bằng độ rộng khối afin, z bằng độ chính xác vector chuyển động; thiết đặt N tới Table_N[x][y][z], trong đó x bằng $\max(\text{abs}(v_{2x} - v_{0x}), \text{abs}(v_{2y} - v_{0y}))$, y bằng độ cao khối afin, z bằng độ chính xác vector chuyển động.

Theo một số cách thực hiện bước 1 được thực hiện như sau:

8.5.3.8 Quy trình dẫn ra đối với kích thước khối con afin

Các đầu vào với quy trình xử lý này là:

- hai biến cbWidth và cbHeight xác định độ rộng và độ cao của khối lập mã luma,
- số các vector chuyển động điểm điều khiển numCpMv,
- các vector chuyển động điểm điều khiển cpMvLX[cpIdx], với cpIdx = 0..numCpMv – 1 và X bằng 0 hoặc 1,
- các cờ ứng dụng danh mục dự đoán predFlagLX, với X bằng 0 hoặc 1.

Các đầu ra của quy trình xử lý này là:

- kích thước của các khối con lập mã luma theo hướng chiều ngang sizeSbX và theo hướng chiều dọc sizeSbY,

sizeSbX được thiết đặt bằng cbWidth, sizeSbY được thiết đặt bằng cbHeight.

Khi predFlagLX bằng 1, phần sau đây áp dụng với X bằng 0 và 1:

- Sự thay đổi theo chiều ngang của vector chuyển động dX, sự thay đổi theo chiều dọc của vector chuyển động dY và vector chuyển động cơ sở mvBaseScaled được dẫn ra bằng cách truy xuất quy trình xử lý được quy định trong khoản 8.5.3.9 với độ rộng khối lập mã luma cbWidth, độ cao khối lập mã luma cbHeight, số các vector chuyển

động điểm điều khiển numCpMv và các vector chuyển động điểm điều khiển cpMvLX[cpIdx] với cpIdx = 0..numCpMv – 1 là các đầu vào.

- Biến mvWx và mvWy được dẫn ra như sau:
 - $mvWx = \max(\text{abs}(dX[0]), \text{abs}(dX[1]))$
 - $mvWy = \max(\text{abs}(dY[0]), \text{abs}(dY[1]))$
- Biến sizeSbXTemp được quy định trong bảng 8-5 theo trị số của mvWx.
- Biến sizeSbYTemp được quy định trong bảng 8-5 theo trị số của mvWy.
- Biến sizeSbX được điều chỉnh như sau:
 - $sizeSbX = \min(sizeSbX, sizeSbXTemp)$
- Biến sizeSbY được điều chỉnh như sau:
 - $sizeSbY = \min(sizeSbY, sizeSbYTemp)$

Bảng 8-5 – Bản mô tả của sizeSbXTemp cho các trị số đầu vào khác nhau của mvWx

mvWx	0	1	2	3	4	> 4
sizeSbX	cbWidth	32	16	8	8	4

Bảng 8-5 – Bản mô tả của sizeSbYTemp cho các trị số đầu vào khác nhau của mvWy

mvWx	0	1	2	3	4	> 4
sizeSbY	cbHeight	32	16	8	8	4

Trong đó khoản 8.5.3.9 được mô tả như sau:

8.5.3.9. Quy trình dẫn ra đối với các thông số mô hình chuyển động afin từ các vector chuyển động điểm điều khiển

Các đầu vào với quy trình xử lý này là:

- hai biến cbWidth và cbHeight xác định độ rộng và độ cao của khối lập mã luma,
- số các vector chuyển động điểm điều khiển numCpMv,
- các vector chuyển động điểm điều khiển cpMvLX[cpIdx], với cpIdx = 0..numCpMv – 1 và X bằng 0 hoặc 1.

Các đầu ra của quy trình xử lý này là:

- sự thay đổi theo chiều ngang của vector chuyển động dX,
- sự thay đổi theo chiều dọc của vector chuyển động dY,
- vector chuyển động mvBaseScaled tương ứng với góc trên cùng bên trái của khối lập mã luma.

Các biến log2CbW (cũng được biểu thị là log2CbWidth) và log2CbH (cũng được biểu thị là log2CbHeight) được dẫn ra như sau:

$$\log2CbW = \text{Log2}(cbWidth)$$

(8-688)

$$\log2CbH = \text{Log2}(cbHeight) \quad (8-689)$$

Sự thay đổi theo chiều ngang của vectơ chuyển động dX được dẫn ra như sau:

$$dX[0] = (cpMvLX[1][0] - cpMvLX[0][0]) << (7 - \log2CbW) \quad (8-690)$$

$$dX[1] = (cpMvLX[1][1] - cpMvLX[0][1]) << (7 - \log2CbW) \quad (8-691)$$

Sự thay đổi theo chiều dọc của vectơ chuyển động dY được dẫn ra như sau:

– Nếu numCpMv bằng 3, dY được dẫn ra như sau:

$$dY[0] = (cpMvLX[2][0] - cpMvLX[0][0]) << (7 - \log2CbH) \quad (8-692)$$

$$dY[1] = (cpMvLX[2][1] - cpMvLX[0][1]) << (7 - \log2CbH) \quad (8-693)$$

– Theo cách khác (numCpMv bằng 2), dY được dẫn ra như sau:

$$dY[0] = -dX[1] \quad (8-694)$$

$$dY[1] = dX[0] \quad (8-695)$$

Vectơ chuyển động mvBaseScaled tương ứng với góc trên cùng bên trái của khối lập mã luma được dẫn ra như sau:

$$mvBaseScaled[0] = cpMvLX[0][0] << 7 \quad (8-696)$$

$$mvBaseScaled[1] = cpMvLX[0][1] << 7 \quad (8-697)$$

Bước 3. Các điều kiện có khả năng ứng dụng EIF

Các điều kiện có khả năng ứng dụng EIF có thể ví dụ như sau

1. Các giới hạn độ rộng dải bộ nhớ. Loại của các giới hạn này đảm bảo rằng kích thước của vùng trong ảnh tham chiếu tương ứng với khối affin hiện thời (khối EIF) không lớn hơn ngưỡng được định rõ trước T. Ví dụ về định nghĩa của vùng trong ảnh tham chiếu tương ứng với khối affin hiện thời được mô tả trên Fig.9.

2. Bộ đệm bên trong được giới hạn bởi các dòng R, trong đó R là trị số được định rõ trước và có thể ví dụ là 3, 4, 5 hoặc lớn hơn. Điều này nghĩa là trong suốt thời gian xử lý của một dòng (một hàng) của khối (khối con) hiện thời không nhiều hơn các dòng R từ ảnh tham chiếu có thể được sử dụng.

3. Truy cập bộ nhớ sẽ là tuần tự, mà nghĩa là nếu đối với dòng thứ i của khối hiện thời dòng thứ j của ảnh tham chiếu được nạp, sau đó đối với dòng thứ (i+1) của khối hiện thời chỉ các dòng j+1, j+2, ... có thể được nạp.

4. Không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất.

5. Các bất đẳng thức cho các thông số mô hình chuyển động affin, ví dụ như sau

$$a. \begin{cases} a \leq dHorX \leq b \\ c \leq dHorY \leq d \\ e \leq dVerX \leq f \\ g \leq dVerY \leq h \end{cases}, \text{ trong đó } a, b, c, d, e, f, g, h \text{ là các trị số được định}$$

rõ trước hoặc âm/dương vô cực.

$$b. \begin{cases} dHorX \leq a * dVerX \\ dVerY \leq b * dHorY \end{cases}, \text{ trong đó } a \text{ và } b \text{ là các trị số được định rõ trước}$$

Theo một ví dụ với các điều kiện có khả năng ứng dụng EIF cụ thể, thuật toán cơ bản như sau:

Thuật toán 2. Sử dụng thích nghi EIF và bù chuyển động afin trên cơ sở khối với các điều kiện có khả năng ứng dụng EIF cụ thể

1. Tính kích thước khối con tốt nhất MxN dựa vào các thông số mô hình chuyển động afin.
2. Nếu cả độ rộng khối con tốt nhất M hoặc độ cao khối con tốt nhất N lớn hơn hoặc bằng 8, thì thực hiện việc bù chuyển động khối con với khối con MxN.
3. Theo cách khác, kiểm tra các điều kiện có khả năng ứng dụng EIF:
 - a. Điều kiện có khả năng ứng dụng EIF 1: Bộ đệm bên trong được giới hạn bởi các dòng R, trong đó R là trị số được định rõ trước và có thể ví dụ là 3, 4, 5 hoặc lớn hơn. Điều này nghĩa là trong suốt thời gian xử lý của một dòng (một hàng) của khối (khối con) hiện thời không nhiều hơn các dòng R từ ảnh tham chiếu có thể được sử dụng.
 - b. Điều kiện có khả năng ứng dụng EIF 2: Truy cập bộ nhớ sẽ là tuần tự, mà nghĩa là nếu đối với dòng thứ i của khối hiện thời dòng thứ j của ảnh tham chiếu được nạp, sau đó đối với dòng thứ (i+1) của khối hiện thời chỉ các dòng j+1, j+2, ... có thể được nạp.
 - c. Điều kiện có khả năng ứng dụng EIF 3: Không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất.
 - d. ...
 - e. Điều kiện có khả năng ứng dụng EIF P
 - f. Nếu tất cả các điều kiện có khả năng ứng dụng EIF được đáp ứng, thì thực hiện việc bù chuyển động EIF. Quy trình bù chuyển động EIF bao gồm các bước:

i. Kiểm tra điều kiện có khả năng ứng dụng EIF P + 1: Kiểm tra các giới hạn độ rộng dải bộ nhớ. Loại của các giới hạn này đảm bảo rằng kích thước của vùng trong ảnh tham chiếu tương ứng với khối afin hiện thời (khối EIF) không lớn hơn ngưỡng được định rõ trước T. Ví dụ về định nghĩa của vùng trong ảnh tham chiếu tương ứng với khối afin hiện thời được mô tả trên Fig.9.

ii. ...

iii. Kiểm tra điều kiện có khả năng ứng dụng EIF P+K

iv. Nếu một trong số các điều kiện có khả năng ứng dụng EIF P+1 .. P+K không được đáp ứng, thì

1. tính khoảng vector chuyển động thứ nhất mà đưa ra việc hoàn tất của các điều kiện mà đã không được đáp ứng.

2. thiết đặt khoảng vector chuyển động thứ hai bằng khoảng vector chuyển động thứ nhất

v. Theo cách khác

1. tính khoảng vector chuyển động thứ ba

2. thiết đặt khoảng vector chuyển động thứ hai bằng khoảng vector chuyển động thứ ba

vi. Cắt các vector chuyển động được tính theo chế độ afin để đảm bảo, rằng các vector này trong khoảng vector chuyển động thứ hai.

g. Theo cách khác thiết đặt $M = \max(M, 8)$, $N = \max(N, 8)$ và thực hiện việc bù chuyển động khối con với khối con $M \times N$.

Theo một số ví dụ P có thể bằng 3 mà nghĩa là thuật toán nêu trên sẽ không có các bước 3.d – 3.e. Theo một số ví dụ K có thể bằng 1 mà nghĩa là thuật toán nêu trên sẽ không có các bước 3.f.ii – 3.f.iii.

Theo một cách thực hiện làm ví dụ các bước 3.a-e được thực hiện như sau:

Khi predFlagLX bằng 1, phần sau đây áp dụng với X bằng 0 và 1:

Sự thay đổi theo chiều ngang của vector chuyển động dX, sự thay đổi theo chiều dọc của vector chuyển động dY và vector chuyển động cơ sở mvBaseScaled được dẫn ra bằng cách truy xuất quy trình xử lý được quy định trong khoản 8.5.3.9 với độ rộng khối lập mã luma cbWidth, độ cao khối lập mã luma cbHeight, số các vector chuyển

động điểm điều khiển numCpMv và các vector chuyển động điểm điều khiển cpMvLX[cpIdx] với cpIdx = 0..numCpMv – 1 là các đầu vào.

Nếu dY[1] nhỏ hơn ((-1) << 9), thì biến eifCanBeAppliedX bằng sai

– Theo cách khác,

–Nếu (max(0, dY[1]) + Abs(dX[1])) * (1 + eifSubblockSize) lớn

hơn (1 << 9) thì biến eifCanBeAppliedX bằng sai.

Trong đó khoản 8.5.3.9 được mô tả như theo các ví dụ nêu trên.

Biến eifCanBeAppliedX bằng đúng ở đây nghĩa là tất cả các điều kiện có khả năng ứng dụng EIF 1 – P (các bước 3.a – 3.e của thuật toán 2) được đáp ứng.

Chi tiết về bước 3.f.i của thuật toán 2 được đưa ra dưới đây.

Tính toán mức tiêu thụ truy cập bộ nhớ cho khối afin trong trường hợp sử dụng EIF

Khi EIF được sử dụng cho việc bù chuyển động của khối afin với kích thước WxH, các bước sau đây được thực hiện cho việc tính toán mức tiêu thụ truy cập bộ nhớ.

1. Dẫn ra vị trí của mỗi mẫu góc của khối afin.

2. Dẫn ra vị trí của mỗi mẫu góc của khối mà được sử dụng trên bước 3 của EIF (biểu thị nó là khối trung gian EIF).

3. Dẫn ra các vector chuyển động cho mỗi mẫu góc của khối trung gian EIF.

4. Dẫn ra vị trí của khối được biến đổi trong ảnh tham chiếu.

5. Dẫn ra kích thước hộp giới hạn cho khối được biến đổi.

6. Nhận mức tiêu thụ truy cập bộ nhớ dựa vào kích thước khối được biến đổi và độ dài bộ lọc (EIF sử dụng nội suy song tuyến tính, vì vậy độ dài bộ lọc bằng 2).

Chi tiết về cách thực hiện của các bước này được mô tả dưới đây.

Bước 1. Dẫn ra vị trí của mỗi mẫu góc của khối afin

Biểu thị (x0, y0) là tọa độ của mẫu trên cùng bên trái của khối afin. Sau đó vị trí của khối afin có thể được mô tả bởi các tọa độ của các mẫu góc của nó (trên cùng bên trái, trên cùng bên phải, dưới cùng bên trái, dưới cùng bên phải):

$$\begin{cases} (x_0, y_0) = (x_0, y_0) \\ (x_1, y_1) = (x_0 + W - 1, y_0) \\ (x_2, y_2) = (x_0, y_0 + H - 1) \\ (x_3, y_3) = (x_0 + W - 1, y_0 + H - 1) \end{cases} \quad (2-1)$$

Bước 2. Dẫn ra vị trí của mỗi mẫu góc của khối trung gian EIF

Nội suy song tuyến tính ở bước 2 của EIF được thực hiện cho khối (W+2)x(H+2) (một lẻ điểm ảnh từ mỗi biên được bổ sung). Khối (W+2)x(H+2) này được biểu thị là khối EIF

trung gian. Các tọa độ của các mẫu góc khối EIF trung gian (trên cùng bên trái, trên cùng bên phải, dưới cùng bên trái, dưới cùng bên phải) là:

$$\begin{cases} (X_0, Y_0) = (x_0 - 1, y_0 - 1) \\ (X_1, Y_1) = (x_0 + W, y_0 - 1) \\ (X_2, Y_2) = (x_0 - 1, y_0 + H) \\ (X_3, Y_3) = (x_0 + W, y_0 + H) \end{cases} \quad (2-2)$$

Các tọa độ của các góc của khối afin và của khối EIF trung gian được mô tả trên Fig 8.

Bước 3. Dẫn ra các vector chuyển động cho mỗi mẫu góc của khối trung gian EIF

Các vector chuyển động được dẫn ra theo phương trình (1-1). Theo một số ví dụ, việc cắt vector chuyển động có thể được sử dụng để đảm bảo rằng các mẫu bên ngoài của ảnh hiện thời với lề của một kích thước CTU không được sử dụng.

$$\begin{cases} V_0 = V_{X_0, Y_0} = (mv_{0x} + (-1)dHorX + (-1)dVerX, mv_{0y} + (-1)dHorY + (-1)dVerY) \\ V_1 = V_{X_1, Y_1} = (mv_{0x} + (W)dHorX + (-1)dVerX, mv_{0y} + (W)dHorY + (-1)dVerY) \\ V_2 = V_{X_2, Y_2} = (mv_{0x} + (-1)dHorX + (H)dVerX, mv_{0y} + (-1)dHorY + (H)dVerY) \\ V_3 = V_{X_3, Y_3} = (mv_{0x} + (W)dHorX + (H)dVerX, mv_{0y} + (W)dHorY + (H)dVerY) \end{cases} \quad (2-3)$$

Bước 4. Dẫn ra vị trí của khối được biến đổi trong ảnh tham chiếu

Lần lượt biểu thị các phần theo chiều ngang và theo chiều dọc của vector V_i là V_{ix} và V_{iy} .

Vị trí của khối được biến đổi trong ảnh tham chiếu có thể được mô tả bởi các tọa độ của các mẫu góc của nó (trên cùng bên trái, trên cùng bên phải, dưới cùng bên trái, dưới cùng bên phải):

$$\begin{cases} (X'_0, Y'_0) = (X_0 + V_{0x}, Y_0 + V_{0y}) \\ (X'_1, Y'_1) = (X_1 + V_{1x}, Y_1 + V_{1y}) \\ (X'_2, Y'_2) = (X_2 + V_{2x}, Y_2 + V_{2y}) \\ (X'_3, Y'_3) = (X_3 + V_{3x}, Y_3 + V_{3y}) \end{cases} \quad (2-4)$$

Bước 5. Dẫn ra kích thước hộp giới hạn cho khối được biến đổi

Kích thước hộp giới hạn cho khối được biến đổi trong ảnh tham chiếu có thể được tính bởi phương trình sau đây, trong đó hàm lớn nhất quay lại trị số lớn nhất của các đối số, hàm nhỏ nhất quay lại trị số nhỏ nhất của các đối số:

$$\begin{cases} W' = \max(X'_0, X'_1, X'_2, X'_3) - \min(X'_0, X'_1, X'_2, X'_3) + 1 \\ H' = \max(Y'_0, Y'_1, Y'_2, Y'_3) - \min(Y'_0, Y'_1, Y'_2, Y'_3) + 1 \end{cases} \quad (2-5)$$

Vị trí của khối được biến đổi trong ảnh tham chiếu và hộp giới hạn tương ứng được mô tả trên Fig.9.

Theo một ví dụ, $W' = \text{Ceil}(W')$, $H' = \text{Ceil}(H')$ được thực hiện sau phương trình (2-5).

Theo ví dụ khác, $W' = \text{Floor}(W')$, $H' = \text{Floor}(H')$ được thực hiện sau phương trình (2-5).

Bước 6. Nhận mức tiêu thụ truy cập bộ nhớ

Mức tiêu thụ truy cập bộ nhớ của khối afin trong một ảnh tham chiếu có thể được quyết định, bởi kích thước hộp giới hạn cho khối được biến đổi và độ dài của bộ lọc nội suy MC cho khối chuyển động afin T' , ví dụ, 2, 4, 6, 8....:

$$Mem = (W' + T' - 1) * (H' + T' - 1) \quad (2-6)$$

Đối với EIF, nội suy song tuyến tính được sử dụng, do đó độ dài bộ lọc là 2 và mức tiêu thụ truy cập bộ nhớ bằng

$$Mem = (W' + 1) * (H' + 1) \quad (2-7)$$

Các giới hạn mô hình chuyển động afin đối với EIF

Biểu thị độ rộng dải bộ nhớ trường hợp kém nhất đích là $T = \frac{S_{wc}}{W * H}$, trong đó W và H lần lượt là độ rộng và các độ cao khối hiện thời, và S_{wc} là mức tiêu thụ truy cập bộ nhớ được phép lớn nhất cho khối hiện thời, theo độ rộng dải bộ nhớ trường hợp đích. Để đảm bảo rằng độ rộng dải bộ nhớ EIF không lớn hơn độ rộng dải bộ nhớ trường hợp đích, mức tiêu thụ truy cập bộ nhớ của khối afin sẽ bị ràng buộc theo điều kiện sau đây:

$$\frac{(W' + 1) * (H' + 1)}{W * H} \leq T$$

Hoặc

$$(W' + 1) * (H' + 1) \leq T * W * H \quad (2-8)$$

Hoặc

$$(W' + 1) * (H' + 1) \leq S_{wc}$$

Trị số T có thể được định rõ trước trong cả bộ mã hóa và bộ giải mã, hoặc được quy định trong tập hợp thông số của chuỗi video bộ mã hóa/giải mã, ví dụ, các mức chuỗi, mức ảnh, tập hợp thông số mức lát chẳng hạn.

Theo một ví dụ, nếu mức tiêu thụ truy cập bộ nhớ được phép lớn nhất trên mỗi mẫu được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 4x4, sau đó T có thể được dẫn ra như sau, trong đó T là độ dài của bộ lọc nội suy:

$$T = \frac{(4+T'-1)*(4+T'-1)}{4*4} \quad (2-9)$$

Đối với T' bằng 6 giới hạn như sau:

$$(W' + 1) * (H' + 1) \leq (4 + 6 - 1) * (4 + 6 - 1) \quad (2-10)$$

Theo ví dụ khác, nếu mức tiêu thụ truy cập bộ nhớ được phép lớn nhất trên mỗi mẫu được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x8, sau đó T có thể được dẫn ra

như sau, trong đó T là độ dài của bộ lọc nội suy:

$$T = \frac{(8+T' - 1) * (8+T' - 1)}{8 * 8} \quad (2-11)$$

Theo ví dụ khác, mức tiêu thụ truy cập bộ nhớ được phép lớn nhất của mỗi mẫu có thể khác nhau theo hướng dự đoán của khối hiện thời, nghĩa là: khi khối hiện thời là dự đoán một chiều, sử dụng ngưỡng T_{UNI} , khi khối hiện thời là dự đoán đôi, sử dụng ngưỡng T_{BI} .

Ví dụ, T_{UNI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 4×4 , T_{BI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8×4 , thì:

$$T_{UNI} = \frac{(4+T' - 1) * (4+T' - 1)}{4 * 4} \quad (2-12)$$

$$T_{BI} = \frac{(8+T' - 1) * (4+T' - 1)}{8 * 4} \quad (2-13)$$

Theo ví dụ khác, T_{UNI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 4×4 , T_{BI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8×8 , thì:

$$T_{UNI} = \frac{(4+T' - 1) * (4+T' - 1)}{4 * 4} \quad (2-14)$$

$$T_{BI} = \frac{(8+T' - 1) * (8+T' - 1)}{8 * 8} \quad (2-15)$$

T' theo các ví dụ nêu trên là độ dài của bộ lọc nội suy bù chuyển động (MC) cho khối chuyển động tịnh tiến, ví dụ, 2, 4, 6, 8....

Trị số của T , T_{UNI} và T_{BI} có thể tùy thuộc vào độ rộng và độ cao của khối hiện thời.

Đối với khối afin dự đoán đôi, ràng buộc nêu trên được áp dụng riêng cho cả list0 và list1.

Theo một ví dụ khác mức tiêu thụ truy cập bộ nhớ được tính cho list0 và cho list1 là Mem_0 và Mem_1 , và tổng của các thành phần này bị giới hạn. Ví dụ, nếu T_{BI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8×8 , giới hạn sau đây được sử dụng:

$$Mem_0 + Mem_1 \leq 2 * (8 + T' - 1) * (8 + T' - 1) \quad (2-16)$$

Nếu các vector chuyển động của khối afin không thể theo sau các điều kiện ràng buộc (2-8), khối không thể sử dụng bù chuyển động EIF.

Theo một ví dụ, nếu khối afin không thể sử dụng bù chuyển động EIF, bù chuyển động khối con với kích thước khối con nhỏ nhất 8×8 được sử dụng thay cho EIF.

Theo ví dụ khác, nếu khối afin không thể sử dụng bù chuyển động EIF, bù chuyển động tịnh tiến với vector chuyển động được tính cho trung tâm của khối afin được sử dụng thay cho EIF.

