



(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
1-0050120
(51)^{2020.01} H04N 19/117; H04N 19/82; H04N 19/186 (13) B

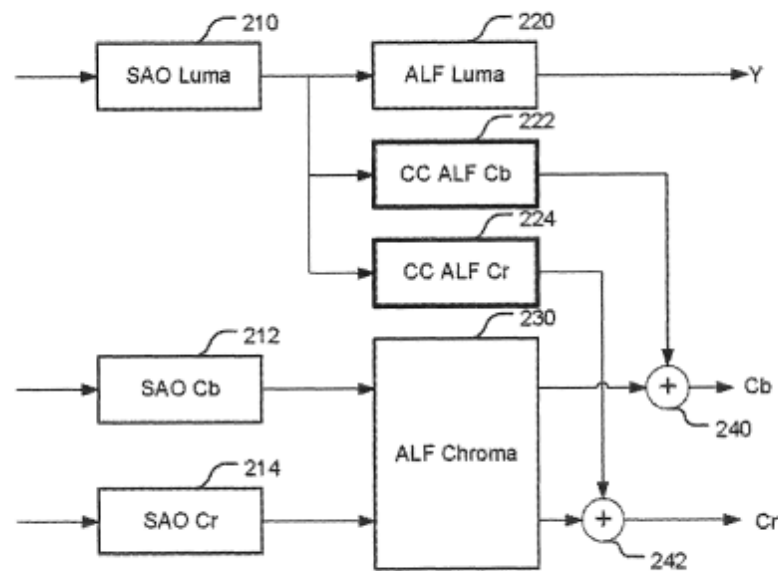
(21) 1-2022-01784 (22) 16/09/2020
(86) PCT/CN2020/115570 16/09/2020 (87) WO2021/052361 25/03/2021
(30) 62/900,702 16/09/2019 US; 62/940,321 26/11/2019 US; 62/954,024 27/12/2019 US;
62/956,356 02/01/2020 US; 17/021,050 15/09/2020 US
(45) 25/08/2025 449 (43) 25/07/2022 412A
(73) HFI INNOVATION INC. (TW)
3F.-7, No.5, Taiyuan 1st St., Zhubei City, Hsinchu County 302, Taiwan
(72) CHEN, Ching-Yeh (TW); CHUBACH, Olena (UA); CHUANG, Tzu-Der (TW);
HSU, Chih-Wei (TW); HUANG, Yu-Wen (TW).
(74) Công ty TNHH Trường Xuân (AGELESS CO.,LTD.)

(54) PHƯƠNG PHÁP LẬP MÃ VIDEO VÀ THIẾT BỊ LẬP MÃ VIDEO

(21) 1-2022-01784

(57) Sáng chế đề cập đến phương pháp lập mã video và thiết bị lập mã video. Theo một phương pháp, tổng các hệ số lọc cho mỗi một trong số một hoặc nhiều phép lọc vòng lặp thành phần màu chéo ứng viên được ràng buộc là trị số cố định. Một hoặc nhiều thành phần cú pháp biểu thị trị số liên quan đến tổng số các phép lọc ứng viên trong bộ phép lọc được báo hiệu hoặc được phân tích trong bộ tham số thích ứng APS (Adaptation Parameter Set).

Fig. 2A



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phép lọc vòng lặp thích ứng ALF (Adaptive Loop Filter - ALF) trong hệ thống lập mã video. Cụ thể hơn là, sáng chế đề cập đến xử lý ALF thành phần chéo ràng buộc và phương pháp lập mã video và thiết bị lập mã video.

Tình trạng kỹ thuật của sáng chế

Ước lượng chuyển động là kỹ thuật lập mã liên khung hình hiệu quả nhằm khai thác sự sai lệch thời gian (sự dư thời gian) trong các chuỗi video. Lập mã liên khung hình được bù chuyển động đã được sử dụng rộng rãi trong nhiều tiêu chuẩn lập mã video quốc tế khác nhau. Ước lượng chuyển động được sử dụng trong nhiều tiêu chuẩn lập mã khác nhau thường là kỹ thuật dựa vào khối, trong đó thông tin chuyển động chẳng hạn cơ chế lập mã và vectơ chuyển động được xác định cho từng khối vĩ mô hoặc cấu hình khối tương tự. Ngoài ra, lập mã nội ảnh cũng được ứng dụng một cách thích ứng, trong đó hình ảnh được xử lý mà không tham chiếu đến bất kỳ hình ảnh nào khác. Các phần dư được dự đoán nội ảnh hoặc được dự đoán liên ảnh thường được xử lý tiếp bằng phép biến đổi, phép lượng tử hóa, và lập mã entropi để tạo ra luồng bit video được nén. Trong quá trình mã hóa, các giả tạo lập mã bị sinh ra, nhất là trong quá trình lượng tử hóa. Để làm giảm các giả tạo lập mã, quá trình xử lý phụ được áp dụng cho video tái cấu trúc nhằm nâng cao chất lượng hình ảnh trong các hệ thống lập mã mới hơn. Quá trình xử lý phụ vừa nêu thường được tích hợp trong phép toán vòng lặp để cho bộ mã hóa và bộ giải mã có thể suy diễn ra các hình ảnh tham chiếu giống nhau nhằm giúp cho hiệu suất hệ thống được cải thiện.

Fig. 1A minh họa hệ thống lập mã video nội ảnh/liên ảnh thích ứng được lấy làm ví dụ mà tích hợp phép xử lý vòng lặp kín có phép lọc vòng lặp thích ứng (ALF). Đối với dự đoán liên ảnh, ước lượng chuyển động (ME)/bù chuyển động (MC) 112 được sử dụng để tạo ra dữ liệu dự đoán dựa vào dữ liệu video từ một hình ảnh hoặc nhiều hình ảnh khác. Chuyển mạch 114 chọn dữ liệu dự đoán nội ảnh 110 hoặc dự đoán liên ảnh và dữ liệu dự đoán được chọn đó được cấp đến bộ cộng 116 để tạo ra các sai số dự đoán, hay còn gọi là các phần dư. Sai số dự đoán sau đó được xử lý bằng phép biến đổi T (Transformation - T) 118 và tiếp theo được xử lý bằng phép lượng tử hóa Q

(Quantization - Q) 120. Các phần dư được biến đổi và được lượng tử hóa sau đó được lập mã bằng bộ mã hóa entropi 122 để tạo ra luồng bit video tương ứng với dữ liệu video được nén. Dòng bit liên quan đến các hệ số biến đổi sau đó được chèn vào các thông tin phụ chẳng hạn chuyển động, cơ chế, và các thông tin khác liên quan đến vùng hình ảnh. Thông tin phụ có thể cũng được lập mã entropi để giảm băng thông mong muốn. Theo đó, dữ liệu liên quan đến thông tin phụ được cấp đến bộ mã hóa entropi 122 như được thể hiện trên Fig. 1A. Khi cơ chế dự đoán liên ảnh được sử dụng, một hình ảnh hoặc nhiều hình ảnh tham chiếu cũng phải được tái cấu trúc tại bộ mã hóa. Do vậy, các phần dư được biến đổi và được lượng tử hóa đó được xử lý bằng phép lượng tử hóa ngược IQ (Inverse Quantization - IQ) 124 và phép biến đổi ngược IT (Inverse Transformation - IT) 126 để khôi phục các phần dư. Các phần dư sau đó được cộng trở lại vào dữ liệu dự đoán 136 tại bộ tái cấu trúc REC (Reconstruction - REC) 128 để tái cấu trúc dữ liệu video. Dữ liệu video tái cấu trúc có thể được lưu ở bộ đệm hình ảnh tham chiếu 134 và được sử dụng để dự đoán các khung hình khác.

Như được thể hiện trên Fig. 1A, dữ liệu video đầu vào trải qua một loạt quá trình xử lý trong hệ thống mã hóa. Dữ liệu video tái cấu trúc từ REC 128 có thể có thể bị nhiều suy yếu do loạt quá trình xử lý vừa nêu. Theo đó, nhiều phép xử lý vòng lặp kín được áp dụng cho dữ liệu video tái cấu trúc trước khi dữ liệu video tái cấu trúc đó được lưu ở bộ đệm hình ảnh tham chiếu 134 để nâng cao chất lượng video. Ở các hệ thống lập mã video, nhiều phép lọc vòng lặp kín khác nhau chẳng hạn phép lọc tách khối DF (Deblocking Filter - DF) 130, phép bù thích ứng mẫu SAO (Sample Adaptive Offset - SAO) 131 và phép lọc vòng lặp thích ứng (ALF) 132 được sử dụng nhằm nâng cao chất lượng hình ảnh.

Bộ giải mã tương ứng dùng cho bộ mã hóa của Fig. 1A được thể hiện trên Fig. 1B. Dòng bit video được giải mã bằng bộ giải mã entropi 142 để khôi phục các phần dư được biến đổi và được lượng tử hóa. Tại bộ giải mã, chỉ phép bù chuyển động (MC) 113 được thực hiện thay cho ME/MC. Quá trình giải mã vừa nêu là tương tự với vòng lặp tái cấu trúc tại bộ mã hóa. Các phần dư được biến đổi và được lượng tử hóa đã được phục hồi ở trên, thông tin SAO/ALF và thông tin hệ thống khác được sử dụng để tái cấu trúc dữ liệu video. Video tái cấu trúc được xử lý tiếp bằng DF 130, SAO 131 và ALF 132 để tạo ra video được giải mã được cải thiện cuối cùng.

Xử lý ALF thành phần chéo

Theo JVET-O0636 (Kiran Misra, và các cộng sự, “Phép lọc vòng lặp thích ứng thành phần chéo cho chroma”, Nhóm chuyên gia lập mã video chung JVET (Joint Video Experts Team - JVET) của ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 15: Gothenburg, SE, từ ngày 3 đến 12 tháng 7 năm 2019, Tài liệu: JVET-O0636), Phép lọc vòng lặp thích ứng thành phần chéo CC-ALF (Cross-Component Adaptive Loop Filter - CC-ALF). CC-ALF sử dụng các trị số mẫu độ chói (mẫu luma) để lọc từng thành phần độ sắc (chroma). Fig. 2A minh họa cấu trúc của CC-ALF tương ứng với các phép lọc vòng lặp khác theo JVET-O0636. Trên Fig. 2A, xử lý ALF được thực hiện được thực hiện sau phép SAO (210, 212 và 214) tương ứng. Ở hệ thống lập mã video thông thường, chỉ ALF Luma 220 và ALF Chroma 230 được sử dụng. Theo JVET-O0636, hai phép lọc ALF phụ, tức là, CC ALF Cb 222 và CC ALF Cr 224 được sử dụng để tạo ra các tín hiệu điều chỉnh để chèn vào Cb và Cr được xử lý ALF bằng cách sử dụng các bộ cộng 240 và 242 một cách tương ứng.

CC-ALF thực thi bằng cách áp dụng phép lọc dạng hình thoi, đường thẳng đối với kênh luma cho từng thành phần chroma như được thể hiện trên Fig. 2B theo JVET-O0636. Các hệ số lọc được truyền theo APS, được phóng tỷ lệ với thừa số 2^{10} , và được làm tròn cho biểu diễn điểm cố định. Việc áp dụng các phép lọc được điều khiển theo kích cỡ khối biến thiên và được báo hiệu bằng cờ được lập mã nội dung nhận được cho từng khối của các mẫu. Kích cỡ khối cùng với cờ kích hoạt CC-ALF được nhận tại phân mức lát (slice) cho từng thành phần chroma. Cú pháp và các ngôn ngữ cho CC-ALF cũng được bộc lộ trong JVET-O0636.

Như được nêu ở trên, CC-ALF thực thi bằng cách áp dụng phép lọc dạng hình thoi, đường thẳng đối với kênh luma cho từng thành phần chroma. Fig. 2C minh họa ví dụ về cách bố trí của CC-ALF 250 theo các mẫu luma. Phần chú thích 260 của các ký hiệu được thể hiện, trong đó vòng tròn rộng thể hiện mẫu luma và vòng tròn chứa các đường gạch chéo thể hiện mẫu chroma. Ví dụ, các phép lọc dạng hình thoi 253 và 255 được thể hiện cho các mẫu chroma 252 và 254 một cách tương ứng. Dải động hệ số lọc được giới hạn là 6 bit. Phép chọn lọc được thực hiện ở phân mức CTU có hỗ trợ cho tối đa 4 phép lọc. Phép chọn đường đối xứng được sử dụng tại biên giới ảo (để tương thích hơn với ALF). Các vùng đệm hệ số lớp thời gian không được sử dụng (để tương thích

hơn với khái niệm APS). Tín hiệu ra dư được cắt xén từ $-2\text{BitDepthC}-1$ thành $2\text{BitDepthC}-1-1$.