Theo ví dụ khác, nếu khối afin không thể sử dụng bù chuyển động EIF, hộp giới hạn được dẫn ra dựa vào W' và H' cho các bất đẳng thức (2-8) này được thỏa mãn. Ví dụ W'

và H' có thể được tính là:

$$W' = \sqrt{T * W * H} - 1$$

$$H' = \sqrt{T * W * H} - 1$$

Sau đó trong suốt thời gian tính toán MV trong EIF theo phương trình (1-1), phần theo chiều ngang của các vector chuyển động được cắt tới khoảng $[mv_{0x}, mv_{0x} + W' - 1]$, và phần theo chiều dọc của các vector chuyển động được cắt tới khoảng $[mv_{0y}, mv_{0y} + H' - 1]$. Theo một ví dụ, nếu tỷ lệ khía cạnh khối là k ($W=kH$), H' được tính bằng cách giải phương trình

$$k(H')^2 + (k+1)H' + 1 - TWH = 0$$

và thực hiện các bước sau đây: $H' = \text{Floor}(H')$, $W' = kH'$.

Theo một ví dụ khác các bước 3.f.i và 3.f.iv.1 được thực hiện như sau:

Tính toán mức tiêu thụ truy cập bộ nhớ cho khối afin trong trường hợp sử dụng EIF

Các bước sau đây được thực hiện cho việc tính toán mức tiêu thụ truy cập bộ nhớ.

7. Dẫn ra vị trí của mỗi mẫu góc của khối con $W \times H$.
8. Dẫn ra vị trí của mỗi mẫu góc của khối con được sử dụng trên bước 3 của EIF (biểu thị nó là khối con trung gian EIF).
9. Dẫn ra các vector chuyển động cho mỗi mẫu góc của khối con trung gian EIF.
10. Dẫn ra vị trí của khối con được biến đổi trong ảnh tham chiếu.
11. Dẫn ra kích thước hộp giới hạn cho khối con được biến đổi.
12. Nhận mức tiêu thụ truy cập bộ nhớ dựa vào kích thước hộp giới hạn đối với kích thước khối con được biến đổi và độ dài bộ lọc (EIF sử dụng nội suy song tuyến tính, vì vậy độ dài bộ lọc bằng 2).

Chi tiết về cách thực hiện của các bước này được mô tả dưới đây.

Bước 1. Dẫn ra vị trí của mỗi mẫu góc của khối con afin

Biểu thị (x_0, y_0) là tọa độ của mẫu trên cùng bên trái của khối afin. Theo phương án này đối với việc tính toán mức tiêu thụ truy cập bộ nhớ, giả sử rằng tọa độ của mẫu trên cùng bên trái của khối afin bằng $(1, 1)$. Vị trí (x_0, y_0) không có nghĩa cho việc tính toán mức tiêu thụ truy cập bộ nhớ và với $(x_0, y_0) = (1, 1)$ các phương trình đơn giản hơn.

Sau đó vị trí của khối afin có thể được mô tả bởi các tọa độ của các mẫu góc của nó (trên cùng bên trái, trên cùng bên phải, dưới cùng bên trái, dưới cùng bên phải):

$$\begin{cases} (x_0, y_0) = (1, 1) \\ (x_1, y_1) = (W, 1) \\ (x_2, y_2) = (1, H) \\ (x_3, y_3) = (W, H) \end{cases} \quad (3-1)$$

Bước 2. Dẫn ra vị trí của mỗi mẫu góc của khối con trung gian EIF

Do EIF sử dụng bộ lọc 3 nhánh ở bước 3, nên nội suy song tuyến tính ở bước 2 của EIF được thực hiện cho khối con $(W+2) \times (H+2)$ (một lẻ điểm ảnh từ mỗi biên được bổ sung). Khối con $(W+2) \times (H+2)$ này được biểu thị là khối con EIF trung gian. Các tọa độ của các mẫu góc khối EIF trung gian (trên cùng bên trái, trên cùng bên phải, dưới cùng bên trái, dưới cùng bên phải) là:

$$\begin{cases} (X_0, Y_0) = (0, 0) \\ (X_1, Y_1) = (1 + W, 0) \\ (X_2, Y_2) = (0, 1 + H) \\ (X_3, Y_3) = (1 + W, 1 + H) \end{cases} \quad (3-2)$$

Các tọa độ của các góc của khối con afin và của khối con EIF trung gian được mô tả trên Fig 8.

Bước 3. Dẫn ra các vector chuyển động cho mỗi mẫu góc của khối con trung gian EIF

Vector chuyển động ban đầu (mv_{0x}, mv_{0y}) không có nghĩa cho việc tính toán mức tiêu thụ truy cập bộ nhớ và với $(mv_{0x}, mv_{0y}) = (dHorX + dVerX, dHorY + dVerY)$ các phương trình đơn giản hơn.

Các vector chuyển động được dẫn ra theo phương trình (1-1).

$$\begin{cases} V_0 = V_{X_0, Y_0} = (0, 0) \\ V_1 = V_{X_1, Y_1} = ((W + 1)dHorX, (W + 1)dHorY) \\ V_2 = V_{X_2, Y_2} = ((H + 1)dVerX, (H + 1)dVerY) \\ V_3 = V_{X_3, Y_3} = ((W + 1)dHorX + (H + 1)dVerX, (W + 1)dHorY + (H + 1)dVerY) \end{cases} \quad (3-3)$$

Bước 4. Dẫn ra vị trí của khối được biến đổi trong ảnh tham chiếu

Vị trí của khối được biến đổi trong ảnh tham chiếu có thể được mô tả bởi các tọa độ của các mẫu góc của nó (trên cùng bên trái, trên cùng bên phải, dưới cùng bên trái, dưới cùng bên phải):

$$\begin{cases} (X'_0, Y'_0) = (0, 0) \\ (X'_1, Y'_1) = (1 + W + (W + 1)dHorX, (W + 1)dHorY) \\ (X'_2, Y'_2) = ((H + 1)dVerX, 1 + H + (H + 1)dVerY) \\ (X'_3, Y'_3) = (1 + W + (W + 1)dHorX + (H + 1)dVerX, 1 + H + (W + 1)dHorY + (H + 1)dVerY) \end{cases} \quad (3-4)$$

Bước 5. Dẫn ra kích thước hộp giới hạn cho khối con được biến đổi

Kích thước hộp giới hạn cho khối con được biến đổi trong ảnh tham chiếu có thể được

tính bởi phương trình sau đây, trong đó hàm lớn nhất quay lại trị số lớn nhất của các đối số, hàm nhỏ nhất quay lại trị số nhỏ nhất của các đối số:

$$\begin{cases} W' = \max(X'_0, X'_1, X'_2, X'_3) - \min(X'_0, X'_1, X'_2, X'_3) + 1 \\ H' = \max(Y'_0, Y'_1, Y'_2, Y'_3) - \min(Y'_0, Y'_1, Y'_2, Y'_3) + 1 \end{cases} \quad (3-5)$$

Vị trí của khối con được biến đổi trong ảnh tham chiếu và hộp giới hạn tương ứng được mô tả trên Fig.9.

Theo một ví dụ $W' = \text{Ceil}(W)$, $H' = \text{Ceil}(H)$ được thực hiện sau phương trình (3-5).

Theo một ví dụ khác $W' = \text{Floor}(W)$, $H' = \text{Floor}(H)$ được thực hiện sau phương trình (3-5).

Bước 6. Nhận mức tiêu thụ truy cập bộ nhớ

Mức tiêu thụ truy cập bộ nhớ của khối con afin trong một ảnh tham chiếu có thể được quyết định, bởi kích thước hộp giới hạn cho kích thước khối con được biến đổi và độ dài của bộ lọc nội suy MC cho khối chuyển động afin T' , ví dụ, 2, 4, 6, 8.....:

$$Mem = (W' + T' - 1) * (H' + T' - 1) \quad (3-6)$$

Đối với EIF, nội suy song tuyến tính được sử dụng, do đó độ dài bộ lọc là 2 và mức tiêu thụ truy cập bộ nhớ bằng

$$Mem = (W' + 1) * (H' + 1) \quad (3-7)$$

Các giới hạn mô hình chuyển động afin đối với EIF

Biểu thị độ rộng dải bộ nhớ trường hợp kém nhất đích là $T = \frac{S_{wc}}{W*H}$, trong đó W và H lần lượt là độ rộng và các độ cao khối con hiện thời, và S_{wc} là mức tiêu thụ truy cập bộ nhớ được phép lớn nhất cho khối con hiện thời, theo độ rộng dải bộ nhớ trường hợp đích. Để đảm bảo rằng độ rộng dải bộ nhớ EIF không lớn hơn độ rộng dải bộ nhớ trường hợp đích, mức tiêu thụ truy cập bộ nhớ của khối con EIF sẽ bị ràng buộc theo điều kiện sau đây:

$$\begin{aligned} \frac{(W' + 1) * (H' + 1)}{W * H} &\leq T \\ \text{hoặc} \\ (W' + 1) * (H' + 1) &\leq T * W * H \\ \text{hoặc} \\ (W' + 1) * (H' + 1) &\leq S_{wc} \end{aligned} \quad (3-8)$$

Trị số T có thể được định rõ trước trong cả bộ mã hóa và bộ giải mã, hoặc được quy định trong tập hợp thông số của chuỗi video bộ mã hóa/giải mã, ví dụ, các mức chuỗi, mức ảnh, tập hợp thông số mức lát chẳng hạn.

Theo một ví dụ, nếu mức tiêu thụ truy cập bộ nhớ được phép lớn nhất trên mỗi mẫu được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 4x4, sau đó T có thể được dẫn ra như sau, trong đó T là độ dài của bộ lọc nội suy:

$$T = \frac{(4+T' - 1) * (4+T' - 1)}{4 * 4} \quad (3-9)$$

Đối với T' bằng 6 giới hạn như sau:

$$(W' + 1) * (H' + 1) \leq (4 + 6 - 1) * (4 + 6 - 1) \quad (3-10)$$

Theo ví dụ khác, nếu mức tiêu thụ truy cập bộ nhớ được phép lớn nhất trên mỗi mẫu được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x8, sau đó T có thể được dẫn ra như sau, trong đó T là độ dài của bộ lọc nội suy:

$$T = \frac{(8+T' - 1) * (8+T' - 1)}{8 * 8} \quad (3-11)$$

Theo ví dụ khác, mức tiêu thụ truy cập bộ nhớ được phép lớn nhất của mỗi mẫu có thể khác nhau theo hướng dự đoán của khối hiện thời, nghĩa là: khi khối hiện thời là dự đoán một chiều, sử dụng ngưỡng T_{UNI} , khi khối hiện thời là dự đoán đôi, sử dụng ngưỡng T_{BI} .

Ví dụ, T_{UNI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 4x4, T_{BI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x4, thì:

$$T_{UNI} = \frac{(4+T' - 1) * (4+T' - 1)}{4 * 4} \quad (3-12)$$

$$T_{BI} = \frac{(8+T' - 1) * (4+T' - 1)}{8 * 4} \quad (3-13)$$

Theo ví dụ khác, T_{UNI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 4x4, T_{BI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x8, thì:

$$T_{UNI} = \frac{(4+T' - 1) * (4+T' - 1)}{4 * 4} \quad (3-14)$$

$$T_{BI} = \frac{(8+T' - 1) * (8+T' - 1)}{8 * 8} \quad (3-15)$$

T' theo các ví dụ nêu trên là độ dài của bộ lọc nội suy bù chuyển động (MC) cho khối chuyển động tịnh tiến, ví dụ, 2, 4, 6, 8....

Trị số của T , T_{UNI} và T_{BI} có thể tùy thuộc vào độ rộng và độ cao của khối hiện thời.

Đối với khối afin dự đoán đôi, ràng buộc nêu trên được áp dụng riêng cho cả list0 và list1.

Theo một ví dụ khác mức tiêu thụ truy cập bộ nhớ được tính cho list0 và cho list1 là Mem_0 và Mem_1 và tổng của các thành phần này bị giới hạn. Ví dụ nếu T_{BI} được định rõ là mức tiêu thụ truy cập bộ nhớ của khối 8x8 giới hạn sau đây được sử dụng:

$$Mem_0 + Mem_1 \leq 2 * (8 + T' - 1) * (8 + T' - 1) \quad (3-16)$$

Nếu các vectơ chuyển động của khối afin không thể theo sau các điều kiện ràng buộc (3-8), khối không thể sử dụng bù chuyển động EIF.

Theo một ví dụ nếu khối afin không thể sử dụng bù chuyển động EIF, bù chuyển động khối con với kích thước khối con nhỏ nhất 8x8 được sử dụng thay cho EIF.

Theo một ví dụ khác nếu khối afin không thể sử dụng bù chuyển động EIF, bù chuyển động tịnh tiến với vectơ chuyển động được tính cho trung tâm của khối afin được sử dụng thay cho EIF.

Theo một ví dụ khác nếu khối afin không thể sử dụng bù chuyển động EIF, hộp giới hạn được dẫn ra dựa vào W' và H' cho các bất đẳng thức (3-8) này được thỏa mãn. Ví dụ W' và H' có thể được tính là:

$$W' = \sqrt{T * W * H} - 1$$

$$H' = \sqrt{T * W * H} - 1$$

Sau đó trong suốt thời gian tính toán MV trong EIF theo phương trình (1-1), phần theo chiều ngang của các vectơ chuyển động được cắt tới khoảng $[mv_{0x}, mv_{0x} + W' - 1]$ và phần theo chiều dọc của các vectơ chuyển động được cắt tới khoảng $[mv_{0y}, mv_{0y} + H' - 1]$. Theo một ví dụ nếu tỷ lệ khía cạnh khối là k ($W=kH$), H' được tính bằng cách giải phương trình

$$k(H')^2 + (k+1)H' + 1 - TWH = 0$$

và thực hiện các bước sau đây: $H' = \text{Floor}(H')$, $W' = kH'$.

Theo một cách thực hiện làm ví dụ các bước 3.f.i và 3.f.iii của thuật toán 2 được thực hiện như sau:

Biến clipMVX được dẫn ra như sau với X bằng 0 và 1:

– clipMVX được thiết đặt bằng sai

Biến eifSubblockSize được thiết đặt bằng 4.

Khi predFlagLX bằng 1, phần sau đây áp dụng với X bằng 0 và 1:

– Sự thay đổi theo chiều ngang của vectơ chuyển động dX , sự thay đổi theo chiều dọc của vectơ chuyển động dY và vectơ chuyển động cơ sở mvBaseScaled được dẫn ra bằng cách truy xuất quy trình xử lý được quy định trong khoản 8.5.3.9 với độ rộng khối lập mã luma cbWidth, độ cao khối lập mã luma cbHeight, số các vectơ chuyển động điểm điều khiển numCpMv và các vectơ chuyển động điểm điều khiển cpMvLX[cpIdx] với cpIdx = 0..numCpMv – 1 là các đầu vào.

– Biến mvWx và mvWy được dẫn ra như sau:

- $mvW_x = \max(\text{abs}(dX[0]), \text{abs}(dX[1]))$
- $mvW_y = \max(\text{abs}(dY[0]), \text{abs}(dY[1]))$
- Biến sizeSbXTemp được quy định trong bảng 8-5 theo trị số của mvW_x.
- Biến sizeSbYTemp được quy định trong bảng 8-5 theo trị số của mvW_y.
- Biến sizeSbX được điều chỉnh như sau:
 - $\text{sizeSbX} = \min(\text{sizeSbX}, \text{sizeSbXTemp})$
- Biến sizeSbY được điều chỉnh như sau:
 - $\text{sizeSbY} = \min(\text{sizeSbY}, \text{sizeSbYTemp})$

Bảng 8-5 – Bản mô tả của sizeSbXTemp cho các trị số đầu vào khác nhau của mvW_x

mvW _x	0	1	2	3	4	> 4
sizeSbX	cbWidth	32	16	8	8	4

Bảng 8-5 – Bản mô tả của sizeSbYTemp cho các trị số đầu vào khác nhau của mvW_y

mvW _y	0	1	2	3	4	> 4
sizeSbY	cbHeight	32	16	8	8	4

- Biến clipMVX được điều chỉnh như sau:
- Các mảng X[i], Y[i] được dẫn ra như sau:
 - $X[0] = 0$
 - $X[1] = (\text{eifSubblockSize} + 1) * (dX[0] + (1 \ll 9))$
 - $X[2] = (\text{eifSubblockSize} + 1) * dY[0]$
 - $X[3] = X[1] + X[2]$
 - $Y[0] = 0$
 - $Y[1] = (\text{eifSubblockSize} + 1) * dX[1]$
 - $Y[2] = (\text{eifSubblockSize} + 1) * (dY[1] + (1 \ll 9))$
 - $Y[3] = Y[1] + Y[2]$
- Biến Xmax được thiết đặt bằng giá trị lớn nhất của X[i] với i bằng 0..3
- Biến Xmin được thiết đặt bằng giá trị nhỏ nhất của X[i] với i bằng 0..3
- Biến Ymax được thiết đặt bằng giá trị lớn nhất của Y[i] với i bằng 0..3
- Biến Ymin được thiết đặt bằng giá trị nhỏ nhất của Y[i] với i bằng 0..3
- Biến W được thiết đặt bằng $(X_{\max} - X_{\min} + (1 \ll 9) - 1) \gg 9$
- Biến H được thiết đặt bằng $(Y_{\max} - Y_{\min} + (1 \ll 9) - 1) \gg 9$
- Nếu $(W + 2) * (H + 2)$ lớn hơn 81, biến clipMVX được thiết đặt bằng đúng

Các biến eifCanBeApplied và clipMV được dẫn ra như sau:

$$\text{clipMV} = \text{clipMV0} \mid \text{clipMV1}$$

Biến clipMV bằng đúng ở đây nghĩa là điều kiện có thể ứng dụng độ rộng dải bộ nhớ EIF không được đáp ứng và việc cắt mv thêm được cần (bước 3.f.iv của thuật toán 2)

Theo một ví dụ các bước 3.f.iv -3.f.v được thực hiện như sau:

8.5.4.5 Dẫn ra các thông số cắt cho vector chuyển động afin

Các đầu vào với quy trình xử lý này là:

- vị trí (xCb, yCb) trong các bộ phận mẫu đầy đủ,
- hai biến cbWidth và cbHeight xác định độ rộng và độ cao của khối lập mã hiện thời,
- sự thay đổi theo chiều ngang của vector chuyển động dX,
- sự thay đổi theo chiều dọc của vector chuyển động dY,
- vector chuyển động mvBaseScaled,
- độ rộng của ảnh trong các mẫu pic_width,
- độ cao của ảnh trong các mẫu pic_height,
- cờ clipMV xác định xem việc cắt MV được áp dụng hay không.

Các đầu ra của quy trình xử lý này là:

- hor_max, ver_max, hor_min và ver_min mà biểu thị các thành phần theo chiều ngang và theo chiều dọc vector chuyển động được phép lớn nhất và nhỏ nhất.

Vector chuyển động trung tâm mv_center được dẫn ra như sau:

$$mv_center[0] = (mvBaseScaled[0] + dX[0] * (cbWidth >> 1) + dY[0] * (cbHeight >> 1)) \quad (8-743)$$

$$mv_center[1] = (mvBaseScaled[1] + dX[1] * (cbWidth >> 1) + dY[1] * (cbHeight >> 1)) \quad (8-743)$$

Quy trình làm tròn đối với các vector chuyển động như được quy định trong khoản 8.5.3.10 được truy xuất với mv_center, rightShift được thiết đặt bằng 5, và leftShift được thiết đặt bằng 0 là các đầu vào và vector chuyển động được làm tròn quay lại là mv_center.

Vector chuyển động mv_center được cắt như sau:

$$mv_center[0] = Clip3(-2^{17}, 2^{17} - 1, mv_center[0]) \quad (8-686)$$

$$mv_center[1] = Clip3(-2^{17}, 2^{17} - 1, mv_center[1]) \quad (8-686)$$

Các biến hor_max_pic, ver_max_pic, hor_min_pic và ver_min_pic được dẫn ra như sau:

$$hor_max_pic = (pic_width + 128 - xCb - cbWidth) << 4 \quad (8-743)$$

$$ver_max_pic = (pic_height + 128 - yCb - cbHeight) << 4 \quad (8-743)$$

$$\text{hor_min_pic} = (-128 - \text{xCb}) \ll 4 \quad (8-743)$$

$$\text{ver_min_pic} = (-128 - \text{yCb}) \ll 4 \quad (8-743)$$

Nếu clipMV bằng sai, sau đó các biến đầu ra hor_max, ver_max, hor_min và ver_min mà biểu thị các thành phần theo chiều ngang và theo chiều dọc vectơ chuyển động được phép lớn nhất và nhỏ nhất được dẫn ra như sau:

$$\text{hor_max} = \text{hor_max_pic} \ll 5 \quad (8-743)$$

$$\text{ver_max} = \text{ver_max_pic} \ll 5 \quad (8-743)$$

$$\text{hor_min} = \text{hor_min_pic} \ll 5 \quad (8-743)$$

$$\text{ver_min} = \text{ver_min_pic}, \ll 5 \quad (8-743)$$

Theo cách khác nếu clipMV bằng đúng, các bước sau đây được áp dụng:

– Các biến mv_hor_min, mv_ver_min, mv_hor_max và mv_ver_max được dẫn ra như sau:

$$\text{mv_hor_min} = \text{mv_center} [0] - \text{deviationMV} [\log_2 \text{CbWidth} - 3] \quad (8-743)$$

$$\text{mv_ver_min} = \text{mv_center} [1] - \text{deviationMV} [\log_2 \text{CbHeight} - 3] \quad (8-743)$$

$$\text{mv_hor_max} = \text{mv_center} [0] + \text{deviationMV} [\log_2 \text{CbWidth} - 3] \quad (8-743)$$

$$\text{mv_ver_max} = \text{mv_center} [1] + \text{deviationMV} [\log_2 \text{CbHeight} - 3] \quad (8-743)$$

Với mảng deviationMV được quy định với $k=0,4$ là $\text{deviationMV}[k] = \{ 64, 128, 272, 560, 1136 \}$.

– Các biến hor_max, ver_max, hor_min và ver_min được dẫn ra như sau:

$$\text{hor_max} = \min(\text{hor_max_pic}, \text{mv_hor_max}) \ll 5 \quad (8-743)$$

$$\text{ver_max} = \min(\text{ver_max_pic}, \text{mv_ver_max}) \ll 5 \quad (8-743)$$

$$\text{hor_min} = \max(\text{hor_min_pic}, \text{mv_hor_min}) \ll 5 \quad (8-743)$$

$$\text{ver_min} = \max(\text{ver_min_pic}, \text{mv_ver_min}) \ll 5 \quad (8-743)$$

Ví dụ 1 sử dụng phương án 4 trong MPEG-5/EVC

Theo ví dụ này, kích thước khối con EIF bằng 4x4, số các dòng được nạp lớn nhất trong suốt thời gian xử lý dòng thứ nhất R bằng 3, dX[0] tương ứng với dHorX, dX[1] tương ứng với dHorY, dY[0] tương ứng với dVerX và dY[1] tương ứng với dVerY.

các biến dX[0], dX[1], dY[0], dY[1] theo độ chính xác 1/512 .

8.5.3.7 Quy trình dẫn ra đối với các mảng vector chuyển động từ các vector chuyển động điểm điều khiển afin

Các đầu vào với quy trình xử lý này là:

- vị trí luma (x_{Cb} , y_{Cb}) của mẫu trên cùng bên trái của khối lập mã luma hiện thời liên quan đến mẫu luma trên cùng bên trái của ảnh hiện thời,
- hai biến $cbWidth$ và $cbHeight$ xác định độ rộng và độ cao của khối lập mã luma,
- số các vector chuyển động điểm điều khiển $numCpMv$,
- các vector chuyển động điểm điều khiển $cpMvLX[cpIdx]$, với $cpIdx = 0..numCpMv - 1$ và X bằng 0 hoặc 1,
- các cờ ứng dụng danh mục dự đoán $predFlagLX$, với X bằng 0 hoặc 1,
- chỉ số tham chiếu $refIdxLX$ và X bằng 0 hoặc 1.

Các đầu ra của quy trình xử lý này là:

- số các khối con lập mã luma theo hướng chiều ngang $numSbX$ và theo hướng chiều dọc $numSbY$,
- kích thước của các khối con lập mã luma theo hướng chiều ngang $sizeSbX$ và theo hướng chiều dọc $sizeSbY$,
- mảng vector chuyển động khối con luma $mvLX[xSbIdx][ySbIdx]$ với $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$ và X bằng 0 hoặc 1,
- mảng vector chuyển động khối con chroma $mvCLX[xSbIdx][ySbIdx]$ với $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$ và X bằng 0 hoặc 1.

Các biến $sizeSbX$, $sizeSbY$, $numSbX$, $numSbY$, và cờ $clipMV$ được dẫn ra theo 8.5.3.8.

Khi $predFlagLX$ bằng 1, phần sau đây áp dụng với X bằng 0 và 1:

- Sự thay đổi theo chiều ngang của vector chuyển động dX , sự thay đổi theo chiều dọc của vector chuyển động dY và vector chuyển động cơ sở $mvBaseScaled$ được dẫn ra bằng cách truy xuất quy trình xử lý được quy định trong khoản 8.5.3.9 với độ rộng khối lập mã luma $cbWidth$, độ cao khối lập mã luma $cbHeight$, số các vector chuyển động điểm điều khiển $numCpMv$ và các vector chuyển động điểm điều khiển $cpMvLX[cpIdx]$ với $cpIdx = 0..numCpMv - 1$ là các đầu vào.

- Đối với $ySbIdx = 0..numSbY - 1$:
 - Đối với $xSbIdx = 0..numSbX - 1$:
 - Vector chuyển động luma $mvLX[xSbIdx][ySbIdx]$ được dẫn ra như sau:

$$xPosSb = sizeSbX * xSbIdx + (sizeSbX >> 1) \quad (8-682)$$

$$yPosSb = sizeSbY * ySbIdx + (sizeSbY >> 1) \quad (8-683)$$

$$mvLX[xSbIdx][ySbIdx][0] = (mvBaseScaled[0] + dX[0] * xPosSb + dY[0] * yPosSb) \quad (8-684)$$

$$mvLX[xSbIdx][ySbIdx][1] = (mvBaseScaled[1] + dX[1] * xPosSb + dY[1] * yPosSb) \quad (8-685)$$

– Quy trình làm tròn đối với các vector chuyển động như được quy định trong khoản 8.5.3.10 được truy xuất với mvX được thiết đặt bằng mvLX[xSbIdx][ySbIdx], rightShift được thiết đặt bằng 5, và leftShift được thiết đặt bằng 0 là các đầu vào và mvLX[xSbIdx][ySbIdx] được làm tròn là đầu ra.

– Các vector chuyển động mvLX[xSbIdx][ySbIdx] được cắt như sau:

$$mvLX[xSbIdx][ySbIdx][0] = Clip3(-2^{17}, 2^{17} - 1, mvLX[xSbIdx][ySbIdx][0]) \quad (8-686)$$

$$mvLX[xSbIdx][ySbIdx][1] = Clip3(-2^{17}, 2^{17} - 1, mvLX[xSbIdx][ySbIdx][1]) \quad (8-687)$$

– Quy trình dẫn ra đối với các vector chuyển động chroma trong khoản 8.5.2.14 được truy xuất với mvLX[xSbIdx][ySbIdx] là đầu vào, và vector chuyển động chroma mvCLX[xSbIdx][ySbIdx] là đầu ra.

8.5.3.8 Quy trình dẫn ra đối với kích thước khối con afin

Các đầu vào với quy trình xử lý này là:

- hai biến cbWidth và cbHeight xác định độ rộng và độ cao của khối lập mã luma,
- số các vector chuyển động điểm điều khiển numCpMv,
- các vector chuyển động điểm điều khiển cpMvLX[cpIdx], với cpIdx = 0..numCpMv – 1 và X bằng 0 hoặc 1,
- các cờ ứng dụng danh mục dự đoán predFlagLX, với X bằng 0 hoặc 1.

Các đầu ra của quy trình xử lý này là:

- kích thước của các khối con lập mã luma theo hướng chiều ngang sizeSbX và theo hướng chiều dọc sizeSbY,
- số các khối con lập mã luma theo hướng chiều ngang numSbX và theo hướng chiều dọc numSbY,
- cờ clipMV chỉ báo loại cắt vector chuyển động đối với các khối được xử lý với EIF.

sizeSbX được thiết đặt bằng cbWidth, sizeSbY được thiết đặt bằng cbHeight.

Các biến eifCanBeAppliedX và clipMVX được dẫn ra như sau với X bằng 0 và 1:

- eifCanBeAppliedX được thiết đặt bằng đúng
- clipMVX được thiết đặt bằng sai

Biến `elfSubblockSize` được thiết đặt bằng 4.

Khi `predFlagLX` bằng 1, phần sau đây áp dụng với `X` bằng 0 và 1:

- Sự thay đổi theo chiều ngang của vector chuyển động `dX`, sự thay đổi theo chiều dọc của vector chuyển động `dY` và vector chuyển động cơ sở `mvBaseScaled` được dẫn ra bằng cách truy xuất quy trình xử lý được quy định trong khoản 8.5.3.9 với độ rộng khối lập mã luma `cbWidth`, độ cao khối lập mã luma `cbHeight`, số các vector chuyển động điểm điều khiển `numCpMv` và các vector chuyển động điểm điều khiển `cpMvLX[ cpIdx ]` với `cpIdx = 0..numCpMv - 1` là các đầu vào.

- Biến `mvWx` và `mvWy` được dẫn ra như sau:
 - $mvWx = \max(\text{abs}(dX[0]), \text{abs}(dX[1]))$
 - $mvWy = \max(\text{abs}(dY[0]), \text{abs}(dY[1]))$
- Biến `sizeSbXTemp` được quy định trong bảng 8-5 theo trị số của `mvWx`.
- Biến `sizeSbYTemp` được quy định trong bảng 8-5 theo trị số của `mvWy`.
- Biến `sizeSbX` được điều chỉnh như sau:
 - $sizeSbX = \min(sizeSbX, sizeSbXTemp)$
- Biến `sizeSbY` được điều chỉnh như sau:
 - $sizeSbY = \min(sizeSbY, sizeSbYTemp)$

Bảng 8-5 – Bản mô tả của `sizeSbXTemp` cho các trị số đầu vào khác nhau của `mvWx`

<code>mvWx</code>	0	1	2	3	4	> 4
<code>sizeSbX</code>	<code>cbWidth</code>	32	16	8	8	4

Bảng 8-5 – Bản mô tả của `sizeSbYTemp` cho các trị số đầu vào khác nhau của `mvWy`

<code>mvWx</code>	0	1	2	3	4	> 4
<code>sizeSbY</code>	<code>cbHeight</code>	32	16	8	8	4

- Các biến `elfCanBeAppliedX` và `clipMVX` được điều chỉnh như sau:
- Các mảng `X[i]`, `Y[i]` được dẫn ra như sau:
 - $X[0] = 0$
 - $X[1] = (\text{elfSubblockSize} + 1) * (dX[0] + (1 \ll 9))$
 - $X[2] = (\text{elfSubblockSize} + 1) * dY[0]$
 - $X[3] = X[1] + X[2]$
 - $Y[0] = 0$
 - $Y[1] = (\text{elfSubblockSize} + 1) * dX[1]$
 - $Y[2] = (\text{elfSubblockSize} + 1) * (dY[1] + (1 \ll 9))$
 - $Y[3] = Y[1] + Y[2]$

- Biến Xmax được thiết đặt bằng giá trị lớn nhất của X[i] với i bằng 0..3
- Biến Xmin được thiết đặt bằng giá trị nhỏ nhất của X[i] với i bằng 0..3
- Biến Ymax được thiết đặt bằng giá trị lớn nhất của Y[i] với i bằng 0..3
- Biến Ymin được thiết đặt bằng giá trị nhỏ nhất của Y[i] với i bằng 0..3
- Biến W được thiết đặt bằng $(Xmax - Xmin + (1 \ll 9) - 1) \gg 9$
- Biến H được thiết đặt bằng $(Ymax - Ymin + (1 \ll 9) - 1) \gg 9$
- Nếu $(W + 2) * (H + 2)$ lớn hơn 81, biến clipMVX được thiết đặt bằng đúng
- Nếu dY[1] nhỏ hơn $(-1) \ll 9$, thì biến eifCanBeAppliedX bằng sai
- Theo cách khác,

– Nếu $(\max(0, dY[1]) + \text{Abs}(dX[1])) * (1 + \text{eifSubblockSize})$ lớn hơn $(1 \ll 9)$ thì biến eifCanBeAppliedX bằng sai.

Các biến eifCanBeApplied và clipMV được dẫn ra như sau:

- $\text{eifCanBeApplied} = \text{eifCanBeApplied0} \& \text{eifCanBeApplied1}$
- $\text{clipMV} = \text{clipMV0} | \text{clipMV1}$

Nếu eifCanBeApplied bằng sai so với các biến sizeSbX và sizeSbY được điều chỉnh như sau:

- $\text{sizeSbX} = \max(8, \text{sizeSbX})$
- $\text{sizeSbY} = \max(8, \text{sizeSbY})$

Số các khối con lập mã luma theo hướng chiều ngang numSbX và theo hướng chiều dọc numSbY được dẫn ra như sau:

- $\text{numSbX} = \text{cbWidth} / \text{sizeSbX}$
- $\text{numSbY} = \text{cbHeight} / \text{sizeSbY}$

8.5.3.9 Quy trình dẫn ra đối với các thông số mô hình chuyển động affin từ các vector chuyển động điểm điều khiển

Các đầu vào với quy trình xử lý này là:

- hai biến cbWidth và cbHeight xác định độ rộng và độ cao của khối lập mã luma,
- số các vector chuyển động điểm điều khiển numCpMv,
- các vector chuyển động điểm điều khiển cpMvLX[cpIdx], với cpIdx = 0..numCpMv – 1 và X bằng 0 hoặc 1.

Các đầu ra của quy trình xử lý này là:

- sự thay đổi theo chiều ngang của vector chuyển động dX,
- sự thay đổi theo chiều dọc của vector chuyển động dY,

– vectơ chuyển động mvBaseScaled tương ứng với góc trên cùng bên trái của khối lập mã luma.

Các biến log2CbW (cũng được biểu thị là log2CbWidth) và log2CbH (cũng được biểu thị là log2CbHeight) được dẫn ra như sau:

$$\log2CbW = \text{Log2}(cbWidth) \quad (8-688)$$

$$\log2CbH = \text{Log2}(cbHeight) \quad (8-689)$$

Sự thay đổi theo chiều ngang của vectơ chuyển động dX được dẫn ra như sau:

$$dX[0] = (cpMvLX[1][0] - cpMvLX[0][0]) \ll (7 - \log2CbW) \quad (8-690)$$

$$dX[1] = (cpMvLX[1][1] - cpMvLX[0][1]) \ll (7 - \log2CbW) \quad (8-691)$$

Sự thay đổi theo chiều dọc của vectơ chuyển động dY được dẫn ra như sau:

– Nếu numCpMv bằng 3, dY được dẫn ra như sau:

$$dY[0] = (cpMvLX[2][0] - cpMvLX[0][0]) \ll (7 - \log2CbH) \quad (8-692)$$

$$dY[1] = (cpMvLX[2][1] - cpMvLX[0][1]) \ll (7 - \log2CbH) \quad (8-693)$$

– Theo cách khác (numCpMv bằng 2), dY được dẫn ra như sau:

$$dY[0] = -dX[1] \quad (8-694)$$

$$dY[1] = dX[0] \quad (8-695)$$

Vectơ chuyển động mvBaseScaled tương ứng với góc trên cùng bên trái của khối lập mã luma được dẫn ra như sau:

$$mvBaseScaled[0] = cpMvLX[0][0] \ll 7 \quad (8-696)$$

$$mvBaseScaled[1] = cpMvLX[0][1] \ll 7 \quad (8-697)$$

8.5.4 Quy trình giải mã đối với các mẫu liên dự đoán

Các đầu vào với quy trình xử lý này là:

- vị trí luma (xCb, yCb) xác định mẫu trên cùng bên trái của khối lập mã luma hiện thời liên quan đến mẫu luma trên cùng bên trái của ảnh hiện thời,
- hai biến nCbW và nCbH xác định độ rộng và độ cao của khối lập mã luma hiện thời,
- các biến numSbX và numSbY xác định số các khối con lập mã luma theo hướng theo chiều ngang và theo chiều dọc,
- các vectơ chuyển động luma theo độ chính xác mẫu phân số 1/16 mvL0[xSbIdx][ySbIdx] với xSbIdx = 0 .. NumSbX – 1, và ySbIdx = 0 .. NumSbY – 1, và mvL1[xSbIdx][ySbIdx] với xSbIdx = 0 .. NumSbX – 1, và ySbIdx = 0 .. NumSbY – 1

- các vector chuyển động được tính chỉnh $\text{refMvL0}[x\text{SbIdx}][y\text{SbIdx}]$ và $\text{refMvL1}[x\text{SbIdx}][y\text{SbIdx}]$ với $x\text{SbIdx} = 0 \dots \text{NumSbX} - 1$, và $y\text{SbIdx} = 0 \dots \text{NumSbY} - 1$,
- các vector chuyển động chroma theo độ chính xác mẫu phân số $1/32$ $\text{mvCL0}[x\text{SbIdx}][y\text{SbIdx}]$ với $x\text{SbIdx} = 0 \dots \text{numSbX} - 1$, $y\text{SbIdx} = 0 \dots \text{numSbY} - 1$, và $\text{mvCL1}[x\text{SbIdx}][y\text{SbIdx}]$ với $x\text{SbIdx} = 0 \dots \text{numSbX} - 1$, $y\text{SbIdx} = 0 \dots \text{numSbY} - 1$,
- các vector chuyển động chroma được tính chỉnh theo độ chính xác mẫu phân số $1/32$ $\text{refMvCL0}[x\text{SbIdx}][y\text{SbIdx}]$ với $x\text{SbIdx} = 0 \dots \text{numSbX} - 1$, $y\text{SbIdx} = 0 \dots \text{numSbY} - 1$, và $\text{refMvCL1}[x\text{SbIdx}][y\text{SbIdx}]$ với $x\text{SbIdx} = 0 \dots \text{numSbX} - 1$, $y\text{SbIdx} = 0 \dots \text{numSbY} - 1$,
- các chỉ số tham chiếu refIdxL0 và refIdxL1 ,
- các cờ ứng dụng danh mục dự đoán, predFlagL0 , và predFlagL1 ,
- biến dmvrAppliedFlag chỉ báo việc sử dụng DMVR,
- số các vector chuyển động điểm điều khiển numCpMv ,
- các vector chuyển động điểm điều khiển $\text{cpMvL0}[cpIdx]$ với $cpIdx = 0 \dots \text{numCpMv} - 1$ và $\text{cpMvL1}[cpIdx]$ với $cpIdx = 0 \dots \text{numCpMv} - 1$,
- cờ clipMV xác định loại cắt vector chuyển động.

Các đầu ra của quy trình xử lý này là:

- predSamplesL mảng $(n\text{CbWL}) \times (n\text{CbHL})$ của các mẫu dự đoán luma, trong đó $n\text{CbWL}$ và $n\text{CbHL}$ được dẫn ra như được quy định dưới đây,
- khi ChromaArrayType không bằng 0, preSamplesCb mảng $(n\text{CbWC}) \times (n\text{CbHC})$ của các mẫu dự đoán chroma cho thành phần Cb, trong đó $n\text{CbWC}$ và $n\text{CbHC}$ được dẫn ra như được quy định dưới đây,
- khi ChromaArrayType không bằng 0, predSamplesCr mảng $(n\text{CbWC}) \times (n\text{CbHC})$ của các mẫu phần dư chroma cho thành phần Cr, trong đó $n\text{CbWC}$ và $n\text{CbHC}$ được dẫn ra như được quy định dưới đây.

Các biến $n\text{CbWL}$ và $n\text{CbHL}$ lần lượt được thiết đặt bằng $n\text{CbW}$ và $n\text{CbH}$, và các biến $n\text{CbWC}$ và $n\text{CbHC}$ lần lượt được thiết đặt bằng $n\text{CbW} / \text{SubWidthC}$ và $n\text{CbH} / \text{SubHeightC}$.

Để predSamplesL0L và predSamplesL1L là các mảng $(n\text{CbW}) \times (n\text{CbH})$ của các trị số mẫu luma được dự đoán và, khi ChromaArrayType không bằng 0, các mảng

predSampleL0Cb , predSampleL1Cb , predSampleL0Cr , và predSampleL1Cr ($\text{nCbW} / \text{SubWidthC}$) $\times$ ($\text{nCbH} / \text{SubHeightC}$) của các trị số mẫu chroma được dự đoán.

Đối với X đều bằng 0 và 1, khi predFlagLX bằng 1, phần sau đây áp dụng:

- Ảnh tham chiếu bao gồm mảng hai chiều theo thứ tự refPicLXL của các mẫu luma và, khi ChromaArrayType không bằng 0, hai mảng hai chiều theo thứ tự refPicLXCb và refPicLXCr của các mẫu chroma được dẫn ra bằng cách truy xuất quy trình xử lý được quy định trong khoản 8.5.4.1 với refIdxLX là đầu vào.

- Độ rộng và độ cao của khối con lập mã luma hiện thời sbWidth , sbHeight được dẫn ra như sau:

$$\text{sbWidth} = \text{nCbW} / \text{numSbX} \quad (8-701)$$

$$\text{sbHeight} = \text{nCbH} / \text{numSbY} \quad (8-702)$$

- Nếu affine_flag bằng 1 và một trong số các biến sbWidth , sbHeight nhỏ hơn 8, phần sau đây áp dụng:

- Sự thay đổi theo chiều ngang của vector chuyển động dX , sự thay đổi theo chiều dọc của vector chuyển động dY và vector chuyển động cơ sở mvBaseScaled được dẫn ra bằng cách truy xuất quy trình xử lý được quy định trong khoản 8.5.3.9 với độ rộng khối lập mã luma nCbW , độ cao khối lập mã luma nCbH , số các vector chuyển động điểm điều khiển numCpMv và các vector chuyển động điểm điều khiển $\text{cpMvLX}[\text{cpIdx}]$ với $\text{cpIdx} = 0..\text{numCpMv} - 1$ là các đầu vào.

- Mảng predSamplesLXL được dẫn ra bằng cách truy xuất quy trình nội suy cho bộ lọc nội suy nâng cao được quy định trong khoản 8.5.4.3 với vị trí luma (xSb , ySb), độ rộng khối lập mã luma nCbW , độ cao khối lập mã luma nCbH , sự thay đổi theo chiều ngang của vector chuyển động dX , sự thay đổi theo chiều dọc của vector chuyển động dY , vector chuyển động cơ sở mvBaseScaled , mảng tham chiếu refPicLXL , mẫu bitDepth bitDepthY , độ rộng ảnh $\text{pic_width_in_luma_samples}$, độ cao $\text{pic_height_in_luma_samples}$, cờ clipMV là các đầu vào.

- Nếu ChromaArrayType không bằng 0, phần sau đây áp dụng:

- Các mảng predSamplesLXCb , được dẫn ra bằng cách truy xuất quy trình nội suy cho bộ lọc nội suy nâng cao được quy định trong khoản 8.5.4.3 với vị trí luma (xSb , ySb), độ rộng khối lập mã luma nCbW , độ cao khối lập mã luma nCbH , sự thay đổi theo chiều ngang của vector chuyển động dX , sự thay

đổi theo chiều dọc của vector chuyển động dY, vector chuyển động cơ sở mvBaseScaled, mảng tham chiếu refPicLXCb, mẫu bitDepth bitDepthC, độ rộng ảnh pic_width_in_luma_samples và độ cao pic_height_in_luma_samples, cờ clipMV và cờ isLuma là các đầu vào.

– Các mảng predSamplesLXC_r, được dẫn ra bằng cách truy xuất quy trình nội suy cho bộ lọc nội suy nâng cao được quy định trong khoản 8.5.4.3 với vị trí luma (xSb, ySb), độ rộng khối lập mã luma nCbW, độ cao khối lập mã luma nCbH, sự thay đổi theo chiều ngang của vector chuyển động dX, sự thay đổi theo chiều dọc của vector chuyển động dY, vector chuyển động cơ sở mvBaseScaled, mảng tham chiếu refPicLXC_r, mẫu bitDepth bitDepthC, độ rộng ảnh pic_width_in_luma_samples và độ cao pic_height_in_luma_samples, cờ clipMV và cờ isLuma là các đầu vào.

[...]

8.5.4.3 Quy trình nội suy đối với bộ lọc nội suy nâng cao

Các đầu vào với quy trình xử lý này là:

- vị trí (xCb, yCb) trong các bộ phận mẫu đầy đủ,
- hai biến cbWidth và cbHeight xác định độ rộng và độ cao của khối lập mã hiện thời,
- sự thay đổi theo chiều ngang của vector chuyển động dX,
- sự thay đổi theo chiều dọc của vector chuyển động dY,
- vector chuyển động mvBaseScaled,
- các mảng mẫu ảnh tham chiếu được lựa chọn refPicLX,
- độ sâu bit mẫu bitDepth
- độ rộng của ảnh trong các mẫu pic_width,
- độ cao của ảnh trong các mẫu pic_height,
- cờ clipMV xác định xem việc cắt MV được áp dụng hay không,
- cờ isLuma xác định xem luma hoặc chroma được xử lý hay không

Các đầu ra của quy trình xử lý này là:

- predSamplesLX mảng (cbWidth)x(cbHeight) của các trị số mẫu dự đoán.

Các hệ số bộ lọc nội suy T[p] cho mỗi vị trí mẫu phân số p bằng xFrac hoặc yFrac được quy định trong bảng 8-15.

Các biến `hor_max`, `ver_max`, `hor_min` và `ver_min` được dẫn ra bằng cách truy xuất quy trình xử lý được quy định trong 8.5.4.5 với vị trí (`xCb`, `yCb`) trong các bộ phận mẫu đầy đủ, hai biến `cbWidth` và `cbHeight` xác định độ rộng và độ cao của khối lập mã hiện thời, sự thay đổi theo chiều ngang của vector chuyển động `dX`, sự thay đổi theo chiều dọc của vector chuyển động `dY`, vector chuyển động `mvBaseScaled`, độ rộng của ảnh trong các mẫu `pic_width`, độ cao của ảnh trong các mẫu `pic_height` và cờ `clipMV` là đầu vào, và `hor_max`, `ver_max`, `hor_min` và `ver_min` là đầu ra.