Việc áp dụng các phép lọc được điều khiển theo kích cỡ khối biến thiên và được báo hiệu bằng cờ được lập mã nội dung nhận được cho từng khối của các mẫu. Kích cỡ khối cùng với cờ kích hoạt CC-ALF được nhận tại phân mức slice cho từng thành phần chroma. Cú pháp và các ngôn ngữ cho CC-ALF được trình bày trong phần Phụ lục. Theo đó, các kích cỡ khối sau đây (ở các mẫu chroma) được hỗ trợ 16x16, 32x32, 64x64, 128x128.

Các biên giới ảo giúp giảm việc sử dụng bộ đệm dòng

Để loại bỏ các yêu cầu về bộ đệm dòng của SAO và ALF, chúng tôi đưa ra khái niệm biên giới ảo VB (Virtual Boundary - VB). Như được thể hiện trên Fig. 3, các VB được dịch chuyển hướng lên trên so với các biên giới đơn vị lập mã lớn nhất LCU (Largest Coding Unit - LCU) nằm ngang một khoảng N điểm ảnh. Đối với từng LCU, SAO và ALF có thể xử lý các điểm ảnh phía trên VB trước khi LCU bên dưới đến nhưng không thể xử lý các điểm ảnh phía dưới VB cho đến khi LCU bên dưới đến, điều mà bị gây ra bởi DF. Khi xét đến DF trong HM-4.0, không gian giữa VB và biên giới LCU nằm ngang được thiết lập làm ba điểm ảnh cho luma (tức là $N=3$ trên Fig. 3) và một điểm ảnh cho chroma (tức là $N=1$ trên Fig. 3). SAO và ALF được sửa đổi để đảm bảo sao cho mỗi điểm ảnh sẽ được xử lý ở một phía của VB không yêu cầu bất kỳ dữ liệu nào truy cập từ phía còn lại của VB trừ khi dữ liệu đó có thể luôn hiện hữu theo thời gian mà không sử dụng bất kỳ bộ đệm dòng phụ nào.

Theo hội nghị lần thứ 11 của JVET, ALF có phép thích ứng dựa vào khối 4x4 được sử dụng trong dự thảo tiêu chuẩn lập mã video thế hệ tiếp theo, lập mã video đa năng VVC (Versatile Video Coding - VVC). Đối với một khối 4x4, một cửa sổ 8x8 được yêu cầu để tính toán đặc tính khối và dấu hiệu lọc là một phép lọc hình thoi 7x7 cho luma và một phép lọc hình thoi 5x5 cho chroma. Vì vậy, bộ đệm dòng yêu cầu mà không có các biên giới ảo là 7 bộ đệm dòng luma và 3 bộ đệm dòng chroma ở trên đỉnh của 4 bộ đệm dòng DF luma và 2 bộ đệm dòng DF chroma. Ý tưởng của các biên giới ảo có thể còn được áp dụng trong VVC để giảm việc sử dụng bộ đệm dòng. Ở phần mô tả nêu trên, đơn vị lập mã lớn nhất (LCU) được sửa tên thành đơn vị cây lập mã CTU (Coding Tree Unit - CTU) trong VVC. Tuy nhiên, một số sửa đổi được đề xuất để phù hợp với những

đặc tính của thiết kế ALF hiện tại.

Các thuật toán ALF phi tuyến tính

Theo mô hình kiểm nghiệm lập mã video đa năng phiên bản 3.0 VTM3.0 (Versatile Video Coding Test Model Ver. 3.0 – VTM3.0), quá trình lọc của phép lọc vòng lặp thích ứng, được thực hiện như sau:

$$O(x, y) = \sum_{(i,j)} w(i, j) \cdot I(x + i, y + j), (1)$$

Ở công thức nêu trên, các mẫu $I(x + i, y + j)$ là các mẫu đầu vào, $O(x, y)$ là mẫu ra được lọc (tức là kết quả lọc), và $w(i, j)$ biểu thị các hệ số lọc. Vì ALF được áp dụng cho các mẫu được tái cấu trúc, các mẫu $I(x + i, y + j)$ tương ứng với các mẫu được tái cấu trúc. Điểm ảnh được tái cấu trúc chính giữa tại vị trí chính giữa của xử lý ALF tương ứng với $(i, j) = (0, 0)$, tức là, $I(i, j)$. Trong thực tế, theo VTM3.0, nó được thực hiện bằng cách sử dụng thuật toán số nguyên cho các tính toán chính xác điểm cố định:

$$O(x, y) = \left(\sum_{i=-\frac{L}{2}}^{\frac{L}{2}} \sum_{j=-\frac{L}{2}}^{\frac{L}{2}} w(i, j) \cdot I(x + i, y + j) + 64 \right) \gg 7, (2)$$

Ở công thức nêu trên, L biểu thị độ dài lọc, và trong đó $w(i, j)$ là các hệ số lọc theo độ chính xác điểm cố định.

Công thức (1) có thể có thể được viết lại, mà không ảnh hưởng đến hiệu suất lập mã, theo biểu thức dưới đây:

$$O(x, y) = I(x, y) + \sum_{(i,j) \neq (0,0)} w(i, j) \cdot (I(x + i, y + j) - I(x, y)). (3)$$

Ở công thức nêu trên, $w(i, j)$ là các hệ số lọc giống như công thức (1) ngoại trừ $w(0, 0)$, bằng 1 trong công thức (3) trong khi nó bằng $(1 - \sum_{(i,j) \neq (0,0)} w(i, j))$ trong công thức (1).

Bằng cách sử dụng hệ số lọc của công thức (3), chúng ta có thể dễ dàng đưa sự phi tuyến tính vào để giúp cho ALF hiệu quả hơn bằng cách sử dụng hàm cắt đơn nhằm giảm tác động của các trị số mẫu liền kề ($I(x + i, y + j)$) khi chúng quá sai lệch với trị số mẫu hiện thời ($I(x, y)$) đang được lọc.

Theo JVET-M0385 ((J. Taquet, và các cộng sự, “Phép lọc vòng lặp thích ứng phi tuyến tính”, Nhóm chuyên gia lập mã video chung JVET của ITU-T SG 16 WP 3 và

ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 13: Marrakech, MA, từ 9 đến 18 tháng 1 năm 2019, Tài liệu: JVET- M0385)), ALF phi tuyến tính được bộc lộ. Tài liệu này đưa ra thuật toán cắt thích ứng đối với các trị số mẫu đầu vào của phép lọc vòng lặp thích ứng trong phần mềm kiểm nghiệm VTM3.0. Mục đích của việc cắt xén thích ứng này là nhằm đưa các đặc tính phi tuyến để hạn chế sự sai lệch giữa trị số mẫu đầu vào sẽ được lọc và các trị số mẫu đầu vào liên kế khác của phép lọc.

Theo JVET-M0385, thuật toán lọc ALF được sửa đổi như sau:

$$O'(x, y) = I(x, y) + \sum_{(i,j) \neq (0,0)} w(i, j) \cdot K(I(x + i, y + j) - I(x, y), k(i, j)), (4)$$

Ở công thức nêu trên, $O'(x, y)$ tương ứng với kết quả phép lọc ALF sửa đổi, $I(x, y)$ tương ứng với mẫu trước khi ALF, $K(d, b) = \min(b, \max(-b, d))$ là hàm cắt, và $k(i, j)$ là các tham số cắt, mà phụ thuộc vào hệ số lọc tại (i, j) . Bộ mã hóa thực hiện phép tối ưu hóa để tìm ra $k(i, j)$ tốt nhất.

Khi triển khai theo JVET-M0385, các tham số cắt $k(i, j)$ được định rõ cho từng phép lọc ALF, trong đó một trị số cắt được báo hiệu cho từng hệ số lọc. Điều đó có nghĩa là 12 trị số cắt được báo hiệu trong luồng bit cho từng phép lọc Luma và 6 trị số cắt cho phép lọc Chroma.

Để hạn chế tổn thất báo hiệu và độ phức tạp mã hóa, các trị số cắt được giới hạn thành bộ nhỏ gồm các trị số khả dĩ theo JVET-M0385. Ngoài ra, chỉ 4 trị số khả dĩ được sử dụng cho các slice liên ảnh và 3 trị số khả dĩ được sử dụng cho các slice nội ảnh.

Vì phương sai của các sai lệch cục bộ thường cao hơn đối với Luma so với Chroma, nên hai bộ khác nhau được sử dụng cho các phép lọc Luma và Chroma. Ngoài ra, trị số mẫu lớn nhất được bao gồm trong từng bộ, để cho phép cắt xén có thể được hủy bỏ nếu nó không cần thiết.

Bộ các trị số cắt được đề xuất theo JVET-M0385 được nêu trong bảng 1

Bảng 1: Các trị số cắt cho phép

	Nội ảnh	Liên ảnh
LUMA	{ 10, 102, 1024 }	{ 6, 32, 181, 1024 }
CHROMA	{ 4, 24, 1024 }	{ 4, 25, 161, 1024 }

Các trị số cắt được mã hóa ở tiêu đề slice bằng cách sử dụng mã hóa Golomb tương ứng với hệ số của trị số cắt trong bộ vừa nêu.

Giảm yêu cầu bộ đệm cho ALF bằng cách sử dụng biên giới ảo

Theo JVET-M0301 ((A. Kotra, và các cộng sự, “Non-CE: Giảm bộ đệm dòng phép lọc vòng lặp”, Nhóm chuyên gia lập mã video chung JVET của ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 13: Marrakech, MA, từ ngày 9 đến 18 tháng 1 năm 2019, Tài liệu: JVET- M0301), cơ chế giảm yêu cầu bộ đệm dòng của ALF (phép lọc vòng lặp thích ứng) được bộc lộ. Tài liệu này sử dụng ý tưởng của các biên giới ảo (các VB) mà được dịch chuyển hướng lên trên so với các biên giới CTU nằm ngang một khoảng “N” mẫu. Phân loại khối ALF sửa đổi và phép lọc ALF sửa đổi được áp dụng cho các mẫu gần với biên giới ảo để giảm số lượng bộ đệm dòng yêu cầu. Phân loại khối ALF sửa đổi chỉ sử dụng các mẫu mà ở phía trên VB để phân loại khối 4 x 4 cho trước, mà ở phía trên VB. Tương tự, để phân loại khối 4 x 4 phía dưới VB, các mẫu thuộc những dòng phía dưới VB được sử dụng. Phép lọc ALF sửa đổi sử dụng sự kết hợp giữa phép hủy có điều kiện và các phiên bản cắt xén của phép lọc ALF gốc.

Phép lọc ALF cắt xén tại các biên giới ảo

Phiên bản cắt xén của các phép lọc được sử dụng để lọc các mẫu Luma thuộc những dòng gần với các biên giới ảo. Những phiên bản cắt xén này của các phép lọc có thể được sử dụng cho cả trường hợp N=4 và N=6. Tương tự, phiên bản cắt xén của các phép lọc cũng được sử dụng cho phép lọc ALF chroma.

Đối với ALF cắt xén, các hệ số phải được chuẩn hóa, tức là, tổng của các hệ số còn lại phải bằng với tổng của các hệ số ban đầu. Fig. 4A đến Fig. 4F minh họa các ví dụ về chuẩn hóa hệ số cho phép xử lý lọc ALF luma cắt xén tại biên giới ảo. Fig. 4A minh họa hệ số ALF sửa đổi cho trường hợp trong đó điểm ảnh trên cùng (tức là, điểm ảnh c0) nằm ngoài biên giới ảo. Hệ số c2 được sửa đổi thành c2', trong đó $c2' = c2 + c0$. Fig. 4B minh họa trường hợp trong đó điểm ảnh dưới cùng (tức là, c0) nằm ngoài biên giới ảo. Fig. 4C minh họa hệ số ALF sửa đổi cho trường hợp trong đó hai hàng trên cùng (tức là, các điểm ảnh c0, c1, c2 và c3) nằm ngoài biên giới ảo. Vì vậy, hệ số c5, c6 và c7 được sửa đổi thành c5', c6' và c7', trong đó $c5' = c5 + c1$, $c6' = c6 + c2 + c0$, và $c7' = c7 + c3$. Fig. 4D minh họa hệ số ALF sửa đổi cho trường hợp trong đó hai hàng dưới cùng (tức là,

các điểm ảnh c_0 , c_1 , c_2 và c_3) nằm ngoài biên giới ảo. Fig. 4E minh họa hệ số ALF sửa đổi cho trường hợp trong đó ba hàng trên cùng (tức là, các điểm ảnh từ c_0 đến c_8) nằm ngoài biên giới ảo. Vì vậy, hệ số c_{10} , c_{11} và c_{12} được sửa đổi thành c_{10}' , c_{11}' và c_{12}' , trong đó $c_{10}' = c_{10} + c_4 + c_8$, $c_{11}' = c_{11} + c_5 + c_7 + c_1 + c_3$ và $c_{12}' = c_{12} + 2 \cdot c_6 + 2 \cdot c_2 + 2 \cdot c_0$. Fig. 4F minh họa hệ số ALF sửa đổi cho trường hợp trong đó ba hàng dưới cùng (tức là, các điểm ảnh từ c_0 đến c_8) nằm ngoài biên giới ảo.