Nếu `isLuma` bằng sai, các bước sau đây được áp dụng:

- $xCb = xCb / SubWidthC$ (8-728)
- $yCb = yCb / SubHeightC$ (8-728)
- $cbWidth = cbWidth / SubWidthC$ (8-728)
- $cbHeight = cbHeight / SubHeightC$ (8-728)
- $mvBaseScaled[0] = mvBaseScaled[0] / SubWidthC$ (8-728)
- $mvBaseScaled[1] = mvBaseScaled[1] / SubHeightC$ (8-728)
- $hor_min = hor_min / SubWidthC$ (8-728)
- $hor_max = hor_max / SubWidthC$ (8-728)
- $ver_min = ver_min / SubHeightC$ (8-728)
- $ver_max = ver_max / SubHeightC$ (8-728)

Các biến `shift0`, `shift1`, `offset0` và `offset1` được dẫn ra như sau:

`shift0` được thiết đặt bằng `bitDepth – 8`, `offset0` bằng 0,

`shift1` được thiết đặt bằng `12 – shift0`, `offset1` bằng $2^{shift1 - 1}$

Đối với $x = -1..CbWidth$ và $y = -1..CbHeight$, phần sau đây áp dụng:

- vector chuyển động `mvX` được dẫn ra như sau:

$$mvX[0] = (mvBaseScaled[0] + dX[0] * x + dY[0] * y) \quad (8-728)$$

$$mvX[1] = (mvBaseScaled[1] + dX[1] * x + dY[1] * y) \quad (8-729)$$

$$mvX[0] = Clip3(hor_min, hor_max, mvX[0]) \quad (8-730)$$

$$mvX[1] = Clip3(ver_min, ver_max, mvX[1]) \quad (8-731)$$

- các biến `xInt`, `yInt`, `xFrac` và `yFrac` được dẫn ra như sau:

$$xInt = xCb + (mvX[0] \gg 9) + x \quad (8-730)$$

$$yInt = yCb + (mvX[1] \gg 9) + y \quad (8-731)$$

$$xFrac = (mvX[0] \gg 4) \& 31 \quad (8-732)$$

$$yFrac = (mvX[1] \gg 4) \& 31 \quad (8-733)$$

- các biến `A` và `B` được dẫn ra như sau:

$$A = (refPicLX[xInt][yInt] * T[xFrac][0] +$$

$$+ \text{refPicLX}[xInt + 1][yInt] * T[xFrac][1] + \text{offset0}) \gg \text{shift0} \quad (8-736)$$

$$B = (\text{refPicLX}[xInt][yInt + 1] * T[xFrac][0] + \\ + \text{refPicLX}[xInt + 1][yInt + 1] * T[xFrac][1] + \text{offset0}) \gg \text{shift0} \quad (8-737)$$

– trị số mẫu $b_{x,y}$ tương ứng với vị trí (x, y) được dẫn ra như sau:

$$b_{x,y} = (A * T[yFrac][0] + B * T[yFrac][1] + \text{offset1}) \gg \text{shift1} \quad (8-740)$$

Các hệ số bộ lọc nội suy nâng cao $eF[]$ được quy định là $\{-1, 10, -1\}$.

Các biến shift2 , shift3 , offset2 và offset3 được dẫn ra như sau:

shift2 được thiết đặt bằng $\max(\text{bit_depth} - 11, 0)$, offset2 bằng $2^{\text{shift2}-1}$,

shift3 được thiết đặt bằng $(6 - \max(\text{bit_depth} - 11, 0))$, offset3 bằng $2^{\text{shift3}-1}$,

Đối với $x = 0.. \text{CbWidth} - 1$ và $y = -1.. \text{CbHeight}$, phần sau đây áp dụng:

$$h_{x,y} = (eF[0] * b_{x-1,y} + eF[1] * b_{x,y} + eF[2] * b_{x+1,y} + \text{offset2}) \gg \text{shift2} \quad (8-741)$$

Đối với $x = 0.. \text{CbWidth} - 1$ và $y = 0.. \text{CbHeight} - 1$, phần sau đây áp dụng:

$$\text{predSamplesLXL}[x][y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, \\ (eF[0] * h_{x,y-1} + eF[1] * h_{x,y} + eF[2] * b_{x,y+1} + \text{offset3}) \gg \text{shift3}) \quad (8-742)$$

Bảng 8-15 – Bản mô tả của các hệ số bộ lọc nội suy $T[p]$ cho mỗi vị trí mẫu phân số

p

Vị trí mẫu phân số p	Các hệ số bộ lọc nội suy	
	$T[p][0]$	$T[p][1]$
0	64	0
1	62	2
2	60	4
3	58	6
4	56	8
5	54	10
6	52	12
7	50	14

8	48	16
9	46	18
10	44	20
11	42	22
12	40	24
13	38	26
14	36	28
15	34	30
16	32	32
17	30	34
18	28	36
19	26	38
20	24	40
21	22	42
22	20	44
23	18	46
24	16	48
25	14	50
26	12	52
27	10	54
28	8	56

29	6	58
30	4	60
31	2	62

8.5.4.5 Dẫn ra các thông số cắt cho vector chuyển động afin

Các đầu vào với quy trình xử lý này là:

- vị trí (xCb, yCb) trong các bộ phận mẫu đầy đủ,
- hai biến cbWidth và cbHeight xác định độ rộng và độ cao của khối lập mã hiện thời,
- sự thay đổi theo chiều ngang của vector chuyển động dX,
- sự thay đổi theo chiều dọc của vector chuyển động dY,
- vector chuyển động mvBaseScaled,
- độ rộng của ảnh trong các mẫu pic_width,
- độ cao của ảnh trong các mẫu pic_height,
- cờ clipMV xác định xem việc cắt MV được áp dụng hay không.

Các đầu ra của quy trình xử lý này là:

- hor_max, ver_max, hor_min và ver_min mà biểu thị các thành phần theo chiều ngang và theo chiều dọc vector chuyển động được phép lớn nhất và nhỏ nhất.

Vector chuyển động trung tâm mv_center được dẫn ra như sau:

$$mv_center[0] = (mvBaseScaled[0] + dX[0] * (cbWidth >> 1) + dY[0] * (cbHeight >> 1)) \quad (8-743)$$

$$mv_center[1] = (mvBaseScaled[1] + dX[1] * (cbWidth >> 1) + dY[1] * (cbHeight >> 1)) \quad (8-743)$$

Quy trình làm tròn đối với các vector chuyển động như được quy định trong khoản

8.5.3.10 được truy xuất với mv_center, rightShift được thiết đặt bằng 5, và leftShift được thiết đặt bằng 0 là các đầu vào và vector chuyển động được làm tròn quay lại là mv_center.

Vector chuyển động mv_center được cắt như sau:

$$mv_center[0] = Clip3(-2^{17}, 2^{17} - 1, mv_center[0]) \quad (8-686)$$

$$mv_center[1] = Clip3(-2^{17}, 2^{17} - 1, mv_center[1]) \quad (8-686)$$

Các biến hor_max_pic, ver_max_pic, hor_min_pic và ver_min_pic được dẫn ra như

sau:

$$hor_max_pic = (pic_width + 128 - xCb - cbWidth) << 4 \quad (8-743)$$

$$ver_max_pic = (pic_height + 128 - yCb - cbHeight) << 4 \quad (8-743)$$

$$\text{hor_min_pic} = (-128 - \text{xCb}) \ll 4 \quad (8-743)$$

$$\text{ver_min_pic} = (-128 - \text{yCb}) \ll 4 \quad (8-743)$$

Nếu clipMV bằng sai, sau đó các biến đầu ra hor_max, ver_max, hor_min và ver_min mà biểu thị các thành phần theo chiều ngang và theo chiều dọc vectơ chuyển động được phép lớn nhất và nhỏ nhất được dẫn ra như sau:

$$\text{hor_max} = \text{hor_max_pic} \ll 5 \quad (8-743)$$

$$\text{ver_max} = \text{ver_max_pic} \ll 5 \quad (8-743)$$

$$\text{hor_min} = \text{hor_min_pic} \ll 5 \quad (8-743)$$

$$\text{ver_min} = \text{ver_min_pic} \ll 5 \quad (8-743)$$

Theo cách khác nếu clipMV bằng đúng, các bước sau đây được áp dụng:

– Các biến mv_hor_min, mv_ver_min, mv_hor_max và mv_ver_max được dẫn ra như sau:

$$\text{mv_hor_min} = \text{mv_center} [0] - \text{deviationMV} [\log_2 \text{CbWidth} - 3] \quad (8-743)$$

$$\text{mv_ver_min} = \text{mv_center} [1] - \text{deviationMV} [\log_2 \text{CbHeight} - 3] \quad (8-743)$$

$$\text{mv_hor_max} = \text{mv_center} [0] + \text{deviationMV} [\log_2 \text{CbWidth} - 3] \quad (8-743)$$

$$\text{mv_ver_max} = \text{mv_center} [1] + \text{deviationMV} [\log_2 \text{CbHeight} - 3] \quad (8-743)$$

với mảng deviationMV được quy định với $k=0,4$ là $\text{deviationMV}[k] = \{ 64, 128, 272, 560, 1136 \}$.

– Các biến hor_max, ver_max, hor_min và ver_min được dẫn ra như sau:

$$\text{hor_max} = \min(\text{hor_max_pic}, \text{mv_hor_max}) \ll 5 \quad (8-743)$$

$$\text{ver_max} = \min(\text{ver_max_pic}, \text{mv_ver_max}) \ll 5 \quad (8-743)$$

$$\text{hor_min} = \max(\text{hor_min_pic}, \text{mv_hor_min}) \ll 5 \quad (8-743)$$

$$\text{ver_min} = \max(\text{ver_min_pic}, \text{mv_ver_min}) \ll 5 \quad (8-743)$$

Liên quan đến các phương án và các ví dụ nêu trên, với điều kiện “Nếu $(W + 2) * (H + 2)$ lớn hơn 81, biến clipMV được thiết đặt bằng đúng”, trị số 81 có thể được thay thế bởi 72.

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

Fig.10 là sơ đồ khối thể hiện hệ thống cung cấp nội dung 3100 để thực hiện dịch vụ phân phối nội dung. Hệ thống cung cấp nội dung 3100 này bao gồm thiết bị chụp 3102, thiết bị đầu cuối 3106, và một cách tùy chọn bao gồm màn hình 3126. Thiết bị chụp 3102 truyền thông với thiết bị đầu cuối 3106 qua 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 nhưng không giới hạn ở WIFI, Ethernet, cáp, (3G/4G/5G) không dây, USB, hoặc loại kết hợp bất kỳ của chúng, hoặc tương tự.

Thiết bị chụp 3102 tạo ra dữ liệu, và có thể mã hóa dữ liệu bởi phương pháp mã hóa như được thể hiện trong các phương án nêu trên. Theo cách khác, thiết bị chụp 3102 có thể phân phối dữ liệu tới máy chủ tạo dòng (không được thể hiện trên các hình vẽ), và máy chủ mã hóa dữ liệu và truyền dữ liệu được mã hóa tới thiết bị đầu cuối 3106. Thiết bị chụp 3102 bao gồm nhưng không giới hạn ở camera, điện thoại thông minh hoặc máy tính bảng (Pad), máy tính hoặc máy tính xách tay, hệ thống hội nghị video, PDA, thiết bị được lắp trên phương tiện giao thông, hoặc sự kết hợp của bất kỳ trong số chúng, hoặc tương tự. Ví dụ, 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 thực tế có thể thực hiện việc xử lý mã hóa video. Khi dữ liệu bao gồm audio (nghĩa là, giọng nói), bộ mã hóa audio được bao gồm trong thiết bị chụp 3102 thực tế có thể thực hiện việc xử lý mã hóa audio. Đối với một số kịch bản thực tế, thiết bị chụp 3102 phân phối dữ liệu video hoặc audio được mã hóa bằng cách đa hợp chúng cùng nhau. Đối với các kịch bản thực tế khác, ví dụ trong hệ thống hội nghị video, dữ liệu audio được mã hóa và dữ liệu video được mã hóa không được đa hợp. Thiết bị chụp 3102 phân phối dữ liệu audio được mã hóa và dữ liệu video được mã hóa tới thiết bị đầu cuối 3106 một cách riêng biệt.

Trong hệ thống cung cấp nội dung 3100, thiết bị đầu cuối 310 thu và tái tạo dữ liệu được mã hóa. Thiết bị đầu cuối 3106 có thể là thiết bị có khả năng thu và khôi phục dữ liệu, chẳng hạn như điện thoại thông minh hoặc Pad 3108, máy tính hoặc máy tính cầm tay 3110, đầu ghi video mạng (network video recorder, viết tắt là NVR)/ đầu ghi video kỹ thuật số (digital video recorder, viết tắt là DVR) 3112, TV 3114, thiết bị giải mã tín hiệu (set top box, viết tắt là STB) 3116, hệ thống hội nghị video 3118, hệ thống giám sát video 3120, thiết bị hỗ trợ kỹ thuật số cá nhân (personal digital assistant, viết tắt là PDA) 3122, thiết bị được lắp trên phương tiện giao thông 3124, hoặc sự kết hợp của bất kỳ trong số chúng, hoặc tương tự có khả năng giải mã dữ liệu được mã hóa nêu trên. Ví dụ, 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 quyền ưu tiên thực hiện việc 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 quyền ưu tiên thực hiện việc xử lý giải mã audio.

Đối với thiết bị đầu cuối với màn hình của nó, ví dụ, điện thoại thông minh hoặc Pad 3108, máy tính hoặc máy tính cầm tay 3110, đầu ghi video mạng (NVR)/ đầu ghi video kỹ thuật số (DVR) 3112, TV 3114, thiết bị hỗ trợ kỹ thuật số cá nhân (PDA) 3122, hoặc thiết bị được lắp trên phương tiện giao thông 3124, thiết bị đầu cuối có thể nạp dữ liệu được giải mã vào màn hình của nó. Đối với thiết bị đầu cuối được trang bị không có màn hình, ví dụ, STB 3116, hệ thống hội nghị video 3118, hoặc hệ thống giám sát video 3120, màn hình bên ngoài 3126 được tiếp xúc ở đó để thu và thể hiện dữ liệu được giải mã.

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

Fig.11 là sơ đồ thể hiện cấu trúc của ví dụ về thiết bị đầu cuối 3106. Sau khi thiết bị đầu cuối 3106 thu dòng từ thiết bị chụp 3102, bộ phận triển khai giao thức 3202 phân tích giao thức truyền của dòng. Giao thức bao gồm nhưng không giới hạn ở giao thức tạo dòng thời gian thực (Real Time Streaming Protocol, viết tắt là RTSP), giao thức chuyển đổi siêu văn bản (Hyper Text Transfer Protocol, viết tắt là HTTP), giao thức tạo dòng trực tuyến HTTP (HTTP Live streaming, viết tắt là HLS), MPEG-DASH, giao thức vận chuyển thời gian thực (Real-time Transport protocol, viết tắt là RTP), giao thức nhắn tin thời gian thực (Real Time Messaging Protocol, viết tắt là RTMP), hoặc loại kết hợp bất kỳ của chúng, hoặc tương tự.

Sau khi bộ phận triển khai giao thức 3202 xử lý dòng, tệp dòng được tạo ra. Tệp được đưa ra tới bộ phận giải đa hợp 3204. Bộ phận giải đa hợp 3204 có thể tách riêng dữ liệu được đa hợp thành dữ liệu audio được mã hóa và dữ liệu video được mã hóa. Như được nêu trên, đối với một số kịch bản thực tế, ví dụ trong hệ thống hội nghị video, dữ liệu audio được mã hóa và dữ liệu video được mã hóa không được đa hợp. Trong trường hợp này, dữ liệu được mã hóa được truyền tới bộ giải mã video 3206 và bộ giải mã audio 3208 mà không thông qua bộ phận giải đa hợp 3204.

Qua việc xử lý giải đa hợp, dòng sơ cấp (elementary stream, viết tắt là ES) video, ES audio, và một cách tùy chọn phụ đề được tạo ra. Bộ giải mã video 3206, mà bao gồm bộ

giải mã video 30 như được giải thích theo các phương án nêu trên, giải mã ES video bởi phương pháp giải mã như được thể hiện theo các phương án nêu trên để tạo ra khung video, và nạp dữ liệu này vào bộ phận đồng bộ 3212. Bộ giải mã audio 3208, giải mã ES audio để tạo ra khung audio, và nạp dữ liệu này vào bộ phận đồng bộ 3212. Theo cách khác, khung video có thể lưu trữ ở bộ đệm (không được thể hiện trên Fig.23) trước khi nạp nó vào bộ phận đồng bộ 3212. Tương tự như vậy, khung audio có thể lưu trữ ở bộ đệm (không được thể hiện trên Fig.23) trước khi nạp nó vào bộ phận đồng bộ 3212.

Bộ phận đồng bộ 3212 đồng bộ hóa khung video và khung audio, và cấp video/audio tới màn hình video/audio 3214. Ví dụ, bộ phận đồng bộ 3212 đồng bộ hóa sự biểu diễn của thông tin video và audio. Thông tin có thể lập mã ở cú pháp nhờ sử dụng các dấu thời gian liên quan đến sự biểu diễn của dữ liệu trực quan và audio được lập mã và các dấu thời gian liên quan đến việc phân phối của chính dòng dữ liệu.

Nếu phụ đề được bao gồm trong dòng, bộ giải mã phụ đề 3210 giải mã phụ đề, và đồng bộ hóa nó với khung video và khung audio, và cấp video/audio/phụ đề tới màn hình video/audio/phụ đề 3216.

Sáng chế không giới hạn ở hệ thống nêu trên, và hoặc thiết bị mã hóa ảnh hoặc thiết bị giải mã ảnh theo các phương án nêu trên có thể được kết hợp vào hệ thống khác, ví dụ, hệ thống ô tô.

Fig.12 là sơ đồ khối minh họa phương pháp theo khía cạnh thứ nhất của sáng chế. Phương pháp của khía cạnh thứ nhất bao gồm các bước sau đây:

1201: xác định các vector chuyển động điểm điều khiển (CPMV), cho khối được lập mã ở chế độ afin;

1202: xác định vùng tham chiếu trong ảnh tham chiếu tương ứng với khối con của khối được lập mã afin dựa vào các CPMV được xác định;

1203: thiết đặt biến clipMVX bằng đúng nếu kích thước của vùng tham chiếu lớn hơn ngưỡng được định rõ trước, theo cách khác thiết đặt biến clipMVX bằng sai;

1204: dẫn ra trường vector chuyển động dựa trên điểm ảnh cho khối được lập mã afin; và

1205: nếu biến clipMVX bằng đúng, thực hiện cắt vector chuyển động với khoảng vector chuyển động thứ nhất, trong đó khoảng vector chuyển động thứ nhất được xác định dựa vào các CPMV được xác định và kích thước của khối được lập mã afin.

Các vector chuyển động điểm điều khiển (các CPMV) có thể, ví dụ, được nhận dạng trên Fig.9 là các vector ở các góc của khối con được định rõ bởi các điểm không phải cơ sở tới các góc tương ứng của vùng tham chiếu có các điểm cơ sở.

Dựa vào phân mô tả nêu trên của các phương án, với điều kiện “Nếu $(W + 2) * (H + 2)$ lớn hơn 81, biến clipMV được thiết đặt bằng đúng”, các trị số 81 hoặc 72 tương ứng với ngưỡng được định rõ trước của kích thước của vùng tham chiếu.

Ưu điểm của phương pháp này, ví dụ, là việc lọc nội suy nâng cao (Enhanced Interpolation Filtering, viết tắt là EIF), có thể vẫn được sử dụng ngay cả khi ngưỡng độ rộng dải bộ nhớ vượt quá, do bao gồm việc cắt MV, thay vì không sử dụng EIF chút nào, cũng xem phần dưới đây.

Khoảng vector chuyển động thứ nhất có thể được xác định bằng cách xác định vector chuyển động cho điểm trung tâm của khối được lập mã afin; xác định khoảng vector chuyển động thứ hai dựa vào vector chuyển động cho điểm trung tâm của khối được lập mã afin và các dịch vị được định rõ trước cho kích thước khối tương ứng, kích thước khối bao gồm kích thước của khối được lập mã afin. Xác định vector chuyển động cho điểm trung tâm của khối được lập mã afin có thể được thực hiện bởi các phương trình sau đây (xem phần mô tả nêu trên):

$$\begin{aligned} mv_center[0] &= (mvBaseScaled[0] + dX[0] * (cbWidth >> 1) + dY[0] * (cbHeight >> 1)) \\ mv_center[1] &= (mvBaseScaled[1] + dX[1] * (cbWidth >> 1) + dY[1] * (cbHeight >> 1)), \end{aligned}$$

trong đó mvBaseScaled biểu diễn vector chuyển động của góc trên cùng bên trái của khối, cbWidth và cbHeight lần lượt biểu diễn độ rộng và độ cao của khối, dX[0], dX[1] biểu diễn các độ chênh lệch tương ứng của các phần theo chiều ngang và theo chiều dọc của vector chuyển động trên một mẫu theo hướng chiều ngang, và dY[0], dY[1] biểu diễn các độ chênh lệch tương ứng của các phần theo chiều ngang và theo chiều dọc của vector chuyển động trên một mẫu theo hướng chiều dọc; và mv_center[0] và mv_center[1] lần lượt biểu diễn các phần theo chiều ngang và theo chiều dọc của vector chuyển động cho điểm trung tâm.

Việc xác định khoảng vector chuyển động thứ hai dựa vào vector chuyển động cho điểm trung tâm của khối được lập mã afin và các dịch vị được định rõ trước cho kích thước khối tương ứng có thể được thực hiện nhờ sử dụng các phương trình sau đây:

$$\begin{aligned} mv_hor_min &= mv_center[0] - deviationMV[\log_2 CbWidth - 3] \\ mv_ver_min &= mv_center[1] - deviationMV[\log_2 CbHeight - 3] \end{aligned}$$

$$mv_hor_max = mv_center[0] + deviationMV[\log2CbWidth - 3]$$

$$mv_ver_max = mv_center[1] + deviationMV[\log2CbHeight - 3]$$

trong đó $deviationMV[]$ biểu diễn bảng của các dịch vị được định rõ trước cho kích thước khối tương ứng, và mv_hor_min , mv_ver_min , mv_hor_max và mv_ver_max biểu diễn khoảng vector chuyển động thứ hai. Các dịch vị được định rõ trước cho các kích thước khối tương ứng có thể tỷ lệ với { 64, 128, 272, 560, 1136 }, trong đó 64 tương ứng với kích thước khối có kích thước tương ứng bằng 8, 128 tương ứng với kích thước khối có kích thước tương ứng bằng 16, 272 tương ứng với kích thước khối có kích thước tương ứng bằng 32, 560 tương ứng với kích thước khối có kích thước tương ứng bằng 64, hoặc 1136 tương ứng với kích thước khối có kích thước tương ứng bằng 128.

khoảng vector chuyển động thứ nhất có thể được định tỷ lệ theo các thông số lấy mẫu phụ $SubWidthC$ và $SubHeightC$:

– $hor_min = hor_min / SubWidthC$

– $hor_max = hor_max / SubWidthC$

– $ver_min = ver_min / SubHeightC$

– $ver_max = ver_max / SubHeightC$

trong đó hor_min , ver_min , hor_max và ver_max biểu diễn khoảng vector chuyển động thứ nhất được định tỷ lệ.

Phương pháp còn có thể bao gồm các bước: thu nhận các mẫu được nội suy trong ảnh tham chiếu dựa vào trường vector chuyển động được dẫn ra nhờ sử dụng nội suy song tuyến tính; và áp dụng bộ lọc thông cao cho các mẫu được nội suy. Theo đó phương pháp có thể áp dụng có lợi cho việc bù chuyển động afin nhờ sử dụng bộ lọc nội suy nâng cao (EIF). Cũng xem các phần luận nêu trên liên quan đến vấn đề này.

Khối con của khối được lập mã afin có thể có kích thước 4x4. Ngưỡng được định rõ trước có thể là 72.

Trong trường hợp rằng việc liên dự đoán afin bao gồm việc dự đoán đôi, bước thiết đặt biến $clipMVX$ có thể bao gồm việc dẫn ra biến $clipMV0$ cho danh mục 0, dẫn ra biến $clipMV1$ cho danh mục 1, và dẫn ra biến $clipMV$ là $clipMV0 \mid clipMV1$, trong đó “|” nghĩa là HOẶC.

Fig.13 là sơ đồ khối minh họa bộ giải mã 1300 D và bộ mã hóa 1300 E theo khía cạnh thứ năm và thứ sáu của sáng chế.

Bộ giải mã 1300 D theo khía cạnh thứ năm bao gồm một hoặc nhiều bộ xử lý 1301 D; và phương tiện ghi đọc được bởi máy tính không tạm thời 1302 D được ghép nối với một

hoặc nhiều bộ xử lý 1301 D và lưu trữ các lệnh cho việc thực thi bởi một hoặc nhiều bộ xử lý 1301 D, trong đó các lệnh, khi được thực thi bởi một hoặc nhiều bộ xử lý, tạo cấu hình bộ giải mã 1300 D để tiến hành phương pháp theo khía cạnh thứ nhất của sáng chế hoặc dạng thực hiện bất kỳ của nó.

Bộ mã hóa 1300 E theo khía cạnh thứ sáu bao gồm một hoặc nhiều bộ xử lý 1301 E; và phương tiện ghi đọc được bởi máy tính không tạm thời 1302 E được ghép nối với một hoặc nhiều bộ xử lý 1301 E và lưu trữ các lệnh cho việc thực thi bởi một hoặc nhiều bộ xử lý 1301 E, trong đó các lệnh, khi được thực thi bởi một hoặc nhiều bộ xử lý, tạo cấu hình bộ mã hóa 1300 E để tiến hành phương pháp theo khía cạnh thứ nhất của sáng chế hoặc dạng thực hiện bất kỳ của nó.

Fig.14 là sơ đồ khối minh họa bộ giải mã 1400 D và bộ mã hóa 1400 E theo khía cạnh thứ tám của sáng chế.

Bộ giải mã 1400 D hoặc bộ mã hóa 1400 E để lập mã chuỗi video bao gồm bộ phận xác định 1401 D/E được tạo cấu hình xác định các vectơ chuyển động điểm điều khiển (CPMV), cho khối được lập mã afin được lập mã ở chế độ afin và để xác định vùng tham chiếu trong ảnh tham chiếu tương ứng với khối con của khối được lập mã afin dựa vào các CPMV được xác định; và bộ phận dự đoán 1402 D/E được tạo cấu hình để thiết đặt biến clipMVX bằng đúng nếu kích thước của vùng tham chiếu lớn hơn ngưỡng được định rõ trước, theo cách khác thiết đặt biến clipMVX bằng sai; dẫn ra trường vectơ chuyển động dựa trên điểm ảnh cho khối được lập mã afin, trong đó, nếu biến clipMVX bằng đúng, bước dẫn ra trường vectơ chuyển động dựa trên điểm ảnh còn bao gồm việc cắt vectơ chuyển động dựa vào khoảng cắt thứ nhất, trong đó khoảng cắt thứ nhất được xác định dựa vào các CPMV được xác định và kích thước của khối được lập mã afin.

Sáng chế đưa ra các ví dụ thêm sau đây:

1. Phương pháp lập mã được thực hiện bởi thiết bị mã hóa/giải mã để lập mã dữ liệu video, trong đó phương pháp này bao gồm các bước:

tính kích thước khối con tốt nhất $M \times N$ dựa vào các thông số mô hình chuyển động afin;

khi cả độ rộng khối con tốt nhất M và độ cao khối con tốt nhất N lớn hơn hoặc bằng trị số được định rõ trước, thực hiện việc bù chuyển động khối con với khối con $M \times N$.

khi không có độ rộng khối con tốt nhất M cũng như không có độ cao khối con tốt nhất N lớn hơn hoặc bằng trị số được định rõ trước, kiểm tra các điều kiện có khả năng ứng dụng bộ lọc nội suy song tuyến tính nâng cao (EIF).

2. Phương pháp theo ví dụ 1, trong đó trị số được định rõ trước là 8.
3. Phương pháp theo ví dụ 1 hoặc 2, trong đó khi tất cả các điều kiện có khả năng ứng dụng EIF được đáp ứng, phương pháp này còn bao gồm bước:
- thực hiện việc bù chuyển động EIF.
4. Phương pháp theo ví dụ 1 hoặc 2, trong đó khi một trong số các điều kiện có khả năng ứng dụng EIF không được đáp ứng, phương pháp này còn bao gồm các bước:
- tính khoảng vectơ chuyển động thứ nhất mà đưa ra việc hoàn tất của các điều kiện mà đã không được đáp ứng; và
- cắt các vectơ chuyển động được tính theo chế độ affin sao cho các vectơ này trong khoảng vectơ chuyển động thứ nhất.
5. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1 đến 4, trong đó nếu ít nhất một trong số các điều kiện nêu trên không được đáp ứng, phương pháp này còn bao gồm bước:
- thiết đặt $M = \max(M, 8)$, $N = \max(N, 8)$ và thực hiện việc bù chuyển động khối con với khối con $M \times N$
6. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1 đến 5, trong đó các điều kiện có khả năng ứng dụng EIF bao gồm ít nhất một trong số:
- các giới hạn độ rộng dải bộ nhớ;
 - bộ đệm bên trong được giới hạn bởi các dòng R, trong đó R là trị số được định rõ trước;
 - truy cập bộ nhớ là tuần tự;
 - không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất;
 - các bất đẳng thức cho các thông số mô hình chuyển động affin.
7. Phương pháp theo ví dụ 6, trong đó các bất đẳng thức cho các thông số mô hình chuyển động affin, như sau:

$$a. \begin{cases} a \leq dHorX \leq b \\ c \leq dHorY \leq d \\ e \leq dVerX \leq f \\ g \leq dVerY \leq h \end{cases}, \text{ trong đó } a, b, c, d, e, f, g, h \text{ là các trị số được định}$$

rõ trước hoặc âm/dương vô cực.

$$b. \begin{cases} dHorX \leq a * dVerX \\ dVerY \leq b * dHorY \end{cases}, \text{ trong đó } a \text{ và } b \text{ là các trị số được định rõ}$$

trước.

8. Phương pháp lập mã được thực hiện bởi thiết bị mã hóa/giải mã để lập mã dữ liệu video, trong đó phương pháp này bao gồm các bước:

tính kích thước khối con tốt nhất $M \times N$ dựa vào các thông số mô hình chuyển động afin; nếu cả độ rộng khối con tốt nhất M và độ cao khối con tốt nhất N lớn hơn hoặc bằng 8, thực hiện việc bù chuyển động khối con với khối con $M \times N$.

theo cách khác, kiểm tra các điều kiện có khả năng ứng dụng EIF:

a. Điều kiện có khả năng ứng dụng EIF 1: Bộ đệm bên trong được giới hạn bởi các dòng R , trong đó R là trị số được định rõ trước và có thể ví dụ là 3, 4, 5 hoặc lớn hơn. Điều này nghĩa là trong suốt thời gian xử lý của một dòng (một hàng) của khối (khối con) hiện thời không nhiều hơn các dòng R từ ảnh tham chiếu có thể được sử dụng.

b. Điều kiện có khả năng ứng dụng EIF 2: Truy cập bộ nhớ sẽ là tuần tự, mà nghĩa là nếu đối với dòng thứ i của khối hiện thời dòng thứ j của ảnh tham chiếu được nạp, sau đó đối với dòng thứ $(i+1)$ của khối hiện thời chỉ các dòng $j+1, j+2, \dots$ có thể được nạp.

c. Điều kiện có khả năng ứng dụng EIF 3: Không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất.

d. ...

e. Điều kiện có khả năng ứng dụng EIF P

f. Nếu tất cả các điều kiện có khả năng ứng dụng EIF được đáp ứng, thì thực hiện việc bù chuyển động EIF. Quy trình bù chuyển động EIF bao gồm các bước:

i. Kiểm tra điều kiện có khả năng ứng dụng EIF $P + 1$: Kiểm tra các giới hạn độ rộng dải bộ nhớ.

ii. ...

iii. Kiểm tra điều kiện có khả năng ứng dụng EIF $P+K$

iv. Nếu một trong số các điều kiện có khả năng ứng dụng EIF $P+1$.. $P+K$ không được đáp ứng, thì

1. tính khoảng vector chuyển động thứ nhất mà đưa ra việc hoàn tất của các điều kiện mà đã không được đáp ứng.

2. cắt các vector chuyển động được tính theo chế độ affin để đảm bảo, rằng các vector này trong khoảng vector chuyển động thứ nhất

g. Theo cách khác thiết đặt $M' = \max(M, 8)$, $N' = \max(N, 8)$ và thực hiện việc bù chuyển động khối con với khối con $M \times N$.

9. Phương pháp lập mã được thực hiện bởi thiết bị mã hóa/giải mã để lập mã dữ liệu video, trong đó phương pháp này bao gồm các bước:

tính kích thước khối con tốt nhất $M \times N$ dựa vào các thông số mô hình chuyển động affin; nếu cả độ rộng khối con tốt nhất M và độ cao khối con tốt nhất N lớn hơn hoặc bằng 8, thực hiện việc bù chuyển động khối con với khối con $M \times N$.

theo cách khác, kiểm tra các điều kiện có khả năng ứng dụng EIF:

a. Điều kiện có khả năng ứng dụng EIF 1: Bộ đệm bên trong được giới hạn bởi các dòng R , trong đó R là trị số được định rõ trước và có thể ví dụ là 3, 4, 5 hoặc lớn hơn. Điều này nghĩa là trong suốt thời gian xử lý của một dòng (một hàng) của khối (khối con) hiện thời không nhiều hơn các dòng R từ ảnh tham chiếu có thể được sử dụng.

b. Điều kiện có khả năng ứng dụng EIF 2: Truy cập bộ nhớ sẽ là tuần tự, mà nghĩa là nếu đối với dòng thứ i của khối hiện thời dòng thứ j của ảnh tham chiếu được nạp, sau đó đối với dòng thứ $(i+1)$ của khối hiện thời chỉ các dòng $j+1, j+2, \dots$ có thể được nạp.

c. Điều kiện có khả năng ứng dụng EIF 3: Không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất.

d. ...

e. Điều kiện có khả năng ứng dụng EIF P

f. Nếu tất cả các điều kiện có khả năng ứng dụng EIF được đáp ứng, thì thực hiện việc bù chuyển động EIF. Quy trình bù chuyển động EIF bao gồm các bước:

v. Kiểm tra điều kiện có khả năng ứng dụng EIF $P + 1$: Kiểm tra các giới hạn độ rộng dải bộ nhớ.

vi. ...

vii. Kiểm tra điều kiện có khả năng ứng dụng EIF $P+K$

viii. Nếu một trong số các điều kiện có khả năng ứng dụng EIF P+1 .. P+K không được đáp ứng, thì

1. tính khoảng vector chuyển động thứ nhất mà đưa ra việc hoàn tất của các điều kiện mà đã không được đáp ứng.
2. thiết đặt khoảng vector chuyển động thứ hai bằng khoảng vector chuyển động thứ nhất

Theo cách khác,

1. tính khoảng vector chuyển động thứ ba
2. thiết đặt khoảng vector chuyển động thứ hai bằng khoảng vector chuyển động thứ ba

ix. Cắt các vector chuyển động được tính theo chế độ afin để đảm bảo, rằng các vector này trong khoảng vector chuyển động thứ hai

g. Theo cách khác thiết đặt $M' = \max(M, 8)$, $N' = \max(N, 8)$ và thực hiện việc bù chuyển động khối con với khối con $M \times N$.

10. Phương pháp theo ví dụ 9, trong đó bước kiểm tra các giới hạn độ rộng dải bộ nhớ bao gồm việc tính toán của các MV của các điểm góc của khối afin.

11. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 9 đến 10, trong đó bước kiểm tra các giới hạn độ rộng dải bộ nhớ bao gồm việc tính toán vùng của hộp giới hạn bao gồm 4 điểm mà được trở tới bởi các MV của các điểm góc của khối afin.

12. Phương pháp theo ví dụ 11, trong đó hộp giới hạn bao gồm 4 điểm mà được trở tới bởi các MV của các điểm góc của khối afin là vùng hình vuông với kích thước $(W+D1) \times (H+D2)$, trong đó $W \times H$ là kích thước của vùng hình vuông nhỏ nhất (vùng với trị số vùng nhỏ nhất) bao gồm 4 điểm mà được trở tới bởi các MV của các điểm góc của khối afin và $D1$ và $D2$ là các số nguyên không âm.

13. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 9 đến 12 trong đó bước tính khoảng vector chuyển động thứ nhất mà đưa ra việc hoàn tất của các điều kiện mà đã không được đáp ứng bao gồm việc tính MV cho điểm trung tâm của khối afin.

14. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 9 đến 13 trong đó bước tính khoảng vector chuyển động thứ nhất mà đưa ra việc hoàn tất của các điều kiện, mà đã không được đáp ứng, bao gồm việc tính việc trải vector chuyển động dựa vào kích thước khối afin.

15. Phương pháp theo ví dụ 14 trong đó việc trải vector chuyển động được tính dựa vào các số nguyên được định rõ trước cho mỗi độ rộng khối khả thi và cho mỗi độ cao khối

khả thi.

16. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 9 đến 15, trong đó khoảng vectơ chuyển động thứ ba được tính để đảm bảo rằng các MV trở tới vùng bên trong ảnh tham chiếu với lề.

17. Phương pháp theo ví dụ 16, trong đó lề tùy thuộc vào kích thước CTU lớn nhất.

18. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 8 đến 17, trong đó N bằng 3 mà nghĩa là thuật toán nêu trên sẽ không có các bước d – e.

19. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 8 đến 17, trong đó K bằng 1 mà nghĩa là thuật toán nêu trên sẽ không có các bước f.ii – f.iii.

20. Phương pháp lập mã được thực hiện bởi thiết bị mã hóa/giải mã để lập mã dữ liệu video, trong đó phương pháp này bao gồm các bước:

tính kích thước khối con tốt nhất $M \times N$ dựa vào các thông số mô hình chuyển động afin;

khi cả độ rộng khối con tốt nhất M và độ cao khối con tốt nhất N lớn hơn hoặc bằng 8, thực hiện việc bù chuyển động khối con với khối con $M \times N$;

Theo cách khác, kiểm tra các điều kiện có khả năng ứng dụng EIF

a) Điều kiện có khả năng ứng dụng EIF 1

b) Điều kiện có khả năng ứng dụng EIF 2

c) ...

d) Điều kiện có khả năng ứng dụng EIF P

e) Nếu tất cả các điều kiện có khả năng ứng dụng EIF được đáp ứng, thì thực hiện việc bù chuyển động EIF. Quy trình bù chuyển động EIF bao gồm các bước:

i. Kiểm tra điều kiện có khả năng ứng dụng EIF P+1

ii. Kiểm tra điều kiện có khả năng ứng dụng EIF P+2

iii. ...

iv. Kiểm tra điều kiện có khả năng ứng dụng EIF P+K

v. Nếu một trong số các điều kiện có khả năng ứng dụng EIF P+1 .. P+K

không được đáp ứng, thì

– tính khoảng vectơ chuyển động thứ nhất mà đưa ra việc hoàn tất của các điều kiện mà đã không được đáp ứng;

– cắt các vectơ chuyển động được tính theo chế độ afin để đảm bảo, rằng các vectơ này trong khoảng vectơ chuyển động thứ nhất;

f) Theo cách khác thiết đặt $M' = \max(M, 8)$, $N' = \max(N, 8)$ và thực hiện việc bù chuyển động khối con với khối con $M \times N$.

21. Phương pháp theo ví dụ 20, trong đó N bằng không mà nghĩa là thuật toán sẽ không có các bước a) – d).

22. Phương pháp theo ví dụ 20, trong đó K bằng không mà nghĩa là thuật toán sẽ không có các bước e).i – e).v.

23. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 20 đến 22, trong đó các điều kiện có khả năng ứng dụng EIF bao gồm ít nhất một trong số:

- các giới hạn độ rộng dải bộ nhớ;
- bộ đệm bên trong được giới hạn bởi các dòng R , trong đó R là trị số được định rõ trước;
- truy cập bộ nhớ là tuần tự;
- không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất;
- các bất đẳng thức cho các thông số mô hình chuyển động affin.

24. Phương pháp theo ví dụ 23, trong đó các bất đẳng thức cho các thông số mô hình chuyển động affin, như sau:

$$c. \begin{cases} a \leq dHorX \leq b \\ c \leq dHorY \leq d \\ e \leq dVerX \leq f \\ g \leq dVerY \leq h \end{cases}, \text{ trong đó } a, b, c, d, e, f, g, h \text{ là các trị số được định}$$

rõ trước hoặc âm/dương vô cực.

$$\begin{cases} dHorX \leq a * dVerX \\ dVerY \leq b * dHorY \end{cases}, \text{ trong đó } a \text{ và } b \text{ là các trị số được định rõ trước.}$$

25. Phương pháp lập mã được thực hiện bởi thiết bị mã hóa/giải mã để lập mã dữ liệu video, trong đó phương pháp này bao gồm các bước:

tính kích thước khối con $M \times N$ dựa vào các thông số mô hình chuyển động affin hoặc dựa vào thông tin mà từ đó các thông số mô hình chuyển động affin có thể được dẫn ra;

trong trường hợp rằng hoặc độ rộng khối con M hoặc độ cao khối con N nhỏ hơn hoặc bằng trị số được định rõ trước, thực hiện quy trình bù chuyển động bộ lọc nội suy song tuyến tính nâng cao (EIF), trong đó bước thực hiện quy trình bù chuyển động EIF bao gồm:

dẫn ra vector chuyển động của khối con tương ứng của khối hình ảnh (chẳng hạn như khối hình ảnh affin) dựa vào các thông số mô hình chuyển động affin trên cơ sở khối con $P \times Q$ (chẳng hạn như 1×1); và

thực hiện việc cắt trên vector chuyển động của khối con, sao cho vector chuyển động được cắt trong khoảng vector chuyển động (chẳng hạn như khoảng vector chuyển động thứ hai).

26. Phương pháp theo ví dụ 25, trong đó trong trường hợp rằng hoặc độ rộng khối con M hoặc độ cao khối con N nhỏ hơn hoặc bằng trị số được định rõ trước, phương pháp này còn bao gồm các bước:

xác định xem tập hợp thứ nhất của các điều kiện có khả năng ứng dụng EIF được đáp ứng hay không;

trong đó bước thực hiện quy trình bù chuyển động EIF, bao gồm:

nếu tập hợp thứ nhất của các điều kiện có khả năng ứng dụng EIF được đáp ứng, thực hiện quy trình bù chuyển động EIF.

27. Phương pháp theo ví dụ 25 hoặc 26, trong đó khoảng vector chuyển động được thiết đặt là khoảng vector chuyển động thứ nhất, hoặc khoảng vector chuyển động được thiết đặt là khoảng vector chuyển động thứ ba.

28. Phương pháp theo ví dụ 27, trong đó khoảng vector chuyển động được thiết đặt là khoảng vector chuyển động thứ ba nếu tập hợp thứ hai của các điều kiện có khả năng ứng dụng EIF được đáp ứng,

Theo cách khác, khoảng vector chuyển động được thiết đặt là khoảng vector chuyển động thứ nhất.

29. Phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên, trong đó phương pháp này còn bao gồm bước:

nếu bất kỳ một trong số tập hợp thứ nhất của các điều kiện có khả năng ứng dụng EIF không được đáp ứng, thực hiện, dựa vào các thông số mô hình chuyển động afin, việc bù chuyển động khối con trên cơ sở khối con $M' \times N'$;

trong đó $M' = \max(M, 8)$, $N' = \max(N, 8)$.

30. Phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên, trong đó phương pháp này còn bao gồm bước:

khi cả độ rộng khối con M và độ cao khối con N lớn hơn hoặc bằng trị số được định rõ trước (chẳng hạn như 8), thực hiện việc bù chuyển động khối con dựa vào các thông số mô hình chuyển động afin trên cơ sở khối con $M \times N$.

31. Phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên, trong đó bước thực hiện quy trình bù chuyển động EIF còn bao gồm:

thực hiện việc bù chuyển động của mỗi khối con dựa vào vector chuyển động được cắt.

32. Phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên, trong đó bước tính kích thước khối con $M \times N$, bao gồm:

dẫn ra kích thước khối con $M \times N$ dựa vào các độ chênh lệch vector chuyển động của một hoặc nhiều điểm điều khiển afin của khối hình ảnh afin và độ rộng và độ cao của khối hình ảnh afin; hoặc

thu nhận kích thước khối con $M \times N$ nhờ sử dụng bảng tra cứu theo một hoặc nhiều trong số:

các độ chênh lệch vector chuyển động của một hoặc nhiều điểm điều khiển afin của khối hình ảnh afin,

kích thước khối của khối hình ảnh afin (chẳng hạn như, độ rộng và/hoặc độ cao của khối hình ảnh afin), và

độ chính xác vector chuyển động được định rõ trước.

33. Phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên, trong đó tập hợp thứ nhất của các điều kiện có khả năng ứng dụng EIF bao gồm ít nhất một trong số:

- các giới hạn độ rộng dải bộ nhớ;
- bộ đệm bên trong được giới hạn bởi các dòng R, trong đó R là trị số được định rõ trước;
- truy cập bộ nhớ là tuần tự;
- không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất;
- các bất đẳng thức cho các thông số mô hình chuyển động afin.

34. Phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên, trong đó kích thước khối con $P \times Q$ bằng 1×1 .

35. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 26 đến 34, trong đó tập hợp thứ nhất của các điều kiện có khả năng ứng dụng EIF bao gồm:

- bộ đệm bên trong được giới hạn bởi các dòng R, trong đó R là trị số được định rõ trước;
- truy cập bộ nhớ là tuần tự;
- không nhiều hơn một dòng bổ sung có thể được nạp cho tất cả các dòng của khối hiện thời ngoại trừ dòng thứ nhất.

36. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 28 đến 35, trong đó tập hợp

thứ hai của các điều kiện có khả năng ứng dụng EIF bao gồm các giới hạn độ rộng dải bộ nhớ.

37. Phương pháp theo ví dụ 36, trong đó bước kiểm tra các giới hạn độ rộng dải bộ nhớ bao gồm việc tính các MV của các điểm góc (ví dụ, các điểm điều khiển) của khối afin.

38. Phương pháp theo ví dụ bất kỳ trong số ví dụ 37, trong đó bước kiểm tra các giới hạn độ rộng dải bộ nhớ còn bao gồm việc tính vùng của hộp giới hạn bao gồm 4 điểm mà được trở tới bởi các MV của các điểm góc của khối afin.

39. Phương pháp theo ví dụ 38, trong đó hộp giới hạn bao gồm 4 điểm mà được trở tới bởi các MV của các điểm góc của khối afin là vùng hình vuông với kích thước $(W+D1) \times (H+D2)$, trong đó $W \times H$ là kích thước của vùng hình vuông nhỏ nhất (chẳng hạn như, vùng với vùng trị số nhỏ nhất) bao gồm 4 điểm mà được trở tới bởi các MV của các điểm góc của khối afin và $D1$ và $D2$ là các số nguyên không âm.

40. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 38 đến 39, trong đó bước kiểm tra các giới hạn độ rộng dải bộ nhớ còn bao gồm:

- ngưỡng tính toán $T1$ dựa vào kích thước khối afin $W \times H$ và ngưỡng được định rõ trước T ,
- so sánh trị số của vùng được tính của hộp giới hạn với ngưỡng $T1$.

41. Phương pháp theo ví dụ 40, trong đó ngưỡng tính toán $T1$ dựa vào kích thước khối afin $W \times H$ và ngưỡng được định rõ trước T được thực hiện là $T1 = W \times H \times T$.

42. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 40 đến 41, trong đó các giới hạn độ rộng dải bộ nhớ được coi là được đáp ứng nếu trị số của vùng được tính của hộp giới hạn nhỏ hơn hoặc bằng ngưỡng $T1$.

43. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 27 đến 42 trong đó việc tính khoảng vector chuyển động thứ nhất bao gồm việc tính MV cho điểm trung tâm của khối afin.

44. Phương pháp theo ví dụ 43, trong đó việc tính khoảng vector chuyển động thứ nhất còn bao gồm việc tính việc trải vector chuyển động dựa vào kích thước khối afin (ví dụ, kích thước khối afin $W \times H$).

45. Phương pháp theo ví dụ 44, trong đó việc trải vector chuyển động được tính dựa vào các số nguyên được định rõ trước cho mỗi độ rộng khối khả thi và cho mỗi độ cao khối khả thi.

46. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 27 đến 45, trong đó khoảng

vectơ chuyển động thứ ba được tính để đảm bảo rằng các MV trở tới vùng bên trong ảnh tham chiếu với lề, trong đó lề là vùng được mở rộng xung quanh ảnh tham chiếu.