Fig. 5A đến 5D minh họa các ví dụ về phép xử lý lọc ALF chroma cắt xén tại biên giới ảo. Fig. 5A minh họa hệ số ALF sửa đổi cho trường hợp trong đó điểm ảnh trên cùng (tức là, điểm ảnh c_0) nằm ngoài biên giới ảo. Hệ số c_2 được sửa đổi thành c_2' , trong đó $c_2' = c_2 + c_0$. Fig. 5B minh họa trường hợp trong đó điểm ảnh dưới cùng (tức là, c_0) nằm ngoài biên giới ảo. Fig. 5C minh họa hệ số ALF sửa đổi cho trường hợp trong đó hai hàng trên cùng (tức là, các điểm ảnh c_0 , c_1 , c_2 và c_3) nằm ngoài biên giới ảo. Vì vậy, hệ số c_5 và c_6 được sửa đổi thành c_5' và c_6' , trong đó $c_5' = c_5 + c_1 + c_3$ và $c_6' = c_6 + 2 \cdot c_2 + 2 \cdot c_0$. Fig. 5D minh họa hệ số ALF sửa đổi cho trường hợp trong đó hai hàng dưới cùng (tức là, các điểm ảnh c_0 , c_1 , c_2 và c_3) nằm ngoài biên giới ảo.

Ở sáng chế này, nhiều vấn đề liên quan đến biên giới ALF khác nhau được giải quyết và các giải pháp nhằm nâng cao hiệu suất, nhằm hợp nhất nhiều hệ thống hoặc nhằm đơn giản hóa phép xử lý biên giới ALF được bộc lộ.

Bản chất kỹ thuật của sáng chế

Các phương pháp và thiết bị dùng để xử lý phép lọc vòng lặp video tái cấu trúc được bộc lộ. Theo phương pháp này, các điểm ảnh tái cấu trúc được nhận. Phép lọc vòng lặp thích ứng thành phần màu chéo CCALF mục tiêu (Cross-Colour Adaptive Loop Filter - CCALF) được xác định, trong đó CCALF mục tiêu thuộc bộ phép lọc bao gồm một hoặc nhiều phép lọc ứng viên và tổng các hệ số lọc cho từng phép lọc ứng viên, ở một hoặc nhiều phép lọc ứng viên vừa nêu, được ràng buộc là một trị số cố định. CCALF mục tiêu được áp dụng cho các mẫu luma tái cấu trúc liên đới để tạo ra mẫu chroma được lọc CCALF cho mẫu chroma tái cấu trúc mục tiêu. Mẫu chroma được lọc cuối cùng được tạo ra bằng cách tổ hợp mẫu chroma được lọc CCALF với mẫu chroma được lọc ALF thông thường, trong đó mẫu chroma được lọc cuối cùng được sử dụng làm đầu ra giải mã video hoặc được sử dụng cho quá trình mã hóa hoặc giải mã video tiếp theo.

Theo một phương án, trị số cố định bằng 0. Theo phương án khác, khi số các hệ số lọc đơn trị cho từng phép lọc ứng viên bằng M , thì chỉ $(M-1)$ hệ số lọc cho từng phép lọc ứng viên được giải mã tại bộ giải mã, và trong đó M là số nguyên lớn hơn 1. Theo một phương án, một hệ số lọc, chứ không phải $(M-1)$ hệ số lọc, không được giải mã tại bộ giải mã hoặc được mã hóa tại bộ mã hóa, và hệ số lọc không được lập mã tương ứng với hệ số được chọn bất kỳ trong số các hệ số lọc đơn trị cho từng phép lọc ứng viên.

Theo phương pháp khác, một hoặc nhiều thành phần cú pháp được báo hiệu trong bộ tham số thích ứng APS (Adaptation Parameter Set - APS) của luồng bit video tại bộ mã hóa hoặc một hoặc nhiều thành phần cú pháp vừa nêu được phân tích trong APS của luồng bit video tại bộ giải mã, trong đó một hoặc nhiều thành phần cú pháp vừa nêu biểu thị trị số liên quan đến tổng số các phép lọc ứng viên trong bộ phép lọc. Phép lọc vòng lặp thích ứng thành phần màu chéo CCALF mục tiêu (Cross-Colour Adaptive Loop Filter - CCALF) được xác định, trong đó CCALF mục tiêu thuộc bộ phép lọc bao gồm một hoặc nhiều phép lọc ứng viên và tổng các hệ số lọc cho mỗi một trong số nhiều phép lọc ứng viên được ràng buộc là một trị số cố định. CCALF mục tiêu được áp dụng cho các mẫu luma tái cấu trúc liên đới để tạo ra mẫu chroma được lọc CCALF cho mẫu chroma tái cấu trúc mục tiêu. Mẫu chroma được lọc cuối cùng được tạo ra bằng cách tổ hợp mẫu chroma được lọc CCALF với mẫu chroma được lọc ALF thông thường, trong đó mẫu chroma được lọc cuối cùng được sử dụng làm đầu ra giải mã video hoặc được sử dụng cho quá trình mã hóa hoặc giải mã video tiếp theo.

Theo một phương án, một thành phần cú pháp được báo hiệu hoặc được phân tích cú pháp cho từng thành phần màu. Theo một phương án, một hoặc nhiều thành phần cú pháp vừa nêu bao gồm trị số tương ứng với tổng số các phép lọc ứng viên trong bộ phép lọc trừ 1.

Mô tả vắn tắt các hình vẽ

Fig. 1A minh họa hệ thống mã hóa video nội ảnh/liên ảnh thích ứng được lấy làm ví dụ có tích hợp phép xử lý vòng lặp kín DF, SAO và ALF;

Fig. 1B minh họa hệ thống giải mã video nội ảnh/liên ảnh thích ứng được lấy làm ví dụ có tích hợp phép xử lý vòng lặp kín DF, SAO và ALF;

Fig. 2A minh họa cấu trúc của CC-ALF tương ứng với các phép lọc vòng lặp

khác theo JVET-O0636 trong đó xử lý ALF được thực hiện sau phép SAO tương ứng;

Fig. 2B minh họa phép lọc dạng hình thoi được áp dụng đối với kênh luma cho từng thành phần chroma theo JVET-O0636;

Fig. 2C minh họa ví dụ về cách bố trí của CC-ALF 250 theo các mẫu luma;

Fig. 3 minh họa ví dụ về các VB được dịch chuyển lên trên so với các biên giới LCU nằm ngang một khoảng N điểm ảnh;

Fig. 4A đến Fig. 4F minh họa các ví dụ về chuẩn hóa hệ số cho phép xử lý lọc ALF luma cắt xén tại biên giới ảo, trong đó Fig. 4A minh họa hệ số ALF sửa đổi cho trường hợp trong đó điểm ảnh trên cùng nằm ngoài biên giới ảo; Fig. 4B minh họa trường hợp trong đó điểm ảnh dưới cùng nằm ngoài biên giới ảo; Fig. 4C minh họa hệ số ALF sửa đổi cho trường hợp trong đó hai hàng trên cùng nằm ngoài biên giới ảo; Fig. 4D minh họa hệ số ALF sửa đổi cho trường hợp trong đó hai hàng dưới cùng nằm ngoài biên giới ảo; Fig. 4E minh họa hệ số ALF sửa đổi cho trường hợp trong đó ba hàng trên cùng nằm ngoài biên giới ảo; và Fig. 4F minh họa hệ số ALF sửa đổi cho trường hợp trong đó ba hàng dưới cùng nằm ngoài biên giới ảo;

Fig. 5A đến 5D minh họa các ví dụ về phép xử lý lọc ALF chroma cắt xén tại biên giới ảo, trong đó Fig. 5A minh họa hệ số ALF sửa đổi cho trường hợp trong đó điểm ảnh trên cùng nằm ngoài biên giới ảo; Fig. 5B minh họa trường hợp trong đó điểm ảnh dưới cùng nằm ngoài biên giới ảo; Fig. 5C minh họa hệ số ALF sửa đổi cho trường hợp trong đó hai hàng trên cùng nằm ngoài biên giới ảo; và Fig. 5D minh họa hệ số ALF sửa đổi cho trường hợp trong đó hai hàng dưới cùng nằm ngoài biên giới ảo;

Fig. 6 minh họa ví dụ về CCALF dạng hình thoi 3x4 có các hệ số từ c0 đến c6;

Fig. 7 minh họa lưu đồ của phép xử lý ALF của video tái cấu trúc được lấy làm ví dụ theo phương án của sáng chế, trong đó tổng các hệ số lọc cho mỗi một trong số một hoặc nhiều phép lọc ứng viên được ràng buộc là một trị số cố định; và

Fig. 8 minh họa lưu đồ của phép xử lý ALF của video tái cấu trúc được lấy làm ví dụ theo phương án của sáng chế, trong đó một hoặc nhiều thành phần cú pháp biểu thị trị số liên quan đến tổng số các phép lọc ứng viên trong bộ phép lọc được báo hiệu hoặc được phân tích trong APS (Bộ tham số thích ứng).

Mô tả chi tiết sáng chế

Phần mô tả dưới đây là phương án tốt nhất thực hiện sáng chế. Phần mô tả đó được trình bày nhằm mục đích minh họa những nguyên lý chung của sáng chế và không được hiểu là nhằm mục đích giới hạn sáng chế. Phạm vi của sáng chế được xác định tốt nhất bằng cách tham chiếu đến các điểm yêu cầu bảo hộ đính kèm.

Phương pháp 1: Các mẫu luma tương ứng trong CC-ALF

Trong CC-ALF, một phép lọc phụ được chèn vào cho thành phần chroma và tín hiệu ra được lọc sẽ là tổng trọng số của phép lọc chroma gốc ban đầu và phép lọc phụ. Nếu định dạng video là YUV444 hoặc RGB444, thì mẫu luma tương ứng của các mẫu chroma sẽ được xử lý sẽ được định rõ bằng cách sử dụng vị trí có cùng không gian trong các thành phần màu khác nhau. Tuy nhiên, nếu định dạng video là YUV420 hoặc YUV444, thì việc xác định mẫu luma tương ứng trong CC-ALF là một vấn đề. Theo một phương án, mẫu luma tương ứng được giảm tần số lấy mẫu theo định dạng YUV hoặc các vị trí lấy mẫu của chroma. Ví dụ, nếu định dạng video là YUV420 và vị trí lấy mẫu của chroma là $(0, 0,5)$ theo vị trí luma, thì mẫu luma tương ứng có thể là một trong số bốn mẫu luma. Theo ví dụ khác, mẫu luma tương ứng có thể là trung bình của hai mẫu luma bên trái. Theo ví dụ khác, mẫu luma tương ứng có thể là trung bình của các mẫu luma trên cùng và dưới cùng bên trái (bên phải). Trong trường hợp khác, mẫu luma tương ứng có thể là trung bình của bốn mẫu luma. Dấu hiệu lọc có thể được xác định theo vị trí chroma hoặc vị trí luma. Nếu dấu hiệu lọc được xác định theo vị trí các mẫu chroma, thì các tín hiệu vào của phép lọc CC-ALF là các mẫu luma tương ứng được nêu ở trên. Nếu dấu hiệu lọc được xác định theo vị trí các mẫu luma, thì vị trí trung tâm của hình dạng lọc trong CC-ALF được xác định bằng mẫu luma tương ứng của mẫu chroma sẽ được xử lý. Theo một phương án, việc xác định mẫu luma tương ứng phụ thuộc vào định dạng video hoặc vị trí lấy mẫu của thành phần chroma.