47. Phương pháp theo ví dụ 46, trong đó lề tùy thuộc vào kích thước CTU lớn nhất.

48. Phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên, trong đó bước dẫn ra vectơ chuyển động của khối con tương ứng của khối hình ảnh tùy thuộc vào mặt phẳng màu (chẳng hạn như luma hoặc chroma) được xử lý bởi EIF.

49. Phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên, trong đó bước xác định xem tập hợp thứ nhất của các điều kiện có khả năng ứng dụng EIF được đáp ứng hay không bao gồm:

xác định xem phần hoặc toàn bộ của tập hợp thứ nhất của các điều kiện có khả năng ứng dụng EIF được đáp ứng cho các thông số chuyển động afin được thu nhận từ các vectơ chuyển động điểm điều khiển được kết hợp với danh mục ảnh tham chiếu thứ nhất (chẳng hạn như list0) hay không, và/hoặc

xác định xem phần hoặc toàn bộ của tập hợp thứ nhất của các điều kiện có khả năng ứng dụng EIF được đáp ứng cho các thông số chuyển động afin được thu nhận từ các vectơ chuyển động điểm điều khiển được kết hợp với danh mục ảnh tham chiếu thứ hai (chẳng hạn như list1) hay không.

50. Phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên, trong đó bước xác định xem tập hợp thứ nhất của các điều kiện có khả năng ứng dụng EIF được đáp ứng hay không bao gồm:

xét đến tập hợp thứ nhất của các điều kiện có khả năng ứng dụng EIF khi được đáp ứng nếu phần hoặc toàn bộ của tập hợp thứ nhất của các điều kiện có khả năng ứng dụng EIF được đáp ứng cho các thông số chuyển động afin được thu nhận từ các vectơ chuyển động điểm điều khiển được kết hợp với danh mục ảnh tham chiếu thứ nhất (chẳng hạn như list0) và nếu phần hoặc toàn bộ của tập hợp thứ nhất của các điều kiện có khả năng ứng dụng EIF được đáp ứng cho các thông số chuyển động afin được thu nhận từ các vectơ chuyển động điểm điều khiển được kết hợp với danh mục ảnh tham chiếu thứ hai (chẳng hạn như list1).

51. Phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên, trong đó bước xác định xem tập hợp thứ hai của các điều kiện có khả năng ứng dụng EIF được đáp ứng hay không bao gồm:

xác định xem phần hoặc toàn bộ của tập hợp thứ hai của các điều kiện có khả năng

ứng dụng EIF được đáp ứng cho các thông số chuyển động afin được thu nhận từ các vector chuyển động điểm điều khiển được kết hợp với danh mục ảnh tham chiếu thứ nhất (chẳng hạn như list0) hay không, và/hoặc

xác định xem phần hoặc toàn bộ của tập hợp thứ hai của các điều kiện có khả năng ứng dụng EIF được đáp ứng cho các thông số chuyển động afin được thu nhận từ các vector chuyển động điểm điều khiển được kết hợp với danh mục ảnh tham chiếu thứ hai (chẳng hạn như list1) hay không.

52. Phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên, trong đó bước xác định xem tập hợp thứ hai của các điều kiện có khả năng ứng dụng EIF được đáp ứng hay không bao gồm:

xét đến tập hợp thứ hai của các điều kiện có khả năng ứng dụng EIF khi được đáp ứng nếu phần hoặc toàn bộ của tập hợp thứ hai của các điều kiện có khả năng ứng dụng EIF được đáp ứng cho các thông số chuyển động afin được thu nhận từ các vector chuyển động điểm điều khiển được kết hợp với danh mục ảnh tham chiếu thứ nhất (chẳng hạn như list0) và nếu phần hoặc toàn bộ của tập hợp thứ hai của các điều kiện có khả năng ứng dụng EIF được đáp ứng cho các thông số chuyển động afin được thu nhận từ các vector chuyển động điểm điều khiển được kết hợp với danh mục ảnh tham chiếu thứ hai (chẳng hạn như list1).

53. Bộ mã hóa (20) bao gồm hệ mạch xử lý để tiến hành phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1 đến 52.

54. Bộ giải mã (30) bao gồm hệ mạch xử lý để tiến hành phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1 đến 52

55. Sản phẩm chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên khi được thực thi trên máy tính hoặc bộ xử lý.

56. Bộ giải mã, bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện ghi đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ việc lập trình cho việc thực thi bởi các bộ xử lý, trong đó bước lập trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ giải mã để tiến hành phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên.

57. Bộ mã hóa, bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện ghi đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ việc lập trình cho việc thực thi bởi các bộ xử lý, trong đó bước lập trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ mã hóa để tiến hành phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên.

58. Phương tiện đọc được bởi máy tính không tạm thời mang mã chương trình mà, khi được thực thi bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên.

Các toán tử toán học

Các toán tử toán học được sử dụng theo sáng chế tương tự các toán tử toán học được sử dụng trong ngôn ngữ lập trình C. Tuy nhiên, các kết quả của phép chia số nguyên và các phép toán dịch chuyển số học được định rõ một cách chính xác hơn, và các phép toán bổ sung được định rõ, chẳng hạn như lũy thừa và phép chia trị số thực. Các quy ước đếm và đánh số nói chung bắt đầu từ 0, ví dụ, "thứ nhất" tương đương với thứ 0, "thứ hai" tương đương với thứ 1 chẳng hạn.

Các toán tử số học

Các toán tử số học sau đây được định rõ như sau:

+ Phép cộng

− Phép trừ (là toán tử hai đối số) hoặc phủ định (là toán tử tiền tố một ngôi)

* Phép nhân, bao gồm phép nhân ma trận

x^y Lũy thừa. Xác định x là lũy thừa của y . Trong các ngữ cảnh khác, ký hiệu như vậy được sử dụng để viết trên không nhằm diễn dịch là lũy thừa.

/ Phép chia số nguyên với việc rút ngắn kết quả về phía không. Ví dụ, $7 / 4$ và $-7 / -4$ được rút ngắn tới 1 và $-7 / 4$ và $7 / -4$ được rút ngắn tới -1 .

$\div$ Được sử dụng để biểu thị phép chia trong các phương trình toán học trong đó không việc rút ngắn hay làm tròn nào được dự định.

$\frac{x}{y}$ Được sử dụng để biểu thị phép chia trong các phương trình toán học trong đó không việc rút ngắn hay làm tròn nào được dự định.

$\sum_{i=x}^y f(i)$ Phép tổng của $f(i)$ với i lấy tất cả các trị số nguyên từ x lên đến và bao gồm y .

$x \% y$ Môđun. Số dư của x được chia cho y , được định rõ chỉ cho các số nguyên x và y với $x \geq 0$ và $y > 0$.

Các toán tử logic

Các toán tử logic sau đây được định rõ như sau:

$x \&\& y$ Logic Boolean "và (and)" của x và y

$x || y$ Logic Boolean "hoặc (or)" của x và y

$!$ Logic Boolean "không (not)"

$x ? y : z$ If x là đúng hoặc không bằng 0, đánh giá tới trị số của y ; theo cách khác, đánh giá tới trị số của z .

Các toán tử quan hệ

Các toán tử quan hệ sau đây được định rõ như sau:

$>$ Lớn hơn so với

$>=$ Lớn hơn hoặc bằng

$<$ Nhỏ hơn so với

$<=$ Nhỏ hơn hoặc bằng

$==$ Bằng

$!=$ Không bằng

Khi toán tử quan hệ được áp dụng cho thành phần cú pháp hoặc biến mà đã được gán trị số "na" (not applicable - không thể áp dụng được), trị số "na" được coi là trị số phân biệt cho thành phần cú pháp hoặc biến. Trị số "na" được coi là không bằng trị số khác bất kỳ.

Các toán tử đảo bit

Các toán tử đảo bit sau đây được định rõ như sau:

$\&$ Đảo bit "and". Khi thực hiện phép toán trên các đối số nguyên, thực hiện phép toán trên sự biểu diễn bù hai của trị số nguyên. Khi thực hiện phép toán trên đối số nhị phân mà chứa ít bit hơn so với đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung nhiều bit quan trọng hơn bằng 0.

$|$ Đảo bit "hoặc (or)". Khi thực hiện phép toán trên các đối số nguyên, thực hiện phép toán trên sự biểu diễn bù hai của trị số nguyên. Khi thực hiện phép toán trên đối số nhị phân mà chứa ít bit hơn so với đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung nhiều bit quan trọng hơn bằng 0.

$\wedge$ Đảo bit "loại trừ hoặc". Khi thực hiện phép toán trên các đối số nguyên, thực hiện phép toán trên sự biểu diễn bù hai của trị số nguyên. Khi thực hiện phép

toán trên đối số nhị phân mà chứa ít bit hơn so với 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 chuyển bên phải số học của sự biểu diễn số nguyên bù hai của các chữ số nhị phân x nhân y . Hàm này được định rõ chỉ cho các trị số số nguyên không âm của y . Các bit được dịch chuyển thành các bit quan trọng nhất (most significant bit, viết tắt là MSB) là kết quả của việc dịch chuyển bên phải có trị số bằng MSB của x trước phép toán dịch chuyển.

$x \ll y$ Việc dịch chuyển bên trái số học của sự biểu diễn số nguyên bù hai của các chữ số nhị phân x nhân y . Hàm này được định rõ chỉ cho các trị số số nguyên không âm của y . Các bit được dịch chuyển thành các bit ít quan trọng nhất (least significant bit, viết tắt là LSB) là kết quả của việc dịch chuyển bên trái có trị số bằng 0.

Các toán tử gán

Các toán tử số học sau đây được định rõ như sau:

=Toán tử gán

$++$ Lượng tăng, nghĩa là, $x++$ tương đương với $x = x + 1$; khi được sử dụng trong chỉ số mảng, đánh giá tới trị số của biến trước phép toán tăng.

$--$ Lượng giảm, nghĩa là, $x--$ tương đương với $x = x - 1$; khi được sử dụng trong chỉ số mảng, đánh giá tới trị số của biến trước phép toán giảm.

$+=$ Lượng tăng bởi lượng được quy định, nghĩa là, $x += 3$ tương đương với $x = x + 3$, và $x += (-3)$ tương đương với $x = x + (-3)$.

$-=$ Lượng giảm bởi lượng được quy định, nghĩa là, $x -= 3$ tương đương với $x = x - 3$, và $x -= (-3)$ tương đương với $x = x - (-3)$.

Ký hiệu khoảng

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

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

Các hàm toán học

Các hàm toán học sau đây được định rõ:

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

$\text{Asin}(x)$ hàm sin ngược lượng giác, thực hiện phép toán trên đối số x mà trong khoảng từ $-1,0$ đến $1,0$, với trị số đầu ra trong khoảng từ $-\pi/2$ đến $\pi/2$, theo đơn vị radian

$\text{Atan}(x)$ hàm tiếp tuyến ngược lượng giác, thực hiện phép toán trên đối số x , với trị số đầu ra trong khoảng từ $-\pi/2$ đến $\pi/2$, theo đơn vị radian

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

$\text{Ceil}(x)$ số nguyên nhỏ nhất lớn hơn hoặc bằng x .

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

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

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

$\text{Cos}(x)$ hàm cosin lượng giác thực hiện phép toán trên đối số x theo đơn vị radian.

$\text{Floor}(x)$ số nguyên lớn nhất nhỏ hơn hoặc bằng x .

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

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

$\text{Log2}(x)$ lôgarit cơ số 2 của x .

$\text{Log10}(x)$ lôgarit cơ số 10 của x .

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

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

$$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0.5)$$

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

$\text{Sin}(x)$ hàm sin lượng giác thực hiện phép toán trên đối số x theo đơn vị radian

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

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

$\text{Tan}(x)$ hàm tiếp tuyến lượng giác thực hiện phép toán trên đối số x theo đơn vị radian

Thứ tự ưu tiên phép toán

Khi thứ tự ưu tiên trong biểu thức không được chỉ báo một cách rõ ràng nhờ sử dụng các dấu ngoặc đơn, các quy tắc sau đây áp dụng:

- Các phép toán ưu tiên cao hơn được đánh giá trước phép toán ưu tiên thấp hơn bất kỳ.
- Các phép toán ưu tiên giống nhau được đánh giá tuần tự từ trái sang phải.

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

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

Bảng: Mức ưu tiên phép toán từ cao nhất (ở trên cùng của bảng) đến thấp nhất (ở dưới cùng của bảng)

Phép toán (với toán hạng x, y, và z)
"x++", "x--"
"!x", "-x" (là toán tử tiền tố một ngôi)
x^y
"x * y", "x / y", "x ÷ y", " $\frac{x}{y}$ ", "x % y"
"x + y", "x - y" (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"
"x y"
"x ? y : z"
"x..y"
"x = y", "x += y", "x -= y"

Phần mô tả văn bản của các phép toán logic

Trong văn bản, câu lệnh của các phép toán logic như sẽ được mô tả về mặt toán học ở dạng sau đây:

Nếu (điều kiện 0)

câu lệnh 0

theo cách khác (điều kiện 1)

câu lệnh 1

...

theo cách khác /* nhận xét thông tin đối với các điều kiện còn lại */

câu lệnh n

có thể được mô tả theo cách thức sau đây:

... Như sau / ... Phần sau đây áp dụng:

- Nếu điều kiện 0, câu lệnh 0
- Theo cách khác, nếu điều kiện 1, câu lệnh 1
- ...
- Theo cách khác (nhận xét thông tin đối với điều kiện còn lại), câu lệnh

n

Mỗi "Nếu ... Theo cách khác, nếu ... Theo cách khác, ..." câu lệnh trong văn bản được giới thiệu với "... Như sau" hoặc "... Phần sau đây áp dụng" được theo ngay sau bởi "Nếu ...". Điều kiện cuối cùng của "Nếu ... Theo cách khác, nếu ... Theo cách khác, ..." luôn "Theo

cách khác, ...". Được xen kẽ "Nếu ... Theo cách khác, nếu ... Theo cách khác, ..." các câu lệnh có thể được nhận dạng bằng cách so khớp "... Như sau" hoặc "... Phần sau đây áp dụng" với kết thúc "Theo cách khác, ...".

Trong văn bản, câu lệnh của các phép toán logic như sẽ được mô tả về mặt toán học ở dạng sau đây:

Nếu (điều kiện 0a && điều kiện 0b)

câu lệnh 0

theo cách khác nếu (điều kiện 1a || điều kiện 1b)

câu lệnh 1

...

theo cách khác

câu lệnh n

có thể được mô tả theo cách thức sau đây:

... Như sau / ... Phần sau đây áp dụng:

- Nếu tất cả trong số các điều kiện sau đây là đúng, câu lệnh 0:
 - điều kiện 0a
 - điều kiện 0b
- Theo cách khác, nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, câu lệnh 1:
 - điều kiện 1a
 - điều kiện 1b
 - ...
- Theo cách khác, câu lệnh n

Trong văn bản, câu lệnh của các phép toán logic như sẽ được mô tả về mặt toán học ở dạng sau đây:

nếu (điều kiện 0)

câu lệnh 0

nếu (điều kiện 1)

câu lệnh 1

có thể được mô tả theo cách thức sau đây:

Khi điều kiện 0, câu lệnh 0

Khi điều kiện 1, câu lệnh 1.

Mặc dù các phương án của sáng chế đã được mô tả chủ yếu dựa vào việc lập mã video, nhưng cần lưu ý rằng các phương án của hệ thống lập mã 10, bộ mã hóa 20 và bộ giải mã 30 (và tương ứng là hệ thống 10) và các phương án khác được mô tả ở đây cũng có thể được tạo cấu hình cho việc lập mã hoặc xử lý ảnh tĩnh, nghĩa là, xử lý hoặc lập mã của ảnh riêng độc lập với ảnh trước đó hoặc nối tiếp bất kỳ như trong việc lập mã video. Nói chung chỉ các bộ phận liên dự đoán 244 (bộ mã hóa) và 344 (bộ giải mã) có thể không khả dụng trong trường hợp việc lập mã xử lý ảnh giới hạn ở ảnh đơn 17. Tất cả các tính năng khác (cũng được đề cập đến là các công cụ hoặc các công nghệ) của bộ mã hóa video 20 và bộ giải mã video 30 có thể được sử dụng như nhau cho việc xử lý ảnh tĩnh, ví dụ, tính toán phần dư 204/304, biến đổi 206, lượng tử hóa 208, lượng tử hóa ngược 210/310, biến đổi (ngược) 212/312, phân chia 262/362, nội dự đoán 254/354, và/hoặc lọc vòng lặp 220, 320, và lập mã entropi 270 và giải mã entropi 304.

Các phương án, ví dụ về bộ mã hóa 20 và bộ giải mã 30, và các chức năng được mô tả ở đây, ví dụ, dựa vào bộ mã hóa 20 và bộ giải mã 30, có thể được thực hiện ở phần cứng, phần mềm, phần sụn, hoặc sự kết hợp bất kỳ của chúng. Nếu được thực hiện ở phần mềm, các chức năng có thể được lưu trữ trên phương tiện đọc được bởi máy tính hoặc được truyền qua các phương tiện truyền thông như một hoặc nhiều lệnh hoặc mã và được thực thi bởi bộ phận xử lý dựa trên phần cứng. Các phương tiện đọc được bởi máy tính có thể bao gồm các phương tiện lưu trữ đọc được bởi máy tính, mà tương ứng với phương tiện hữu hình chẳng hạn như các phương tiện lưu trữ dữ liệu, hoặc phương tiện truyền thông bao gồm phương tiện bất kỳ mà tạo điều kiện thuận lợi cho việc chuyển đổi của chương trình máy tính từ một địa điểm tới địa điểm khác, ví dụ, theo giao thức truyền thông. Bằng cách thức này, các phương tiện đọc được bởi máy tính nói chung có thể tương ứng với (1) các phương tiện lưu trữ đọc được bởi máy tính hữu hình mà không tạm thời hoặc (2) phương tiện truyền thông chẳng hạn như tín hiệu hoặc sóng mang. Các phương tiện lưu trữ dữ liệu có thể là các phương tiện khả dụng bất kỳ mà có thể được truy cập bởi một hoặc nhiều máy tính hoặc một hoặc nhiều bộ xử lý để truy xuất các lệnh, mã và/hoặc các cấu trúc dữ liệu cho việc thực hiện của các kỹ thuật được mô tả theo sáng chế. Sản phẩm chương trình máy tính có thể bao gồm phương tiện đọc được bởi máy tính.

Bằng ví dụ, và không giới hạn, các phương tiện lưu trữ đọc được bởi máy tính như vậy có thể bao gồm RAM, ROM, EEPROM, CD-ROM hoặc phương tiện lưu trữ đĩa quang khác, phương tiện lưu trữ đĩa từ, hoặc các thiết bị lưu trữ từ khác, bộ nhớ tia chớp, hoặc

phương tiện khác bất kỳ mà có thể được sử dụng để lưu trữ mã chương trình mong muốn ở dạng của các lệnh hoặc các cấu trúc dữ liệu và có thể được truy cập bởi máy tính. Ngoài ra, sự kết nối bất kỳ được gọi đúng là phương tiện đọc được bởi máy tính. Ví dụ, nếu các lệnh được truyền từ trang mạng, máy chủ, hoặc nguồn từ xa khác nhờ sử dụng cáp đồng trục, cáp sợi quang, xoắn đôi, đường dây thuê bao kỹ thuật số (digital subscriber line, viết tắt là DSL), hoặc các công nghệ không dây chẳng hạn như hồng ngoại, radio, và vi sóng, sau đó cáp đồng trục, cáp sợi quang, xoắn đôi, DSL, hoặc các công nghệ không dây chẳng hạn như hồng ngoại, radio, và vi sóng được bao gồm trong định nghĩa về phương tiện. Cần hiểu, tuy nhiên, rằng các phương tiện lưu trữ đọc được bởi máy tính và các phương tiện lưu trữ dữ liệu không bao gồm các sự kết nối, các sóng mang, các tín hiệu, hoặc các phương tiện tạm thời khác, nhưng thay vì vậy được hướng tới phương tiện lưu trữ hữu hình, không tạm thời. Đĩa (disk) và đĩa (disc), như được sử dụng ở đây, bao gồm đĩa compac (compact disc, viết tắt là CD), đĩa laze, đĩa quang, đĩa đa năng kỹ thuật số (digital versatile disc, viết tắt là DVD), đĩa mềm và đĩa Blu-ray, trong đó các đĩa (disk) thường tái tạo dữ liệu từ, trong khi các đĩa (disc) tái tạo dữ liệu quang với các laze. Các sự kết hợp của các thiết bị nêu trên cũng sẽ được bao gồm nằm trong phạm vi của các phương tiện đọc được bởi máy tính.

Các lệnh có thể được thực thi bởi một hoặc nhiều bộ xử lý, chẳng hạn như một hoặc nhiều bộ xử lý tín hiệu kỹ thuật số (digital signal processor, viết tắt là DSP), bộ vi xử lý đa năng, các mạch tích hợp chuyên dụng (application specific integrated circuit, viết tắt là ASIC), các mảng logic lập trình được dạng trường (field programmable logic array, viết tắt là FPGA), hoặc hệ mạch logic rời rạc hoặc được tích hợp tương đương khác. Theo đó, thuật ngữ “bộ xử lý,” như được sử dụng ở đây có thể đề cập đến bất kỳ trong số cấu trúc nêu trên hoặc cấu trúc khác bất kỳ thích hợp cho việc thực hiện của các kỹ thuật được mô tả ở đây. Ngoài ra, theo một số khía cạnh, tính năng được mô tả ở đây có thể được cung cấp nằm trong các môđun phần cứng và/hoặc phần mềm dành riêng được tạo cấu hình để mã hóa và giải mã, hoặc được kết hợp vào bộ mã hóa/giải mã được kết hợp. Ngoài ra, các kỹ thuật có thể được thực hiện đầy đủ trong một hoặc nhiều mạch hoặc các thành phần logic.

Các kỹ thuật của sáng chế có thể được thực hiện trong nhiều loại của các thiết bị hoặc các máy, bao gồm thiết bị cầm tay không dây, mạch tích hợp (integrated circuit, viết tắt là IC) hoặc tập hợp của các IC (ví dụ, bộ chip). Các thành phần, các môđun, hoặc các bộ phận khác nhau được mô tả theo sáng chế để nhấn mạnh các khía cạnh chức năng của các thiết

bị được tạo cấu hình để thực hiện các kỹ thuật được bộc lộ, nhưng không cần yêu cầu việc thực hiện bởi các bộ phận phần cứng khác nhau. Hơn nữa, như được nêu trên, các bộ phận khác nhau có thể được kết hợp trong bộ phận phần cứng bộ mã hóa/giải mã hoặc được cung cấp bởi tập hợp của các bộ phận phần cứng tương tác, bao gồm một hoặc nhiều bộ xử lý như được nêu trên, cùng với phần mềm và/hoặc phần sụn thích hợp.

YÊU CẦU BẢO HỘ

1. Phương pháp lập mã được thực hiện bởi thiết bị mã hóa/giải mã để lập mã dữ liệu video, trong đó phương pháp này bao gồm các bước:

đôi với khối được lập mã ở chế độ afin, xác định các vector chuyển động điểm điều khiển (control point motion vector, viết tắt là CPMV);

xác định vùng tham chiếu trong ảnh tham chiếu tương ứng với khối con của khối được lập mã afin dựa vào các CPMV được xác định;

thiết đặt biến clipMVX bằng đúng (TRUE) nếu kích thước của vùng tham chiếu lớn hơn ngưỡng được định rõ trước, theo cách khác thiết đặt biến clipMVX bằng sai (FALSE);

dẫn ra trường vector chuyển động dựa trên điểm ảnh cho khối được lập mã afin,

trong đó, nếu biến clipMVX bằng đúng, bước dẫn ra trường vector chuyển động dựa trên điểm ảnh còn bao gồm việc cắt vector chuyển động với khoảng vector chuyển động thứ nhất được định tỷ lệ, trong đó khoảng vector chuyển động thứ nhất được xác định dựa vào các CPMV được xác định và kích thước của khối được lập mã afin, trong đó khoảng vector chuyển động thứ nhất được định tỷ lệ được dẫn ra bằng cách định tỷ lệ khoảng vector chuyển động thứ nhất theo các thông số lấy mẫu phụ SubWidthC và SubHeightC.