Theo phương án khác, thuật toán cắt phi tuyến tính có thể được áp dụng cho CC-ALF. Theo phương án khác nữa, sai lệch giữa các mẫu liên kề và mẫu sẽ được xử lý được sử dụng trong phép lọc thay vì sử dụng các mẫu liên kề một cách trực tiếp. Mẫu sẽ được xử lý có thể là trị số mẫu trong thành phần chroma, hoặc trị số mẫu của mẫu luma

tương ứng trong CC-ALF.

Ở thiết kế hiện tại, tín hiệu ra của ALF chroma sẽ được cắt thành một dải hợp lệ (ví dụ $[0, 1023]$ cho 10 bit nén video và $[0, 255]$ cho 8 bit nén video). Tín hiệu ra của CCALF sẽ được chèn vào phía trên cùng của trị số được cắt này và thuật toán cắt sẽ được thực hiện lại để đảm bảo tín hiệu ra cuối cùng nằm trong dải hợp lệ vừa nêu. Do đó, có hai thuật toán cắt. Một thuật toán cắt được áp dụng cho tín hiệu ra của ALF chroma, và thuật toán cắt còn lại được áp dụng cho tín hiệu ra của CCALF. Theo một phương án, thuật toán cắt được sử dụng để ràng buộc giá trị tín hiệu ra nằm trong dải hợp lệ trong quá trình ALF chroma có thể được loại bỏ khi CC-ALF được áp dụng, và do đó, thuật toán cắt được áp dụng chỉ sau khi tín hiệu ra CC-ALF được chèn vào tín hiệu ra của ALF chroma.

Theo phương án khác, độ chính xác của các hệ số lọc trong phép lọc phụ trong CC-ALF tương thích với thiết kế gốc ban đầu.

Theo một phương án, hình dạng lọc của CC-ALF tương thích với hình dạng lọc của phép lọc ALF luma. Theo một phương án, phép lọc CC-ALF có thể có 25 thể lọc với 23 hệ số, trong đó các hệ số tận cùng phía trên và tận cùng phía dưới bằng nhau và các hệ số tận cùng bên trái và tận cùng bên phải bằng nhau. Theo một phương án, phép lọc CC-ALF có thể có 25 thể lọc với 22 hệ số, trong đó các hệ số tận cùng phía trên, tận cùng phía dưới, tận cùng bên trái và tận cùng bên phải đều bằng nhau. Theo một phương án, phép lọc CC-ALF đối xứng tương xứng với sự đối xứng của phép lọc ALF. Theo một phương án, khi các hình dạng lọc của CCALF và ALF luma giống nhau, thì các phép xử lý biên giới của CCALF, bao gồm phép xử lý biên giới ảo ALF, phép xử lý biên giới ALF tại các biên giới slice/các biên giới brick (biên giới miếng)/các biên giới ngói (biên giới tile)/các biên giới ảo 360 độ/các biên giới hình ảnh con/các biên giới hình ảnh là giống như các phép xử lý biên giới của ALF luma.

Theo một phương án, hình dạng lọc của CC-ALF có thể tương thích với hình dạng lọc của phép lọc ALF chroma. Theo một phương án, phép lọc CC-ALF có thể có 13 thể lọc với 11 hệ số, trong đó các hệ số trên cùng và dưới cùng bằng nhau và các hệ số bên trái và bên phải bằng nhau. Theo một phương án, phép lọc CC-ALF có thể có 13 thể lọc với 10 hệ số, trong đó các hệ số trên cùng, dưới cùng, bên trái và bên phải đều bằng nhau. Theo một phương án, phép lọc CC-ALF là đối xứng, tương tự như phép lọc ALF,

và do đó, nó có 13 thể lọc và chỉ 7 hệ số. Theo một phương án, khi các hình dạng lọc của CCALF và ALF chroma giống nhau, thì các phép xử lý biên giới của CCALF, bao gồm phép xử lý biên giới ảo ALF, phép xử lý biên giới ALF tại các biên giới slice/các biên giới brick/các biên giới tile/các biên giới ảo 360 độ/các biên giới hình ảnh con/các biên giới hình ảnh là giống như các phép xử lý biên giới của ALF chroma.

Theo một phương án, phép lọc CC-ALF được chia sẻ chung giữa các thành phần màu U chroma và V chroma. Theo một phương án, chỉ một cờ `alf_crosscomponent_cbr_filter_signal_flag` xuất hiện trong APS. Theo một ví dụ, nếu cờ này xuất hiện, thì các hệ số CC-ALF xuất hiện trong APS và có thể được giải mã.

Theo một phương án, các hệ số CC-ALF cho U chroma được xác định từ thành phần luma, và có thể được tái sử dụng cho V chroma. Theo một phương án, các hệ số CC-ALF cho V chroma được xác định từ thành phần luma, và có thể được tái sử dụng cho U chroma. Theo một phương án, các hệ số CC-ALF cho V chroma có thể được tạo ra từ các hệ số CC-ALF cho U chroma, bằng cách lập mã dấu và/hoặc thông tin chuyển بیت. Theo một phương án, các hệ số CC-ALF cho U chroma có thể được tạo ra từ các hệ số CC-ALF cho V chroma, bằng cách lập mã dấu và/hoặc thông tin chuyển بیت.

Theo một phương án, CC-ALF có thể được áp dụng cho chỉ một trong số các thành phần chroma. Theo một ví dụ, cờ có thể được báo hiệu hiện ở phân mức CTB, hoặc phân mức CU, hoặc phân mức khối, biểu thị liệu một hay cả hai thành phần chroma sử dụng CC-ALF. Theo một phương án, nếu cờ nêu trên không được báo hiệu, thì các hệ số CC-ALF được xem là bằng không.

Theo một phương án, `slice_cross_component_alf_cb_reuse_temporal_layer_filter` không được báo hiệu ở tiêu đề slice, điều đó nghĩa là phép chọn lọc của CC-ALF phải luôn được báo hiệu cho từng slice, khi CCALF được sử dụng. Theo một phương án, chỉ các hệ số lọc CCALF được lưu trong APS có thể được tham chiếu bởi các slice hoặc các hình ảnh khác. Tức là, không có sự phụ thuộc dữ liệu của các hệ số lọc CCALF giữa các slice hoặc các hình ảnh khác nhau.

Theo phương án khác, nhiều phép lọc CC-ALF có thể được sử dụng trong một lát (slice)/miếng (brick)/ngói (tile)/ảnh (picture). Khi nhiều phép lọc CC-ALF được sử

dụng, việc chọn nhiều phép lọc CC-ALF như vậy có thể được báo hiệu hiện ở phân mức CTB, hoặc phân mức CU, hoặc phân mức khối. Theo phương án khác, việc chọn nhiều phép lọc CC-ALF có thể suy dẫn ẩn theo việc chọn nhiều phép lọc chroma. Khi nhiều phép lọc CC-ALF được sử dụng, chúng có thể được báo hiệu một cách độc lập trong một APS hoặc được liên kết với nhiều phép lọc chroma. Ví dụ, hai bộ gồm nhiều phép lọc, một bộ cho CC-ALF và bộ còn lại cho các phép lọc chroma, được báo hiệu riêng rẽ trong một APS. Hoặc trong trường hợp khác, chỉ một bộ gồm nhiều phép lọc được báo hiệu trong một APS, và mỗi ứng viên trong bộ vừa nêu bao gồm ít nhất một trong số phép lọc chroma và phép lọc CC-ALF.

Theo phương án khác, phép lọc ALF chroma và phép lọc CC-ALF không được sử dụng đồng thời. Tức là, đối với mỗi mẫu chroma sẽ được xử lý, chỉ nhiều nhất một trong số phép lọc ALF chroma và phép lọc CC-ALF có thể được áp dụng. Theo một phương án, hình dạng lọc của hai phép lọc vừa nêu có thể được hợp nhất tiếp nữa. Theo phương án khác, một cờ được báo hiệu cho mỗi phép lọc để biểu thị phép lọc đó là phép lọc ALF chroma hay phép lọc CC-ALF chroma, sau khi một cờ hiện/ẩn được báo hiệu và cờ hiện/ẩn đó biểu thị một trong số phép lọc ALF chroma và phép lọc CC-ALF được áp dụng. Theo một phương án, một cờ được sử dụng để biểu thị phép lọc ALF chroma được sử dụng được báo hiệu trước, và nếu cờ này bằng một, thì phép lọc ALF chroma được áp dụng. Nếu cờ vừa nêu là false, thì một cờ phụ được sử dụng để biểu thị phép lọc CCALF được sử dụng hay không được báo hiệu. Nếu cờ vừa nêu là true, thì phép lọc CCALF được sử dụng. Nếu không thì, không có phép lọc nào được áp dụng. Theo phương án khác, một cờ được sử dụng để biểu thị phép lọc CC-ALF được sử dụng được báo hiệu trước, và nếu cờ này bằng 1, thì phép lọc CC-ALF được áp dụng. Nếu cờ vừa nêu là false, thì một cờ phụ được sử dụng để biểu thị phép lọc ALF chroma được sử dụng hay không được báo hiệu. Nếu cờ vừa nêu là true, thì phép lọc ALF chroma được sử dụng. Nếu không thì, không có phép lọc nào được áp dụng. Ở các phương án nêu trên, việc chọn phép lọc ALF chroma hay phép lọc CC-ALF có thể ở phân mức slice, phân mức ảnh, phân mức chuỗi ảnh, phân mức CTB, hoặc phân mức khối.

Theo phương án khác, phương pháp được đề xuất được áp dụng cho ALF luma. Tức là, tín hiệu ra ALF cho thành phần luma cũng bị ảnh hưởng bởi các mẫu chroma tương ứng. Theo một ví dụ, hai phép lọc được thêm vào, và các tín hiệu vào của một

phép lọc là thành phần Cb, và các tín hiệu vào của phép lọc còn lại là thành phần Cr. Và tín hiệu ra ALF là tổng trọng số của các tín hiệu ra của hai phép lọc phụ vừa nêu và một phép lọc gốc ban đầu. Trong trường hợp khác, chỉ một phép lọc được thêm vào nhưng các tín hiệu vào của phép lọc này là một trong số các tổ hợp bao gồm Cb, Cr, và trung bình của Cb và Cr.

Phương pháp 2: CCALF được ràng buộc - Tổng các hệ số CCALF bằng 0

Theo phương pháp này, tổng các hệ số CCALF được ràng buộc. Do sự ràng buộc này, sự dư thừa dữ liệu giữa các hệ số CCALF được tạo ra và một số thông tin không cần phải được truyền. Do đó, phương pháp này có thể nâng cao hiệu suất lập mã.

Theo CC-ALF, phép lọc tuyến tính được sử dụng để lọc các giá trị mẫu luma và khởi tạo phép hiệu chỉnh phần dư cho các kênh chroma từ tín hiệu ra được lọc cùng vị trí.

Theo một phương án của sáng chế, tổng các hệ số lọc của tất cả các phép lọc được giới hạn bởi số N nhất định. Theo một phương án, để giữ được các giá trị trung bình chroma của các thành phần Cb và Cr mà không gây ra bất kỳ sự thay đổi màu nhiều nào trong chroma, thì cần có tổng các hệ số lọc của từng phép lọc CCALF bằng 0. Do đó, các phép lọc CCALF sẽ có đáp ứng bằng 0 tại DC và sẽ xóa bỏ toàn bộ các trung bình chroma.

Theo một phương án, khi toàn bộ các hệ số lọc là đơn trị, nếu biết tổng toàn bộ các hệ số đơn trị (N), và số các hệ số lọc đơn trị (M) tại bộ giải mã, thì hệ số đơn trị cuối cùng của phép lọc có thể được xác định. Do đó, chỉ M-1 hệ số lọc đơn trị phải được giải mã. Cần được lưu ý rằng, khi hệ số lọc là đơn trị, thì các hệ số lọc này là khác nhau về trị số.

Theo một phương án, số các hệ số lọc đơn trị (M) trong phép lọc CCALF nhỏ hơn tổng các hệ số trong phép lọc. Fig. 6 minh họa ví dụ về CCALF dạng hình thoi 3x4 có 7 (M=7) các hệ số đơn trị từ c0 đến c6. Tổng các hệ số lọc là 8. Theo một phương án, nếu biết tổng toàn bộ các hệ số lọc N, chỉ 6 (M-1=6) hệ số từ c0 đến c5 phải được giải mã, và hệ số c6 có thể được xác định tại bộ giải mã như sau:

$$c6 = N - (2 * c0 + c1 + c2 + c3 + c4 + c5).$$

Theo một phương án, bất kỳ một trong số các hệ số đơn trị có thể được bỏ qua, chứ không chỉ là hệ số đơn trị cuối cùng. Theo một phương án, hệ số c2 được bỏ qua và

được xác định tại bộ giải mã. Theo phương án khác, hệ số c_5 được bỏ qua và được xác định tại bộ giải mã.

Theo một phương án, có yêu cầu về sự tương hợp luồng bit sao cho hệ số cuối cùng của các hệ số lọc Cb thành phần chéo thứ k $CcAlfApsCoeffCb[adaptation_parameter_set_id][k][7]$ sẽ nằm trong phạm vi $[-32, 31]$.

Theo một phương án, yêu cầu vừa nêu được áp dụng cho bất kỳ một hệ số trong số các hệ số lọc Cb thành phần chéo thứ k $CcAlfApsCoeffCb[adaptation_parameter_set_id][k][i]$, trong đó i nằm trong khoảng từ 0 đến tổng các hệ số trừ 1 trong phép lọc Cb thành phần chéo thứ k .

Theo một phương án, có yêu cầu về sự tương hợp luồng bit sao cho hệ số cuối cùng của các hệ số lọc Cb thành phần chéo thứ k $CcAlfApsCoeffCr[adaptation_parameter_set_id][k][7]$ sẽ nằm trong phạm vi $[-32, 31]$.

Theo một phương án, yêu cầu vừa nêu được áp dụng cho bất kỳ một hệ số trong số các hệ số lọc Cb thành phần chéo thứ k $CcAlfApsCoeffCr[adaptation_parameter_set_id][k][i]$, trong đó i nằm trong khoảng từ 0 đến tổng các hệ số trừ 1 trong phép lọc Cr thành phần chéo thứ k .

Theo một phương án, yêu cầu vừa nêu được áp dụng cho tất cả các hệ số của các hệ số lọc Cb/Cr thành phần chéo thứ k $CcAlfApsCoeffCb/Cr[adaptation_parameter_set_id][k][i]$, trong đó i nằm trong khoảng từ 0 đến tổng các hệ số trong phép lọc Cb/Cr thành phần chéo thứ k .

Phương pháp 3: CCALF được ràng buộc – Hủy bỏ CCALF khi lập mã không tồn thất và gần như không tồn thất

Theo HEVC, trong trường hợp khi CU được lập mã bằng cách sử dụng cơ chế không tồn thất và gần như không tồn thất (bao gồm cơ chế bảng màu được sử dụng cho nội dung 4:4:4), toàn bộ các phép lọc vòng lặp (tức là SAO, ALF, DF) được hủy bỏ.

Theo phiên bản hiện tại của VVC, khi CU được lập mã bằng cơ chế không tồn thất, toàn bộ các phép lọc vòng lặp được hủy bỏ bằng các thiết lập mã hóa. Đồng thời, nếu CU được lập mã bằng cách sử dụng cơ chế bảng màu, SAO, ALF và DF được hủy bỏ ở cả bộ mã hóa và bộ giải mã.

Theo một phương án của sáng chế, khi CU được lập mã bằng cách sử dụng cơ chế không tổn thất và gần như không tổn thất (bao gồm cơ chế bảng màu được sử dụng cho nội dung 4:4:4) toàn bộ các phép lọc vòng lặp phải được hủy bỏ ở cả, bộ mã hóa và bộ giải mã.

Theo một phương án, CCALF sẽ được hủy bỏ ở cả, bộ mã hóa và bộ giải mã cho các CU được lập mã bằng các cơ chế không tổn thất và gần như không tổn thất (bao gồm cơ chế bảng màu được sử dụng cho nội dung 4:4:4).

Theo một phương án, khi cây kép được sử dụng để phân vùng nút cây lập mã (treeType bằng DUAL_TREE_LUMA hoặc DUAL_TREE_CHROMA), toàn bộ các phép lọc vòng lặp có thể được hủy bỏ cho luma và chroma một cách độc lập (tức là, các độ phân giải khác nhau có thể được tạo ra cho luma và chroma).

Theo một phương án, khi cây kép được sử dụng để phân vùng nút cây lập mã, treeType bằng DUAL_TREE_CHROMA, và ít nhất một trong số các CU (CU chroma vừa nêu CU chroma hoặc CU luma tương ứng với CU chroma này) được lập mã bằng cách sử dụng cơ chế không tổn thất, thì CCALF được tự động hủy bỏ cho CU chroma.

Theo một phương án, khi cây kép không được sử dụng để phân vùng nút cây lập mã (tức là treeType bằng SINGLE_TREE), và ít nhất một trong số các CU (CU chroma vừa nêu CU chroma hoặc CU luma tương ứng với CU chroma này) được lập mã bằng cách sử dụng cơ chế không tổn thất, thì CCALF được tự động hủy bỏ cho CU chroma.

Theo một phương án, khi cây kép được sử dụng để phân vùng nút cây lập mã, treeType bằng DUAL_TREE_CHROMA, và CU chroma được lập mã bằng cách sử dụng cơ chế không tổn thất, thì CCALF được tự động hủy bỏ cho CU chroma vừa nêu.

Theo một phương án, khi cây kép không được sử dụng để phân vùng nút cây lập mã, (treeType bằng SINGLE_TREE), và CU chroma được lập mã bằng cách sử dụng cơ chế không tổn thất, thì CCALF được tự động hủy bỏ cho CU chroma vừa nêu.

Theo một phương án, khi cây kép được sử dụng để phân vùng nút cây lập mã (treeType bằng DUAL_TREE_CHROMA), và cả, CU chroma và CU luma tương ứng được lập mã bằng cách sử dụng cơ chế không tổn thất, thì CCALF được tự động hủy bỏ cho CU chroma.

Theo một phương án, khi cây kép không được sử dụng để phân vùng nút cây lập

mã (treeType bằng SINGLE_TREE), và cả, CU chroma và CU luma tương ứng được lập mã bằng cách sử dụng cơ chế không tồn thất, thì CCALF được tự động hủy bỏ cho CU chroma.

Theo một phương án, nếu các khối dư cho cả thành phần Cb và Cr chroma được lập mã thành một khối biến đổi đơn nhất (ví dụ $\text{tu_joint_cbr_residual_flag}[xC][yC] == 1$), và ít nhất một trong số các CU (Cb hoặc Cr) được lập mã bằng cách sử dụng cơ chế không tồn thất, thì CCALF được tự động hủy bỏ cho cả các CU chroma.

Theo một phương án, nếu các khối dư cho cả thành phần Cb và Cr chroma được lập mã thành một khối biến đổi đơn nhất (ví dụ $\text{tu_joint_cbr_residual_flag}[xC][yC] == 1$), và ít nhất một trong số các CU (Cb hoặc Cr) được lập mã bằng cách sử dụng cơ chế không tồn thất, thì CCALF được tự động hủy bỏ cho CU chroma tương ứng.

Phương pháp 4: Báo hiệu số lượng phép lọc trong CCALF

Để biết số lượng các phép lọc CCALF, thành phần cú pháp phụ được yêu cầu để được báo hiệu. Theo phương pháp truyền thống, thành phần cú pháp phụ vừa nêu được báo hiệu ở tiêu đề slice SH (Slice Head - SH). Theo phương pháp truyền thống khác được bộc lộ trong JVET-P1008 (K. Misra, và các cộng sự, “CE5: Cấu trúc của CC-ALF”, Nhóm chuyên gia lập mã video chung JVET(Joint Video Experts Team - JVET) của ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, hội nghị lần thứ 16: Geneva, CH, ngày 1 đến 11 tháng 10 năm 2019, JVET-P1008-v2), tài liệu này bộc lộ cấu trúc cú pháp để báo hiệu số lượng phép lọc trong CCALF, được biến đổi dựa vào VVC phiên bản 6 (J. Chen, và các cộng sự, “Phép lọc vòng lặp thích ứng thành phần chéo cho chroma”, Nhóm chuyên gia lập mã video chung JVET(Joint Video Experts Team - JVET) của ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, hội nghị lần thứ 15: Gothenburg, SE, ngày 3 đến 12 tháng 7 năm 2019, tài liệu: JVET-O2001-vE). Theo JVET-P1008, các thành phần cú pháp $\text{alf_ctb_cross_component_cb_idc}[]$ và $\text{alf_ctb_cross_component_cr_idc}[]$ được báo hiệu ở phân mức CTU. Thành phần cú pháp $\text{alf_ctb_cross_component_cb_idc}[]$ biểu thị liệu phép lọc Cb thành phần chéo được áp dụng cho khối các mẫu thành phần màu Cb hay không. Thành phần cú pháp $\text{alf_ctb_cross_component_cr_idc}[]$ biểu thị liệu phép lọc Cr thành phần chéo được áp

dụng cho khối các mẫu thành phần màu Cr hay không. Tất cả các thành phần cú pháp được nhị phân hóa bằng cách sử dụng lập mã Rice cắt xén TR (Truncated Rice - TR) như được thể hiện ở bảng dưới đây:

Bảng 2

Cấu trúc cú pháp	Thành phần cú pháp	Nhị phân hóa	
		Phép xử lý	Các tham số vào
coding_tree_unit ()	alf_ctb_cross_component_ cb_idc[][]	TR	cMax = (slice_cross_component_ cb_filters_signalled_minus1 + 1), cRiceParam = 0
	alf_ctb_cross_component_ cr_idc[][]	TR	cMax = (slice_cross_component_ t_ cr_filters_signalled_minus1 + 1), cRiceParam = 0

Ở Bảng 2, cMax và cRiceParam là hai tín hiệu vào được cấp cho quá trình nhị phân của alf_ctb_cross_component_cb_idc[][] và alf_ctb_cross_component_cr_idc[][] bằng cách sử dụng lập mã Rice cắt xén TR. Chi tiết về quá trình nhị phân vừa nêu được mô tả trong JVET-O2001-vE. Các thành phần cú pháp slice_cross_component_cb_filters_signalled_minus1 và slice_cross_component_cr_filters_signalled_minus1 được báo hiệu ở tiêu đề slice (SH).

Trong JVET-P1008, một số thành phần cú pháp được giới thiệu, mà được mô tả vắn tắt như sau. Slice_cross_component_alf_cb_enabled_flag xác định liệu phép lọc Cb thành phần chéo được áp dụng cho thành phần màu Cb hay không. Khi

`slice_cross_component_alf_cb_enabled_flag` không xuất hiện, nó được suy diễn bằng 0.

Khi `slice_cross_component_alf_cb_enabled_flag` bằng 1, có yêu cầu về sự tương hợp luồng بیت sao cho, đối với toàn bộ các slice của hình ảnh hiện thời, ALF APS được tính bởi `slice_cross_component_alf_cb_aps_id` sẽ bằng nhau. `Slice_cross_component_cb_filters_signalled_minus1` cộng 1 xác định số lượng các phép lọc Cb thành phần chéo. Trị số của `slice_cross_component_cb_filters_signalled_minus1` sẽ nằm trong phạm vi từ 0 đến 3.

Khi `slice_cross_component_alf_cb_enabled_flag` bằng 1, có yêu cầu về sự tương hợp luồng بیت sao cho `slice_cross_component_cb_filters_signalled_minus1` sẽ nhỏ hơn hoặc bằng trị số của `alf_cross_component_cb_filters_signalled_minus1` ở ALF APS được tính bởi `slice_cross_component_alf_cb_aps_id` của slice hiện thời.

`Slice_cross_component_alf_cr_enabled_flag` xác định liệu phép lọc Cr thành phần chéo được áp dụng cho thành phần màu Cr hay không. Khi `slice_cross_component_alf_cr_enabled_flag` không xuất hiện, nó được suy diễn bằng 0.

`Slice_cross_component_alf_cr_aps_id` mô tả giống như `slice_cross_component_alf_cb_aps_id`, nhưng là cho thành phần màu Cr của slice tham chiếu đến. Khi `slice_cross_component_alf_cr_enabled_flag` bằng 1, có yêu cầu về sự tương hợp luồng بیت sao cho, đối với toàn bộ các slice của hình ảnh hiện thời, ALF APS được tính bởi `slice_cross_component_alf_cr_aps_id` sẽ bằng nhau.

`Slice_cross_component_cr_filters_signalled_minus1` cộng 1 xác định số lượng các phép lọc Cr thành phần chéo. Trị số của `slice_cross_component_cr_filters_signalled_minus1` sẽ nằm trong phạm vi từ 0 đến 3.