2. Phương pháp theo điểm 1, trong đó bước xác định khoảng vector chuyển động thứ nhất bao gồm:

xác định vector chuyển động cho điểm trung tâm của khối được lập mã afin;

xác định khoảng vector chuyển động thứ hai dựa vào vector chuyển động cho điểm trung tâm của khối được lập mã afin và các dịch vị được định rõ trước cho kích thước khối tương ứng, kích thước khối bao gồm kích thước của khối được lập mã afin.

3. Phương pháp theo điểm 2, trong đó bước xác định vector chuyển động cho điểm trung tâm của khối được lập mã afin được thực hiện bởi các phương trình sau đây:

$$mv_center[0] = (mvBaseScaled[0] + dX[0] * (cbWidth >> 1) + dY[0] * (cbHeight >> 1))$$

$$mv_center[1] = (mvBaseScaled[1] + dX[1] * (cbWidth >> 1) + dY[1] * (cbHeight >> 1)),$$

trong đó mvBaseScaled biểu diễn vector chuyển động của góc trên cùng bên trái của khối, cbWidth và cbHeight lần lượt biểu diễn độ rộng và độ cao của khối, dX[0], dX[1] biểu diễn các độ chênh lệch tương ứng của các phần theo chiều ngang và theo chiều dọc của vector chuyển động trên một mẫu theo hướng chiều ngang, và dY[0], dY[1] biểu diễn các độ chênh lệch tương ứng của các phần theo chiều ngang và theo chiều dọc của vector chuyển động trên một mẫu theo hướng chiều dọc; và mv_center

[0] và mv_center [1] biểu diễn các phần theo chiều ngang và theo chiều dọc của vector chuyển động cho điểm trung tâm một cách tương ứng.

4. Phương pháp theo điểm bất kỳ trong số các điểm 2 và 3, trong đó bước xác định khoảng vector chuyển động thứ hai dựa vào vector chuyển động cho điểm trung tâm của khối được lập mã afin và các dịch vị được định rõ trước cho kích thước khối tương ứng được thực hiện bởi các phương trình sau đây:

$$mv_hor_min = mv_center [0] - deviationMV [\log2CbWidth - 3]$$

$$mv_ver_min = mv_center [1] - deviationMV [\log2CbHeight - 3]$$

$$mv_hor_max = mv_center [0] + deviationMV [\log2CbWidth - 3]$$

$$mv_ver_max = mv_center [1] + deviationMV [\log2CbHeight - 3]$$

trong đó deviationMV[] biểu diễn bảng của các dịch vị được định rõ trước cho kích thước khối tương ứng, và mv_hor_min, mv_ver_min, mv_hor_max và mv_ver_max biểu diễn khoảng vector chuyển động thứ hai.

5. Phương pháp theo điểm bất kỳ trong số các điểm 2 và 3, trong đó các dịch vị được định rõ trước cho các kích thước khối tương ứng tỷ lệ với { 64, 128, 272, 560, 1136 }, trong đó

64 tương ứng với kích thước khối có kích thước tương ứng bằng 8,

128 tương ứng với kích thước khối có kích thước tương ứng bằng 16,

272 tương ứng với kích thước khối có kích thước tương ứng bằng 32,

560 tương ứng với kích thước khối có kích thước tương ứng bằng 64, hoặc

1136 tương ứng với kích thước khối có kích thước tương ứng bằng 128.

6. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó sau khi xác định khoảng vector chuyển động thứ nhất, khoảng vector chuyển động thứ nhất được định tỷ lệ theo các thông số lấy mẫu phụ SubWidthC và SubHeightC:

$$hor_min = hor_min / SubWidthC$$

$$hor_max = hor_max / SubWidthC$$

$$ver_min = ver_min / SubHeightC$$

$$ver_max = ver_max / SubHeightC$$

trong đó hor_min, ver_min, hor_max và ver_max biểu diễn khoảng vector chuyển động thứ nhất được định tỷ lệ.

7. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó phương pháp này còn bao gồm các bước:

thu nhận các mẫu được nội suy trong ảnh tham chiếu dựa vào trường vector chuyển động được dẫn ra nhờ sử dụng nội suy song tuyến tính; và

áp dụng bộ lọc thông cao cho các mẫu được nội suy.

8. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó khối con của khối được lập mã afin có kích thước 4×4 .

9. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó ngưỡng được định rõ trước là 72.

10. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó trong trường hợp mà việc liên dự đoán afin bao gồm việc dự đoán đôi, thiết đặt biến clipMVX bao gồm các bước:

dẫn ra biến clipMV0 cho danh mục 0

dẫn ra biến clipMV1 cho danh mục 1

dẫn ra biến clipMV là clipMV0 | clipMV1, trong đó “|” nghĩa là HOẶC (OR).

11. Bộ mã hóa (20) bao gồm hệ mạch xử lý để tiến hành phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 10.

12. Bộ giải mã (30) bao gồm hệ mạch xử lý để tiến hành phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 10.

13. Bộ giải mã (30), bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện ghi đọc được bởi máy tính không tạm thời được ghép nối với một hoặc nhiều bộ xử lý và lưu trữ các lệnh cho việc thực thi bởi một hoặc nhiều bộ xử lý, trong đó bước lập trình, khi được thực thi bởi một hoặc nhiều bộ xử lý, tạo cấu hình bộ giải mã để tiến hành phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 10.

14. Bộ mã hóa (20), bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện ghi đọc được bởi máy tính không tạm thời được ghép nối với một hoặc nhiều bộ xử lý và lưu trữ các lệnh cho việc thực thi bởi một hoặc nhiều bộ xử lý, trong đó bước lập trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ mã hóa để tiến hành phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 10.

15. Phương tiện đọc được bởi máy tính không tạm thời mang mã chương trình mà, khi được thực thi bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 10.

16. Bộ giải mã (30) để lập mã chuỗi video, bao gồm:

bộ phận xác định được tạo cấu hình xác định các vector chuyển động điểm điều khiển

(CPMV), cho khối được lập mã affin được lập mã ở chế độ affin và để xác định vùng tham chiếu trong ảnh tham chiếu tương ứng với khối con của khối được lập mã affin dựa vào các CPMV được xác định; và

bộ phận dự đoán được tạo cấu hình để

thiết đặt biến clipMVX bằng đúng nếu kích thước của vùng tham chiếu lớn hơn ngưỡng được định rõ trước, theo cách khác thiết đặt biến clipMVX bằng sai;

dẫn ra trường vector chuyển động dựa trên điểm ảnh cho khối được lập mã affin,

trong đó, nếu biến clipMVX bằng đúng, bước dẫn ra trường vector chuyển động dựa trên điểm ảnh còn bao gồm việc cắt vector chuyển động với khoảng vector chuyển động thứ nhất được định tỷ lệ, trong đó khoảng vector chuyển động thứ nhất được xác định dựa vào các CPMV được xác định và kích thước của khối được lập mã affin, trong đó khoảng vector chuyển động thứ nhất được định tỷ lệ được dẫn ra bằng cách định tỷ lệ khoảng vector chuyển động thứ nhất theo các thông số lấy mẫu phụ SubWidthC và SubHeightC.

17. Bộ giải mã theo điểm 16, trong đó bước xác định khoảng vector chuyển động thứ nhất bao gồm:

xác định vector chuyển động cho điểm trung tâm của khối được lập mã affin;

xác định khoảng vector chuyển động thứ hai dựa vào vector chuyển động cho điểm trung tâm của khối được lập mã affin và các dịch vị được định rõ trước cho kích thước khối tương ứng, kích thước khối bao gồm kích thước của khối được lập mã affin.

18. Bộ giải mã theo điểm 17, trong đó bước xác định vector chuyển động cho điểm trung tâm của khối được lập mã affin được thực hiện bởi các phương trình sau đây:

$$mv_center[0] = (mvBaseScaled[0] + dX[0] * (cbWidth >> 1) + dY[0] * (cbHeight >> 1))$$

$$mv_center[1] = (mvBaseScaled[1] + dX[1] * (cbWidth >> 1) + dY[1] * (cbHeight >> 1)),$$

trong đó mvBaseScaled biểu diễn vector chuyển động của góc trên cùng bên trái của khối, cbWidth và cbHeight lần lượt biểu diễn độ rộng và độ cao của khối, dX[0], dX[1] biểu diễn các độ chênh lệch tương ứng của các phần theo chiều ngang và theo chiều dọc của vector chuyển động trên một mẫu theo hướng chiều ngang, và dY[0], dY[1] biểu diễn các độ chênh lệch tương ứng của các phần theo chiều ngang và theo chiều dọc của vector chuyển động trên một mẫu theo hướng chiều dọc; và mv_center[0] và mv_center[1] biểu diễn các phần theo chiều ngang và theo chiều dọc của vector chuyển động cho điểm trung tâm một cách tương ứng.

19. Bộ giải mã theo điểm bất kỳ trong số các điểm 17 và 18, trong đó bước xác định khoảng vector chuyển động thứ hai dựa vào vector chuyển động cho điểm trung tâm của khối được

lập mã afin và các dịch vị được định rõ trước cho kích thước khối tương ứng được thực hiện bởi các phương trình sau đây:

$$mv_hor_min = mv_center[0] - deviationMV[\log_2 CbWidth - 3]$$

$$mv_ver_min = mv_center[1] - deviationMV[\log_2 CbHeight - 3]$$

$$mv_hor_max = mv_center[0] + deviationMV[\log_2 CbWidth - 3]$$

$$mv_ver_max = mv_center[1] + deviationMV[\log_2 CbHeight - 3]$$

trong đó $deviationMV[]$ biểu diễn bảng của các dịch vị được định rõ trước cho kích thước khối tương ứng, và mv_hor_min , mv_ver_min , mv_hor_max và mv_ver_max biểu diễn khoảng vector chuyển động thứ hai.

20. Bộ giải mã theo điểm bất kỳ trong số các điểm từ 17 và 18, trong đó các dịch vị được định rõ trước cho các kích thước khối tương ứng tỷ lệ với { 64, 128, 272, 560, 1136 }, trong đó

- 64 tương ứng với kích thước khối có kích thước tương ứng bằng 8,
- 128 tương ứng với kích thước khối có kích thước tương ứng bằng 16,
- 272 tương ứng với kích thước khối có kích thước tương ứng bằng 32,
- 560 tương ứng với kích thước khối có kích thước tương ứng bằng 64, hoặc
- 1136 tương ứng với kích thước khối có kích thước tương ứng bằng 128.

21. Bộ giải mã theo điểm bất kỳ trong số các điểm từ 16 đến 18, trong đó sau khi xác định khoảng vector chuyển động thứ nhất, khoảng vector chuyển động thứ nhất được định tỷ lệ theo các thông số lấy mẫu phụ $SubWidthC$ và $SubHeightC$:

$$hor_min = hor_min / SubWidthC$$

$$hor_max = hor_max / SubWidthC$$

$$ver_min = ver_min / SubHeightC$$

$$ver_max = ver_max / SubHeightC$$

trong đó hor_min , ver_min , hor_max và ver_max biểu diễn khoảng vector chuyển động thứ nhất được định tỷ lệ.

22. Bộ giải mã theo điểm bất kỳ trong số các điểm từ 16 đến 18, trong đó bộ giải mã này còn được tạo cấu hình để thực hiện các bước:

thu nhận các mẫu được nội suy trong ảnh tham chiếu dựa vào trường vector chuyển động được dẫn ra nhờ sử dụng nội suy song tuyến tính; và

áp dụng bộ lọc thông cao cho các mẫu được nội suy.

23. Bộ giải mã theo điểm bất kỳ trong số các điểm từ 16 đến 18, trong đó khối con của khối được lập mã afin có kích thước 4x4.

24. Bộ giải mã theo điểm bất kỳ trong số các điểm từ 16 đến 18, trong đó ngưỡng được định rõ trước là 72.

25. Bộ giải mã theo điểm bất kỳ trong số các điểm từ 16 đến 18, trong đó trong trường hợp rằng việc liên dự đoán afin bao gồm việc dự đoán đôi, bước thiết đặt biến clipMVX bao gồm:

dẫn ra biến clipMV0 cho danh mục 0

dẫn ra biến clipMV1 cho danh mục 1

dẫn ra biến clipMV là clipMV0 | clipMV1, trong đó “|” nghĩa là HOẶC.