Khi `slice_cross_component_alf_cr_enabled_flag` bằng 1, có yêu cầu về sự tương hợp luồng بیت sao cho `slice_cross_component_cr_filters_signalled_minus1` sẽ nhỏ hơn hoặc bằng trị số của `alf_cross_component_cr_filters_signalled_minus1` ở ALF APS được tính bởi `slice_cross_component_alf_cr_aps_id` của slice hiện thời.

Theo một phương án của sáng chế,
`pic_cross_component_cb_filters_signalled_minus1`,
`pic_cross_component_cr_filters_signalled_minus1`,
`slice_cross_component_cb_filters_signalled_minus1` và

`slice_cross_component_cr_filters_signalled_minus1` không xuất hiện trong cú pháp CC-ALF và `alf_ctb_cross_component_cb_idc[ ][ ]` và `alf_ctb_cross_component_cr_idc[ ][ ]` được lập mã ở Bảng 3 như sau:

Bảng 3

<code>alf_ctb_cross_component_cb_idc[][]</code>	TR	$cMax = (alf_cross_component_cb_filter_s_signalled_minus1 + 1)$, $cRiceParam = 0$
<code>alf_ctb_cross_component_cr_idc[][]</code>	TR	$cMax = (alf_cross_component_cr_filter_s_signalled_minus1 + 1)$, $cRiceParam = 0$

Theo cấu trúc cú pháp đã đề xuất, các cú pháp, `slice_cross_component_cb_filters_signalled_minus1` và `slice_cross_component_cr_filters_signalled_minus1` của Bảng 2 được thay thế bằng `alf_cross_component_cb_filters_signalled_minus1` và `alf_cross_component_cr_filters_signalled_minus1` một cách tương ứng ở Bảng 3. Nói cách khác, thay vì sử dụng các thành phần cú pháp được báo hiệu ở tiêu đề slice, số lượng các phép lọc được báo hiệu ở ALF APS và được tham chiếu trực tiếp để xác định số lượng các phép lọc được lập mã cho CC-ALF. Do đó, khác với phép báo hiệu và cú pháp thông thường, cấu trúc CC-ALF được tương thích với cấu trúc ALF. Tương tự, vì các slice khác nhau nằm trong hình ảnh có thể tham chiếu đến các ALF APS khác nhau, nên nếu số lượng các phép lọc CC-ALF xuất hiện trong ALF APS, thì 4 bit trên một slice có thể được tiết kiệm (2 bit trên một thành phần màu).

Bất kỳ phương pháp nào trong số các phương pháp đã đề xuất ở trên có thể được thực hiện ở các bộ mã hóa và/hoặc các bộ giải mã. Ví dụ, bất kỳ phương pháp nào trong số các phương pháp đã đề xuất đó có thể được thực hiện ở mô đun lọc vòng lặp kín của bộ mã hóa và/hoặc bộ giải mã. Ngoài ra, bất kỳ phương pháp nào trong số các phương pháp đã đề xuất đó có thể được thực hiện dưới dạng vi mạch được kết nối với mô đun lọc vòng lặp kín của bộ mã hóa và/hoặc bộ giải mã.

Fig. 7 minh họa lưu đồ của phép xử lý ALF của video tái cấu trúc được lấy làm ví dụ theo phương án của sáng chế, trong đó tổng các hệ số lọc cho mỗi một trong số một hoặc nhiều phép lọc ứng viên được ràng buộc là một trị số cố định. Các bước được thể hiện trên lưu đồ có thể được thực hiện dưới dạng các mã chương trình có thể xử lý bằng một hoặc nhiều bộ xử lý (ví dụ, một hoặc nhiều CPU tại bộ mã hóa). Các bước được thể hiện trên lưu đồ có thể còn được thực hiện dưới dạng phần cứng chẳng hạn một hoặc nhiều thiết bị điện tử hoặc bộ xử lý được bố trí để thực hiện các bước trên lưu đồ. Theo phương pháp này, các mẫu tái cấu trúc được nhận ở bước 710. Phép lọc vòng lặp thích ứng thành phần màu chéo CCALF mục tiêu (Cross-Colour Adaptive Loop Filter - CCALF) được xác định ở bước 720, trong đó CCALF mục tiêu thuộc bộ phép lọc bao gồm một hoặc nhiều phép lọc ứng viên và tổng các hệ số lọc cho từng phép lọc ứng viên, ở một hoặc nhiều phép lọc ứng viên vừa nêu, được ràng buộc là một trị số cố định. CCALF mục tiêu được áp dụng cho các mẫu luma tái cấu trúc liên đới để tạo ra mẫu chroma được lọc CCALF cho mẫu chroma tái cấu trúc mục tiêu ở bước 730. Mẫu chroma được lọc cuối cùng được tạo ra bằng cách tổ hợp mẫu chroma được lọc CCALF với mẫu chroma được lọc ALF thông thường ở bước 740, trong đó mẫu chroma được lọc cuối cùng được sử dụng làm đầu ra giải mã video hoặc được sử dụng cho quá trình mã hóa hoặc giải mã video tiếp theo.

Fig. 8 minh họa lưu đồ của phép xử lý ALF của video tái cấu trúc được lấy làm ví dụ theo phương án của sáng chế, trong đó một hoặc nhiều thành phần cú pháp biểu thị trị số liên quan đến tổng số các phép lọc ứng viên trong bộ phép lọc được báo hiệu hoặc được phân tích trong APS (Bộ tham số thích ứng). Theo phương pháp này, các mẫu tái cấu trúc được nhận ở bước 810. Một hoặc nhiều thành phần cú pháp được báo hiệu trong APS (Bộ tham số thích ứng) của luồng bit video tại bộ mã hóa hoặc một hoặc nhiều thành phần cú pháp vừa nêu được phân tích trong APS của luồng bit video tại bộ giải mã ở bước 820, trong đó một hoặc nhiều thành phần cú pháp vừa nêu biểu thị trị số liên quan đến tổng số các phép lọc ứng viên trong bộ phép lọc. Phép lọc vòng lặp thích ứng thành phần màu chéo CCALF mục tiêu (Cross-Colour Adaptive Loop Filter - CCALF) được xác định ở bước 830, trong đó CCALF mục tiêu thuộc bộ phép lọc bao gồm một hoặc nhiều phép lọc ứng viên và tổng các hệ số lọc cho mỗi một trong số nhiều phép lọc ứng viên được ràng buộc là một trị số cố định. CCALF mục tiêu được áp dụng cho các mẫu luma tái cấu trúc liên đới để tạo ra mẫu chroma được lọc CCALF cho mẫu

chroma tái cấu trúc mục tiêu ở bước 840. Mẫu chroma được lọc cuối cùng được tạo ra bằng cách tổ hợp mẫu chroma được lọc CCALF với mẫu chroma được lọc ALF thông thường ở bước 850, trong đó mẫu chroma được lọc cuối cùng được sử dụng làm đầu ra giải mã video hoặc được sử dụng cho quá trình mã hóa hoặc giải mã video tiếp theo.

Các lưu đồ được thể hiện nhằm mục đích minh họa ví dụ về lập mã video theo sáng chế. Người có chuyên môn trong lĩnh vực kỹ thuật này có thể sửa đổi từng bước, sắp xếp lại các bước, phân chia bước, hoặc kết hợp các bước để thực hiện sáng chế mà không lệch khỏi nội dung của sáng chế. Ở bản mô tả này, cú pháp và các ngôn ngữ cụ thể được sử dụng để minh họa các ví dụ thực hiện các phương án của sáng chế. Người có chuyên môn có thể thực hiện sáng chế bằng cách thay thế cú pháp và các ngôn ngữ này bằng cú pháp và các ngôn ngữ tương đương khác mà không lệch khỏi nội dung của sáng chế.

Phần mô tả đã nêu được trình bày nhằm giúp người có hiểu biết trung bình trong lĩnh vực này có thể thực hiện sáng chế như được nêu theo nội dung của ứng dụng nhất định và yêu cầu của nó. Nhiều sửa đổi khác nhau đối với các phương án được mô tả sẽ là điều dễ hiểu đối với những người có hiểu biết trung bình trong lĩnh vực kỹ thuật này, và những nguyên lý chung được định nghĩa ở bản mô tả này có thể được áp dụng cho các phương án khác. Vì vậy, sáng chế không dự định sẽ được giới hạn chỉ ở các phương án cụ thể đã được thể hiện và được mô tả, mà sẽ được tuân theo phạm vi rộng nhất phù hợp với những nguyên lý và những dấu hiệu mới được bộc lộ ở bản mô tả này. Ở phần mô tả chi tiết sáng chế nêu trên, nhiều phần chi tiết cụ thể khác nhau được minh họa để giúp hiểu toàn diện sáng chế. Tuy nhiên, sẽ được hiểu bởi những người có hiểu biết trung bình trong lĩnh vực kỹ thuật này đó là sáng chế có thể được thực hiện.

Phương án của sáng chế như được mô tả ở trên có thể được thực hiện theo nhiều mã phần cứng, phần mềm khác nhau, hoặc kết hợp cả mã phần cứng và mã phần mềm. Ví dụ, phương án của sáng chế có thể là một hoặc mạch được tích hợp thành chip nén video hoặc mã chương trình được tích hợp thành phần mềm nén video để thực hiện quá trình xử lý được mô tả ở đây. Phương án của sáng chế có thể còn là mã chương trình được xử lý trên bộ xử lý tín hiệu số (DSP) để thực hiện quá trình xử lý được mô tả ở đây. Sáng chế có thể còn sử dụng số lượng các hàm được thực hiện bởi bộ xử lý máy tính, bộ xử lý tín hiệu số, bộ vi xử lý, hoặc vi mạch tích hợp FPGA (Field Programmable Gate

Array - FPGA). Những bộ xử lý này có thể được cấu hình để thực hiện những nhiệm vụ cụ thể theo sáng chế, bằng cách xử lý đoạn mã phần mềm hoặc đoạn mã vi chương trình có thể đọc được bằng máy mà định ra các phương pháp cụ thể được thực hiện bởi sáng chế. Mã phần mềm hoặc mã vi chương trình có thể được phát triển theo các ngôn ngữ lập trình khác nhau và các định dạng hoặc kiểu khác nhau. Mã phần mềm có thể cũng được tương thích theo các nền tảng mục tiêu khác nhau. Tuy nhiên, các định dạng, kiểu, ngôn ngữ lập trình khác nhau của các mã phần mềm và các phương tiện cấu hình nên lập trình khác để thực hiện những nhiệm vụ theo sáng chế sẽ không trệch khỏi nội dung và phạm vi của sáng chế.

Sáng chế có thể được thực hiện theo các dạng cụ thể khác mà không lệch khỏi nội dung hoặc các dấu hiệu cơ bản của sáng chế. Các ví dụ được mô tả được xét theo toàn bộ các khía cạnh chỉ nhằm mục đích minh họa chứ không nhằm mục đích giới hạn. Vì vậy, phạm vi của sáng chế được thể hiện bằng các điểm yêu cầu bảo hộ đính kèm chứ không phải phần mô tả ở trên. Mọi thay đổi mà thuộc nội dung và phạm vi tương đương của các điểm yêu cầu bảo hộ đều được xem là thuộc phạm vi bảo hộ của sáng chế.

YÊU CẦU BẢO HỘ

1. Phương pháp lập mã video, phương pháp này bao gồm:

nhận các mẫu chroma được tái cấu trúc và các mẫu luma tái cấu trúc liên đới trong hình ảnh màu;

xác định phép lọc vòng lặp thích ứng thành phần màu chéo CCALF (Cross-Colour Adaptive Loop Filter) mục tiêu, trong đó CCALF mục tiêu thuộc bộ phép lọc bao gồm một hoặc nhiều phép lọc ứng viên và tổng các hệ số lọc cho từng phép lọc ứng viên, ở một hoặc nhiều phép lọc ứng viên vừa nêu, được ràng buộc là một trị số cố định;

áp dụng CCALF mục tiêu cho các mẫu luma tái cấu trúc liên đới để tạo ra mẫu chroma được lọc CCALF cho mẫu chroma tái cấu trúc mục tiêu; và

tạo ra mẫu chroma được lọc cuối cùng bằng cách tổ hợp mẫu chroma được lọc CCALF và mẫu chroma được lọc ALF thông thường, trong đó mẫu chroma được lọc cuối cùng được sử dụng làm đầu ra giải mã video hoặc được sử dụng cho quá trình mã hóa hoặc giải mã video tiếp theo.

2. Phương pháp theo điểm 1, trong đó trị số cố định bằng 0.

3. Phương pháp theo điểm 1, trong đó khi số lượng các hệ số lọc cho từng phép lọc ứng viên bằng M, chỉ (M-1) hệ số lọc cho từng phép lọc ứng viên được giải mã tại bộ giải mã, và trong đó M là số nguyên lớn hơn 1.

4. Phương pháp theo điểm 3, trong đó một hệ số lọc, chứ không phải (M-1) hệ số lọc, không được giải mã tại bộ giải mã hoặc không được mã hóa tại bộ mã hóa, và trong đó một hệ số lọc vừa nêu tương ứng với bất kỳ hệ số được chọn trong số các hệ số lọc cho từng phép lọc ứng viên.

5. Thiết bị lập mã video, thiết bị này bao gồm một hoặc nhiều mạch điện tử hoặc bộ xử lý được dùng để:

nhận các mẫu chroma được tái cấu trúc và các mẫu luma tái cấu trúc liên đới trong hình ảnh màu;

xác định phép lọc vòng lặp thích ứng thành phần màu chéo CCALF (Cross-Colour Adaptive Loop Filter) mục tiêu, trong đó CCALF mục tiêu thuộc bộ phép lọc bao gồm một hoặc nhiều phép lọc ứng viên và tổng các hệ số lọc cho mỗi một trong số nhiều phép

lọc ứng viên được ràng buộc là một trị số cố định;

áp dụng CCALF mục tiêu cho các mẫu luma tái cấu trúc liên đới để tạo ra mẫu chroma được lọc CCALF cho mẫu chroma tái cấu trúc mục tiêu; và

tạo ra mẫu chroma được lọc cuối cùng bằng cách tổ hợp mẫu chroma được lọc CCALF và mẫu chroma được lọc ALF thông thường, trong đó mẫu chroma được lọc cuối cùng được sử dụng làm đầu ra giải mã video hoặc được sử dụng cho quá trình mã hóa hoặc giải mã video tiếp theo.

6. Phương pháp lập mã video, phương pháp này bao gồm:

nhận các mẫu chroma được tái cấu trúc và các mẫu luma tái cấu trúc liên đới trong hình ảnh màu;

báo hiệu một hoặc nhiều thành phần cú pháp trong bộ tham số thích ứng APS (Adaptation Parameter Set) của luồng bit video tại bộ mã hóa hoặc phân tích một hoặc nhiều thành phần cú pháp vừa nêu trong APS của luồng bit video tại bộ giải mã, trong đó một hoặc nhiều thành phần cú pháp vừa nêu biểu thị trị số liên quan đến tổng các phép lọc ứng viên trong bộ phép lọc;

xác định phép lọc vòng lặp thích ứng thành phần màu chéo CCALF (Cross-Colour Adaptive Loop Filter) mục tiêu, trong đó CCALF mục tiêu thuộc bộ phép lọc;

áp dụng CCALF mục tiêu cho các mẫu luma tái cấu trúc liên đới để tạo ra mẫu chroma được lọc CCALF cho mẫu chroma tái cấu trúc mục tiêu; và

tạo ra mẫu chroma được lọc cuối cùng bằng cách tổ hợp mẫu chroma được lọc CCALF và mẫu chroma được lọc ALF thông thường, trong đó mẫu chroma được lọc cuối cùng được sử dụng làm đầu ra giải mã video hoặc được sử dụng cho quá trình mã hóa hoặc giải mã video tiếp theo.

7. Phương pháp theo điểm 6, trong đó một hoặc nhiều thành phần cú pháp vừa nêu được báo hiệu hoặc được phân tích cú pháp cho từng thành phần màu.

8. Phương pháp theo điểm 6, trong đó một hoặc nhiều thành phần cú pháp vừa nêu bao gồm trị số tương ứng với tổng số các phép lọc ứng viên trong bộ phép lọc trừ 1.

9. Thiết bị lập mã video, thiết bị này bao gồm một hoặc nhiều mạch điện tử hoặc bộ xử lý được dùng để:

nhận các mẫu chroma được tái cấu trúc và các mẫu luma tái cấu trúc liên đới trong hình ảnh màu;

báo hiệu một hoặc nhiều thành phần cú pháp trong bộ tham số thích ứng APS (Adaptation Parameter Set) của luồng bit video tại bộ mã hóa hoặc phân tích một hoặc nhiều thành phần cú pháp vừa nêu trong APS của luồng bit video tại bộ giải mã, trong đó một hoặc nhiều thành phần cú pháp vừa nêu biểu thị trị số liên quan đến tổng các phép lọc ứng viên trong bộ phép lọc;

xác định phép lọc vòng lặp thích ứng thành phần màu chéo CCALF (Cross-Colour Adaptive Loop Filter) mục tiêu, trong đó CCALF mục tiêu thuộc bộ phép lọc;

áp dụng CCALF mục tiêu cho các mẫu luma tái cấu trúc liên đới để tạo ra mẫu chroma được lọc CCALF cho mẫu chroma tái cấu trúc mục tiêu; và

tạo ra mẫu chroma được lọc cuối cùng bằng cách tổ hợp mẫu chroma được lọc CCALF và mẫu chroma được lọc ALF thông thường, trong đó mẫu chroma được lọc cuối cùng được sử dụng làm đầu ra giải mã video hoặc được sử dụng cho quá trình mã hóa hoặc giải mã video tiếp theo.

1/7

Fig. 1A

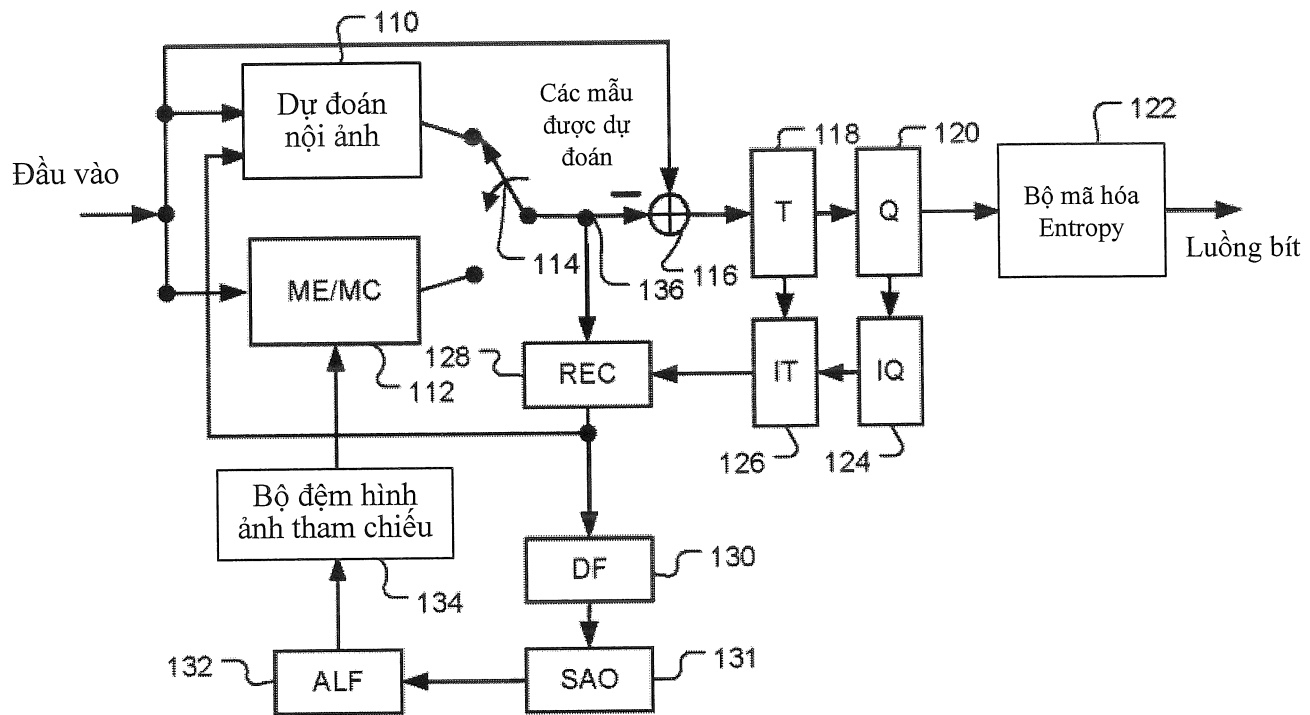
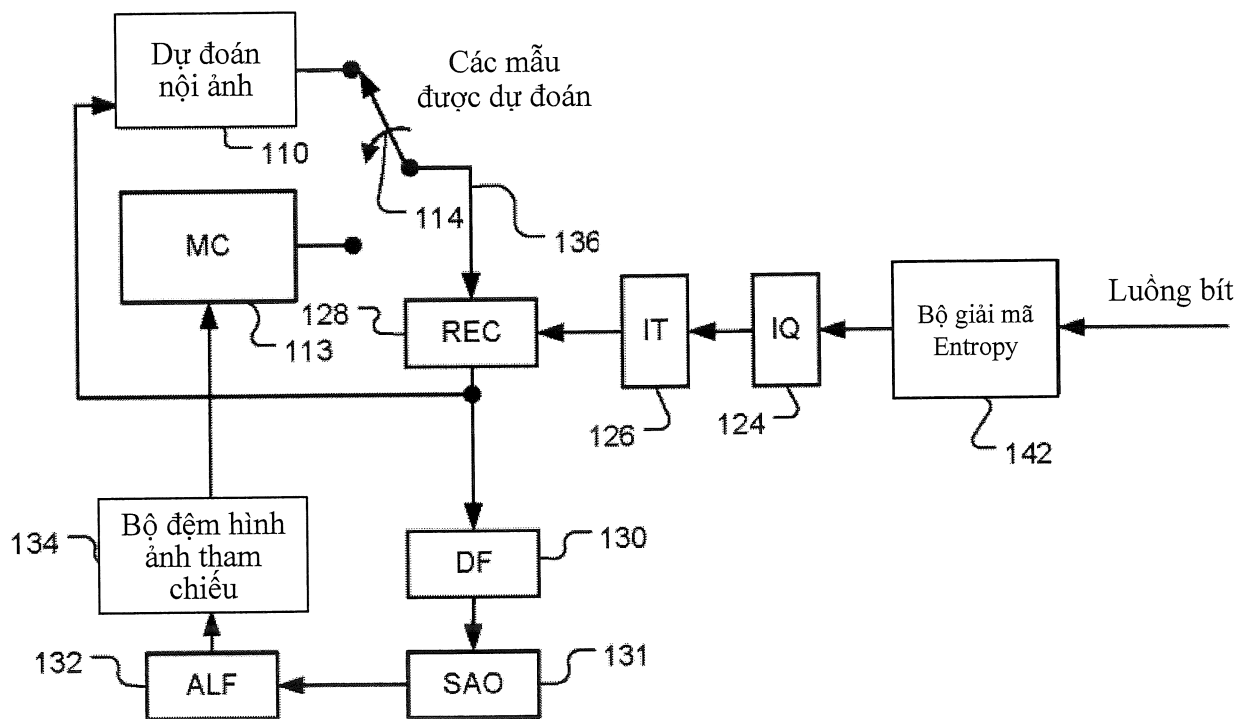


Fig. 1B



2/7

Fig. 2A

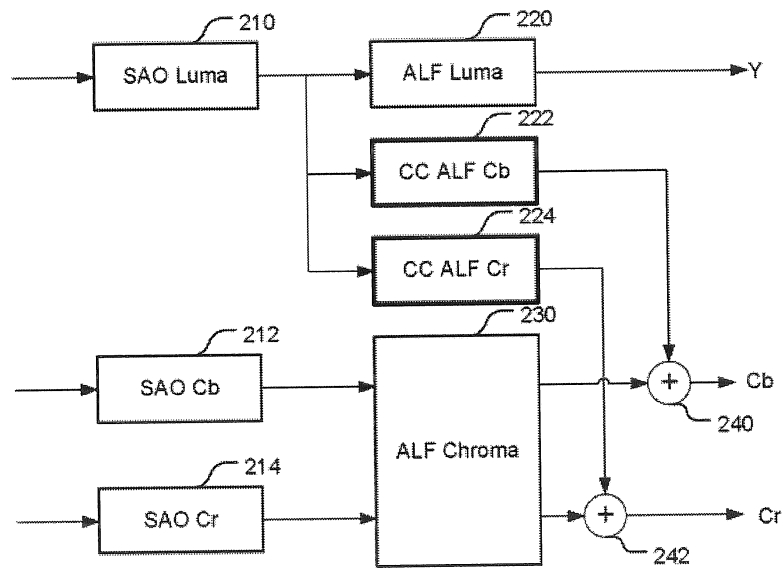
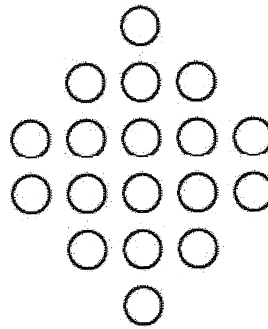