1/8

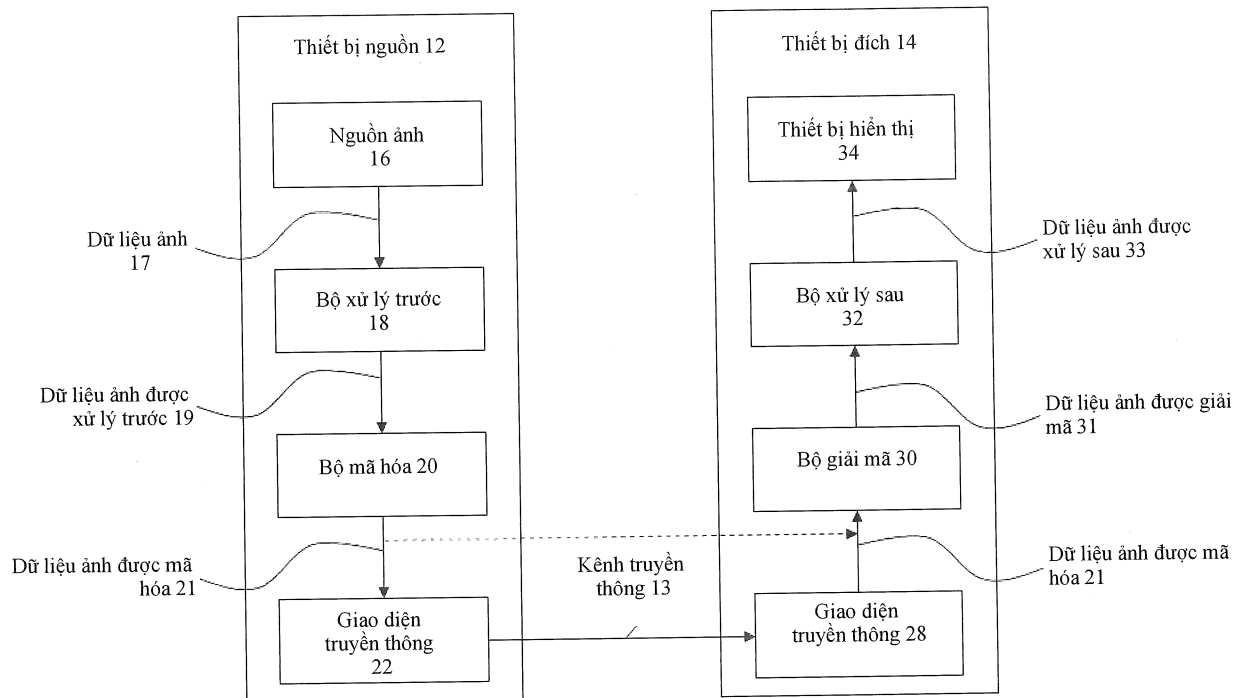


FIG.1A

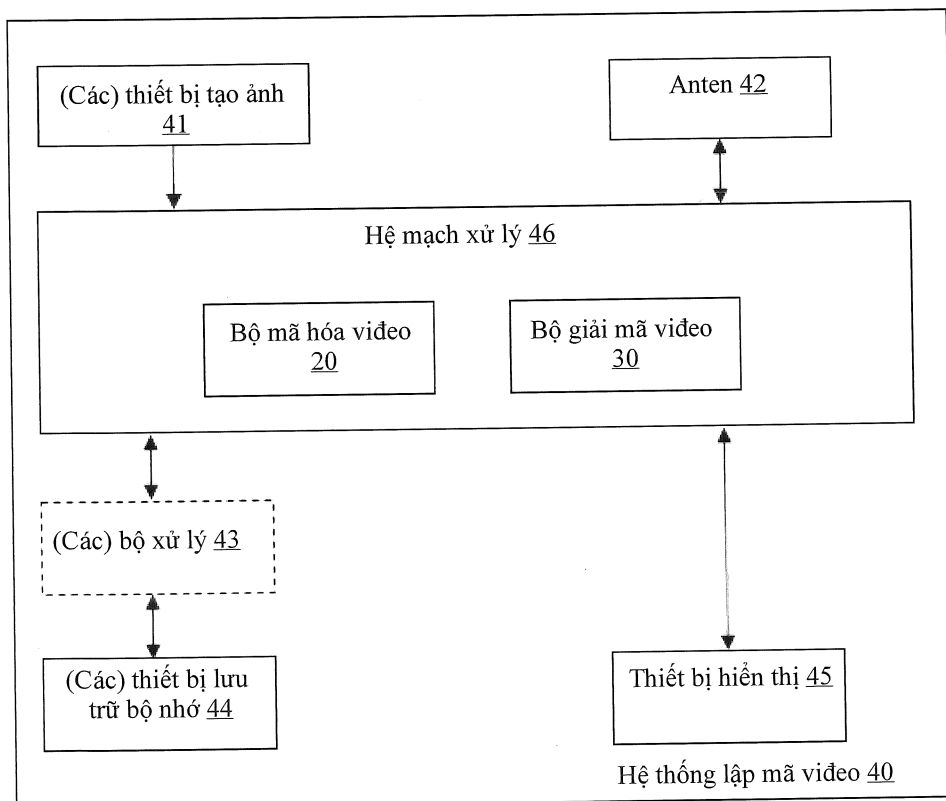


FIG.1B

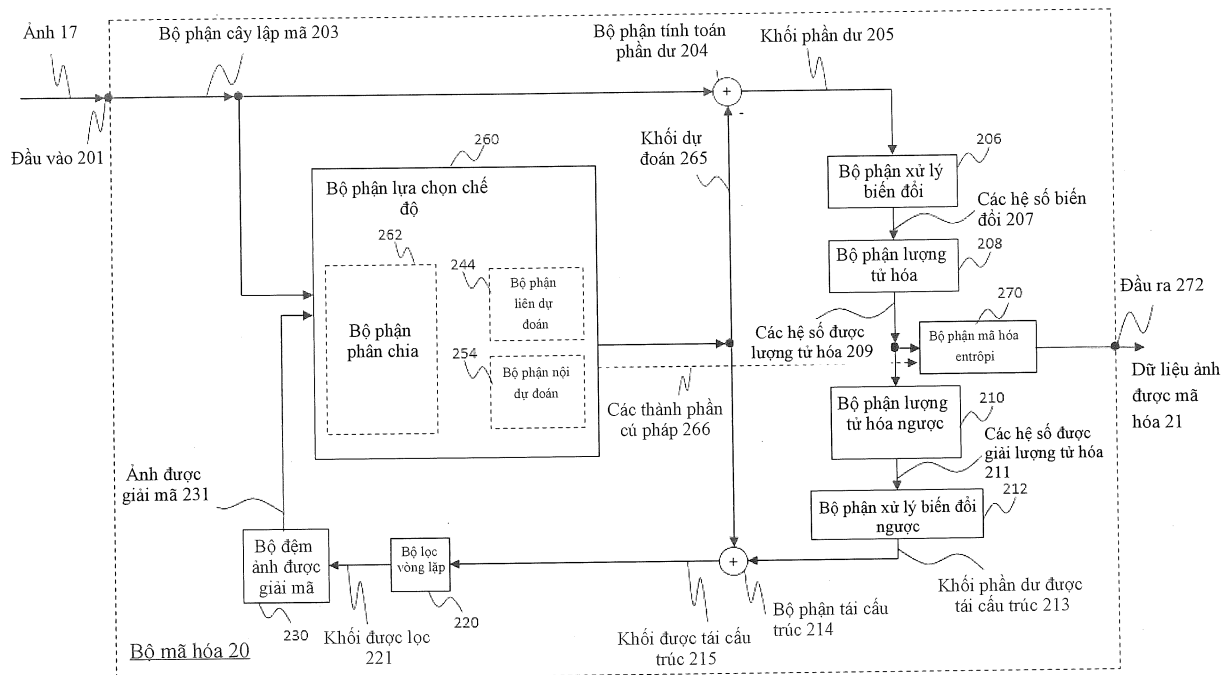


FIG. 2

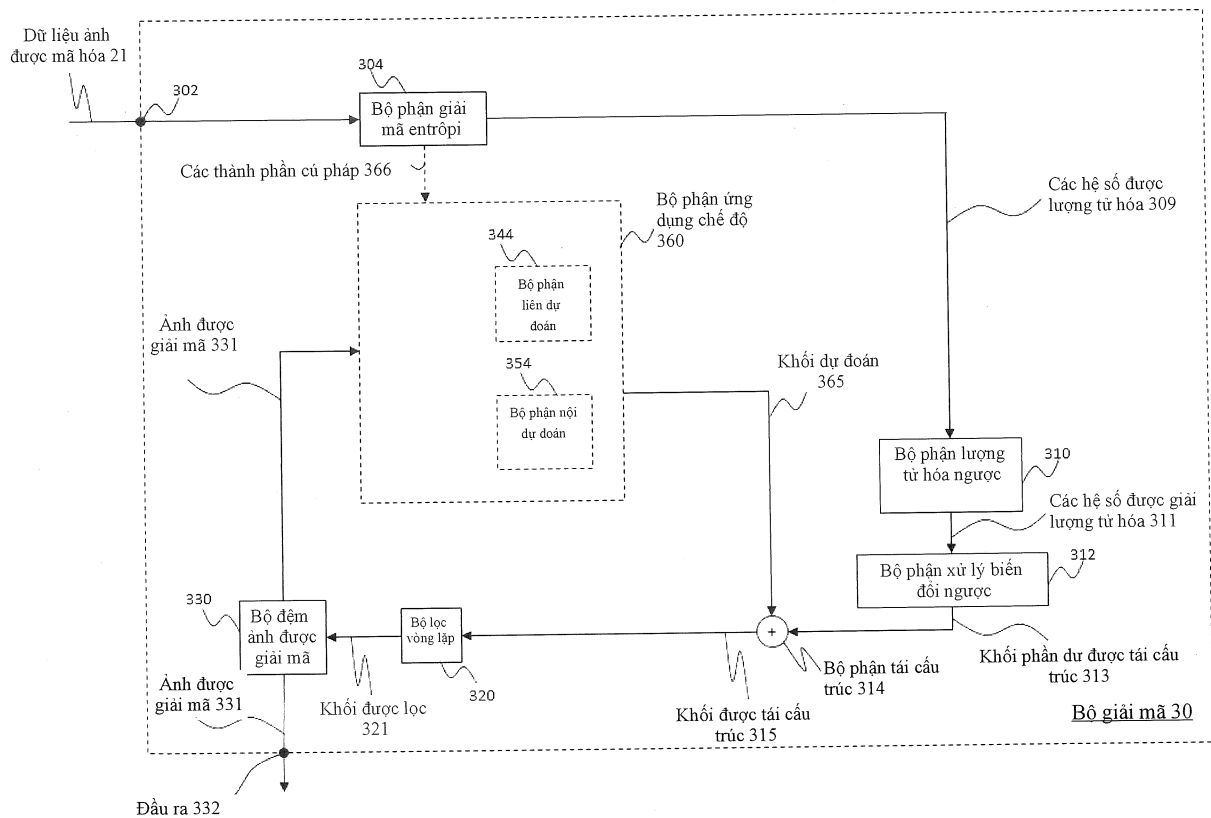


FIG. 3

3/8

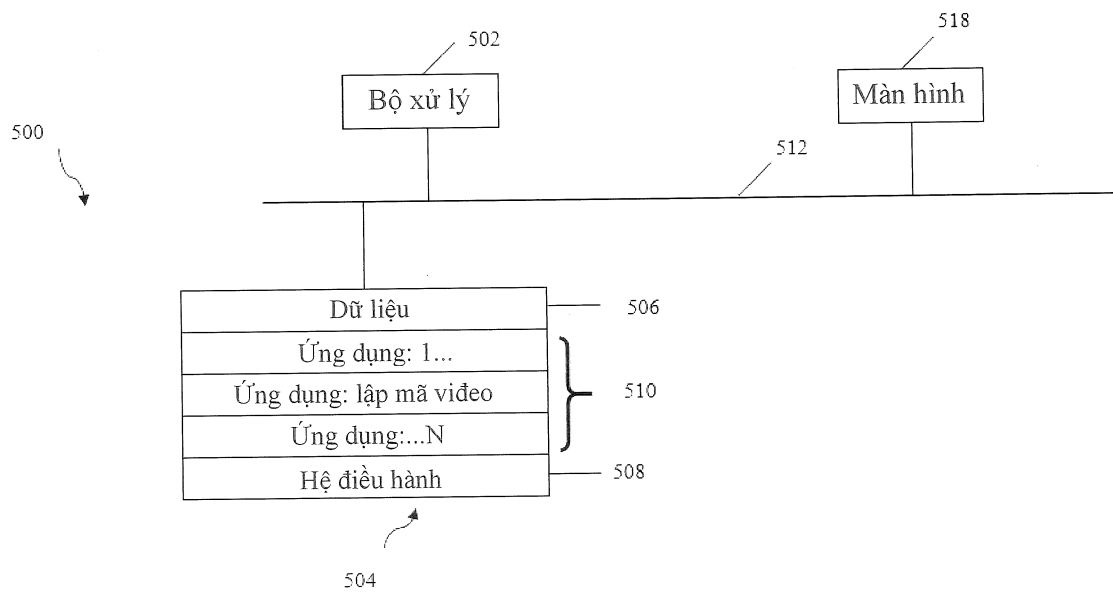
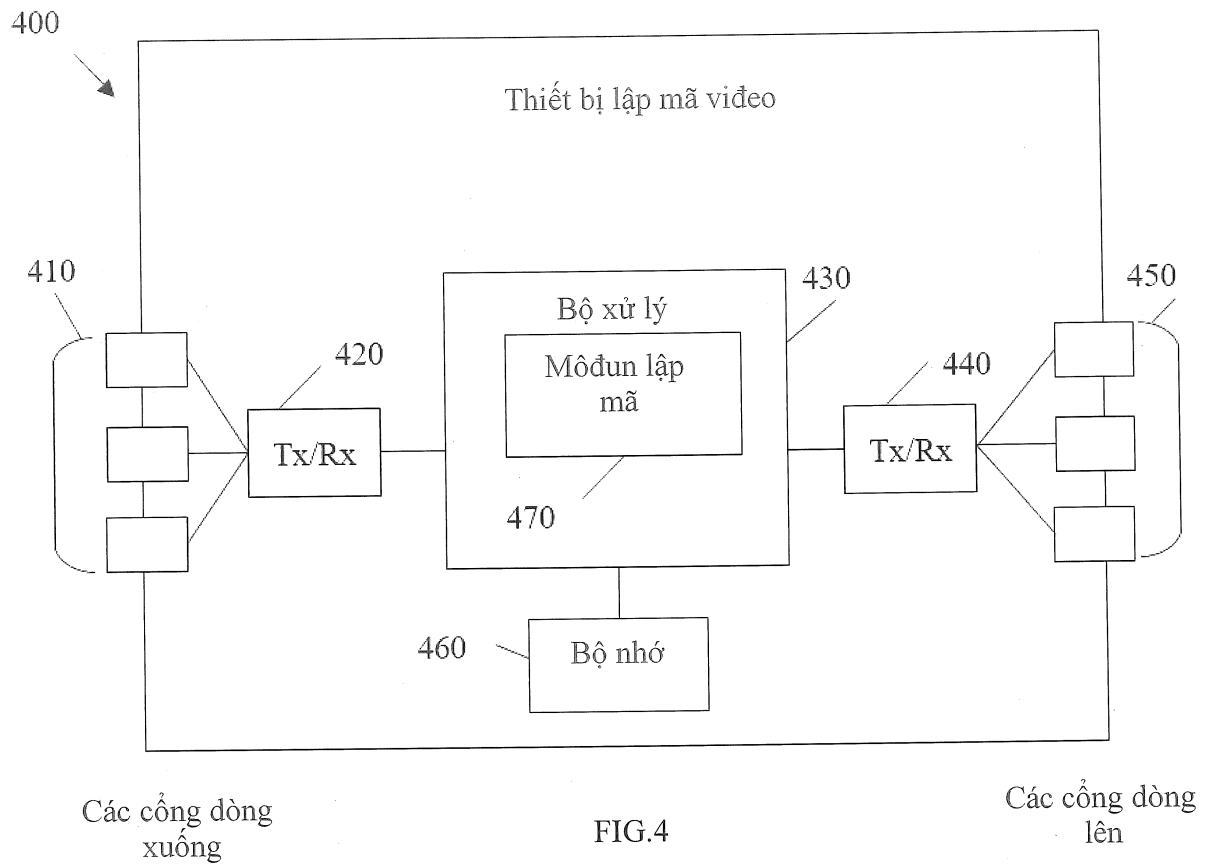
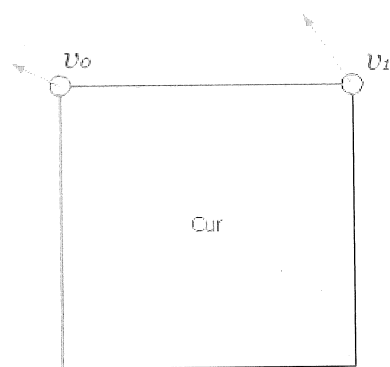
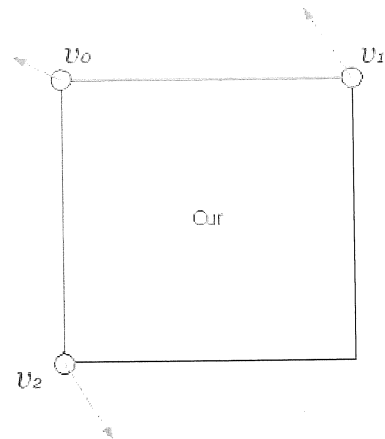


FIG.5

4/8

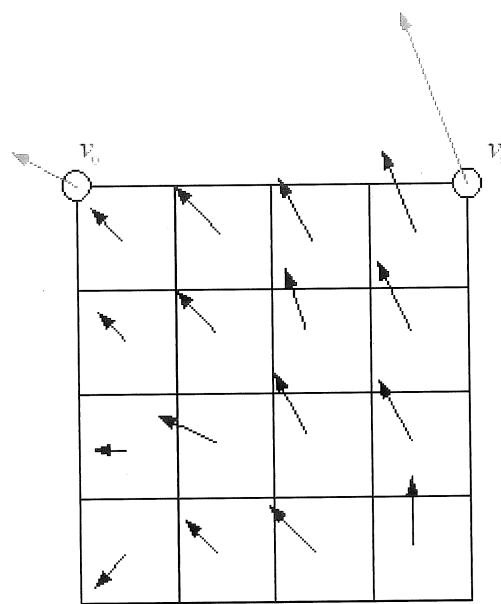


(a) Mô hình afin 4 thông số



(b) Mô hình afin 6 thông số

FIG.6



Trường vector chuyển động khối con afin

FIG.7

5/8

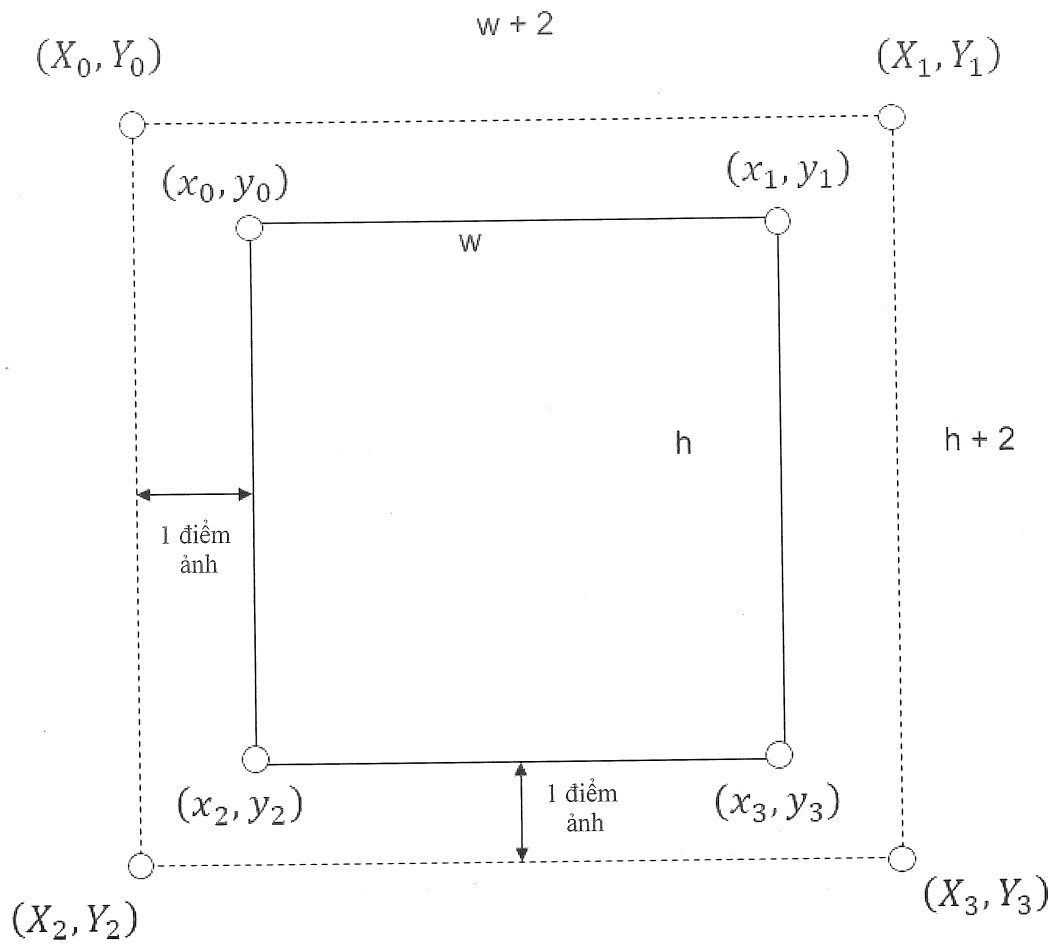


FIG.8

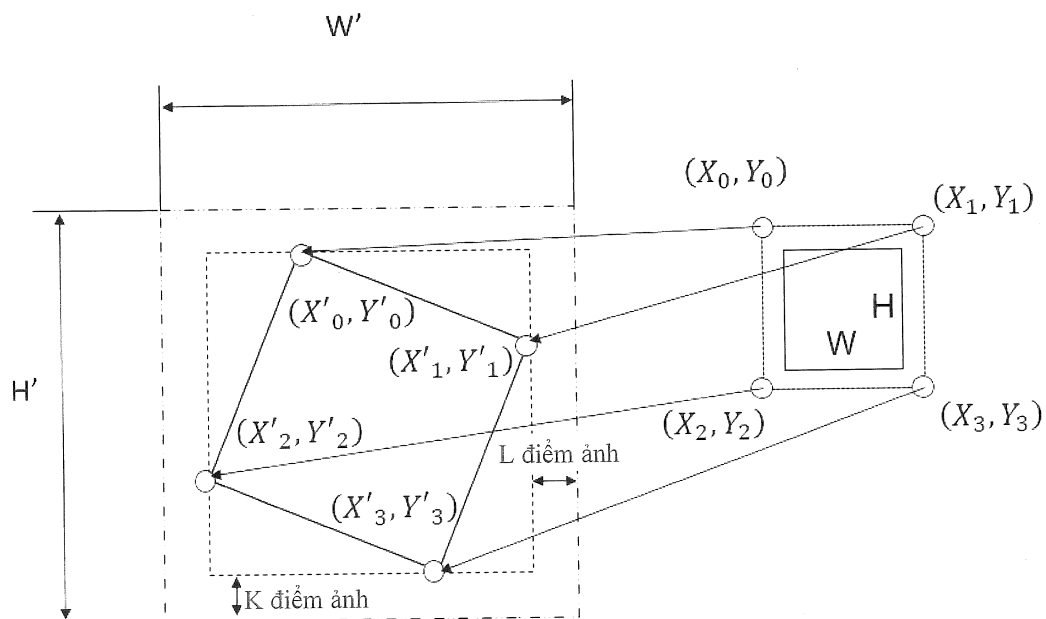


FIG.9

6/8

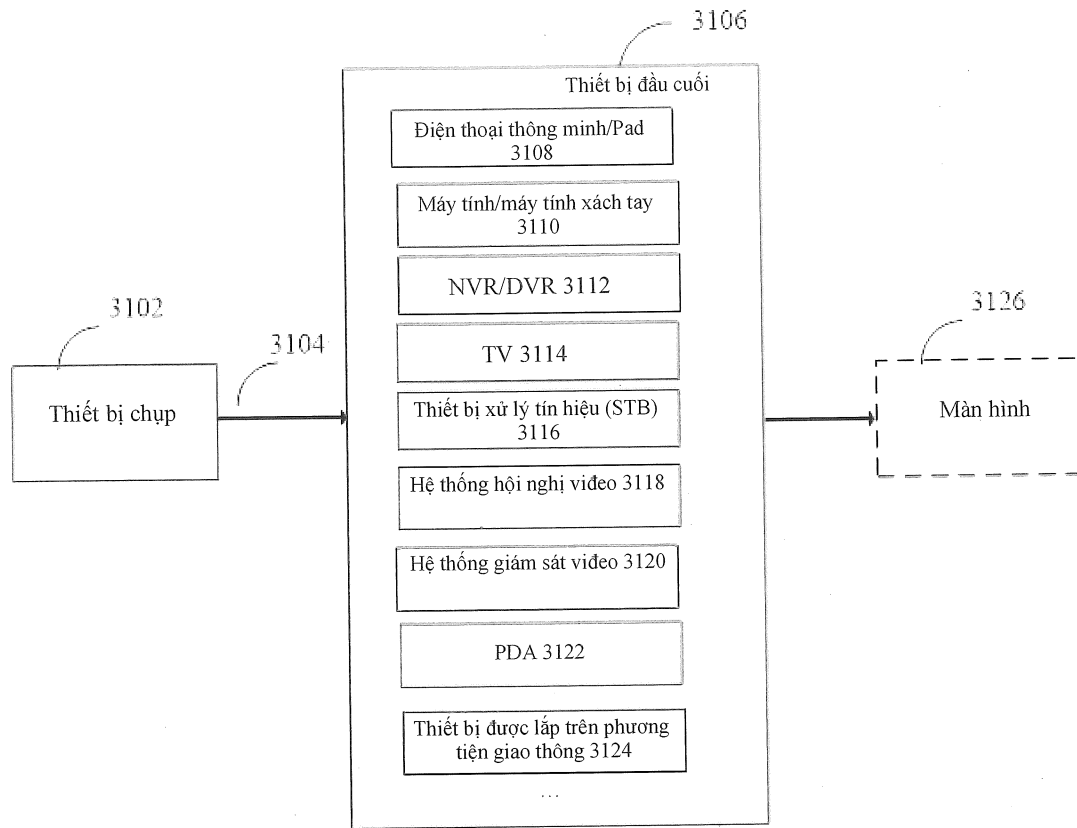


FIG.10

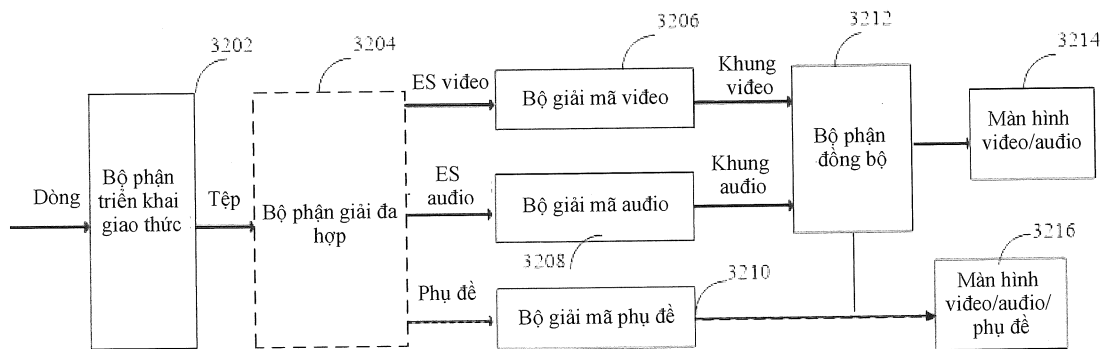


FIG.11

7/8

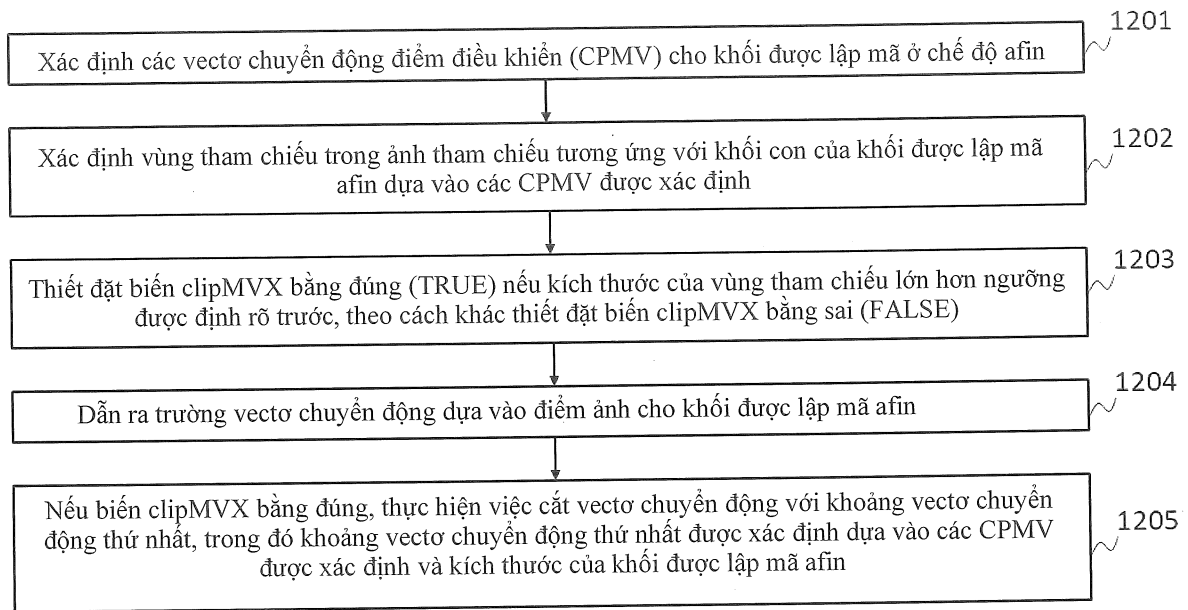


FIG.12

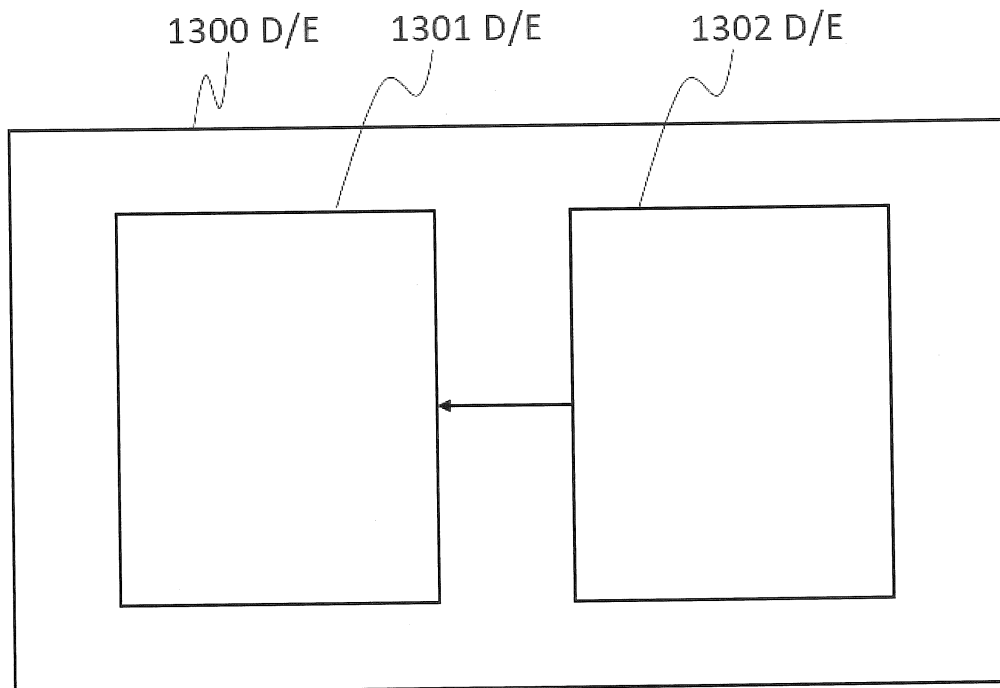


FIG.13

8/8

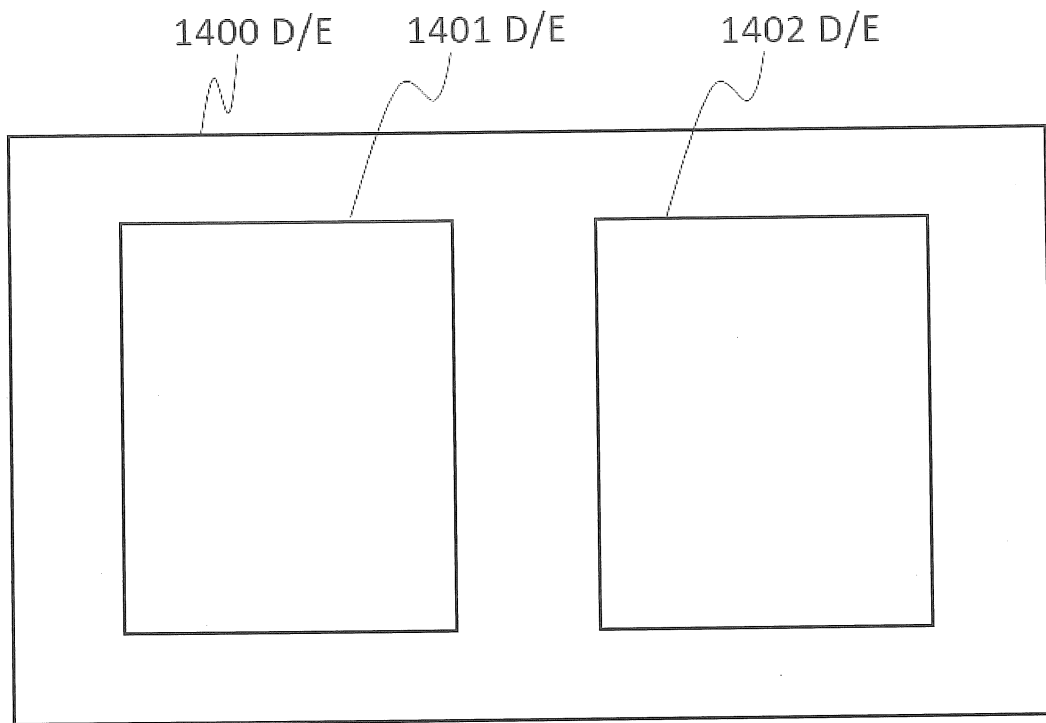


FIG.14