Fig. 2B



3/7

Fig. 2C

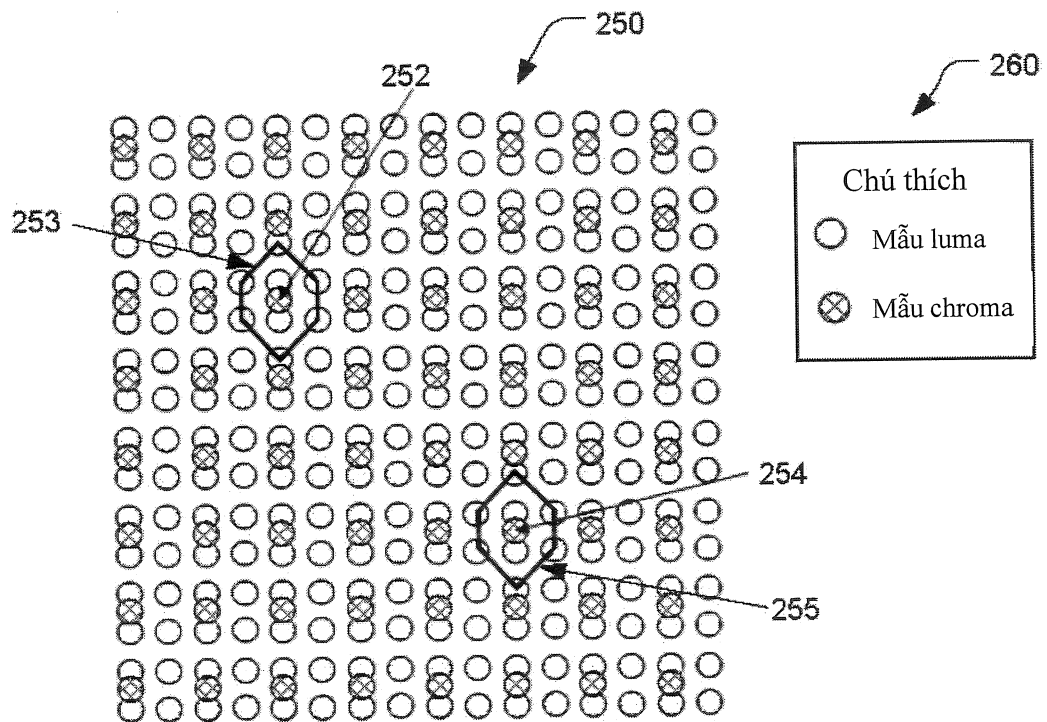
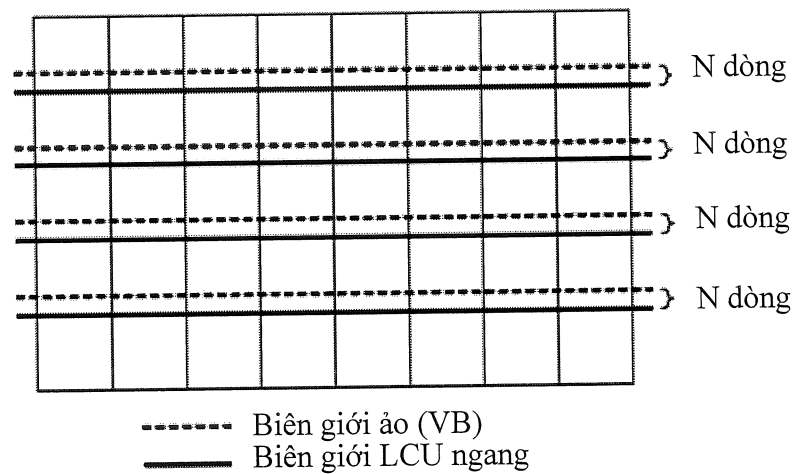


Fig. 3



4/7

Fig. 4A

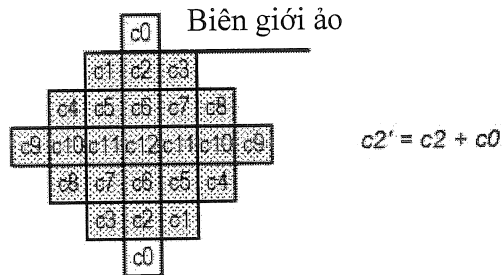


Fig. 4B

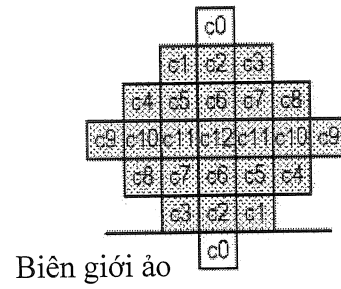


Fig. 4C

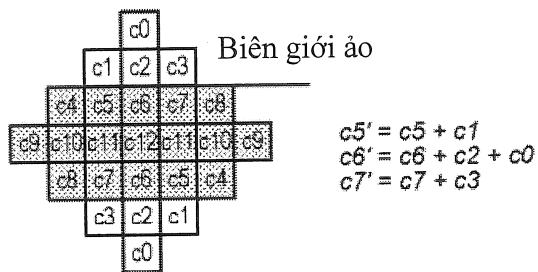


Fig. 4D

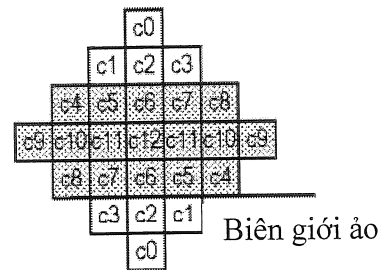


Fig. 4E

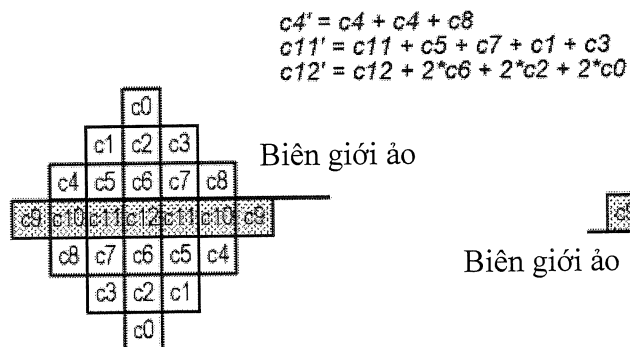
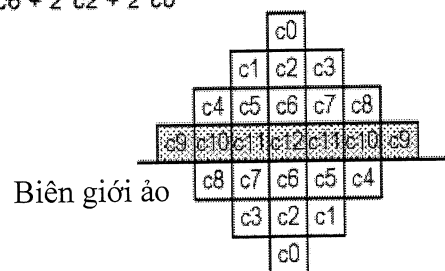


Fig. 4F



5/7

Fig. 5A

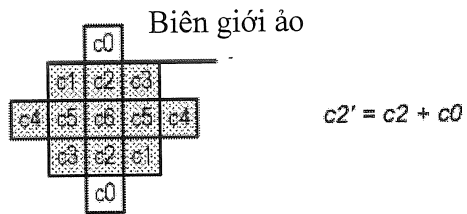


Fig. 5B

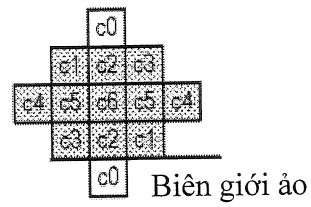


Fig. 5C

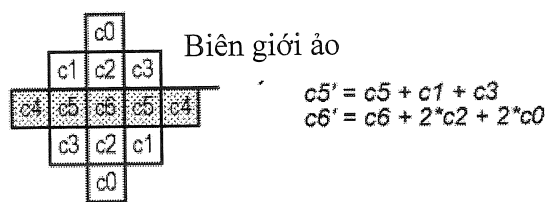


Fig. 5D

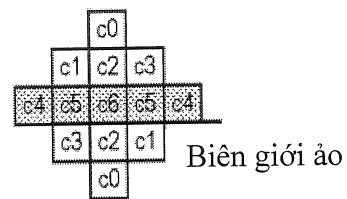
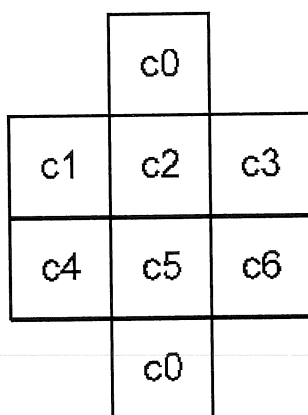
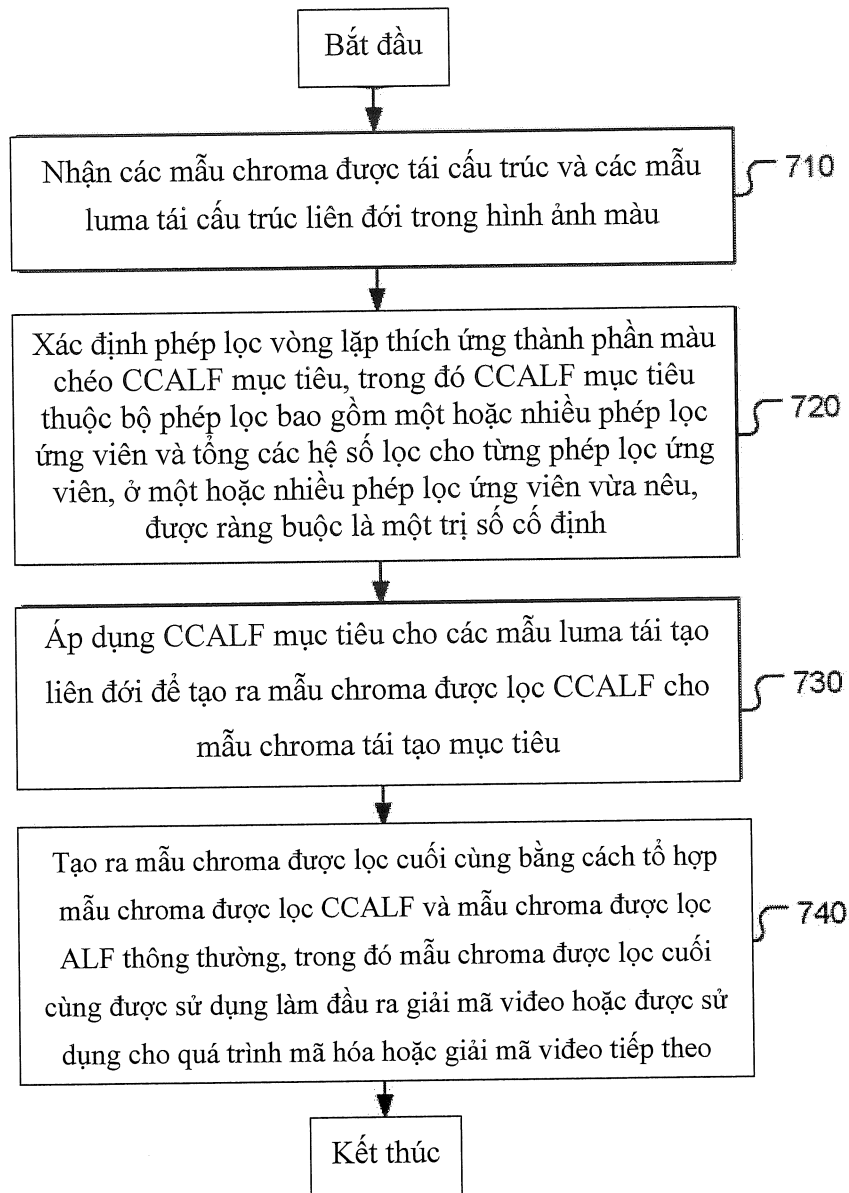


Fig. 6



6/7

Fig. 7



7/7

Fig. 8

