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

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

(11) **1-0049799**

(51)[2021.01] **H04N 19/12**

(13) **B**

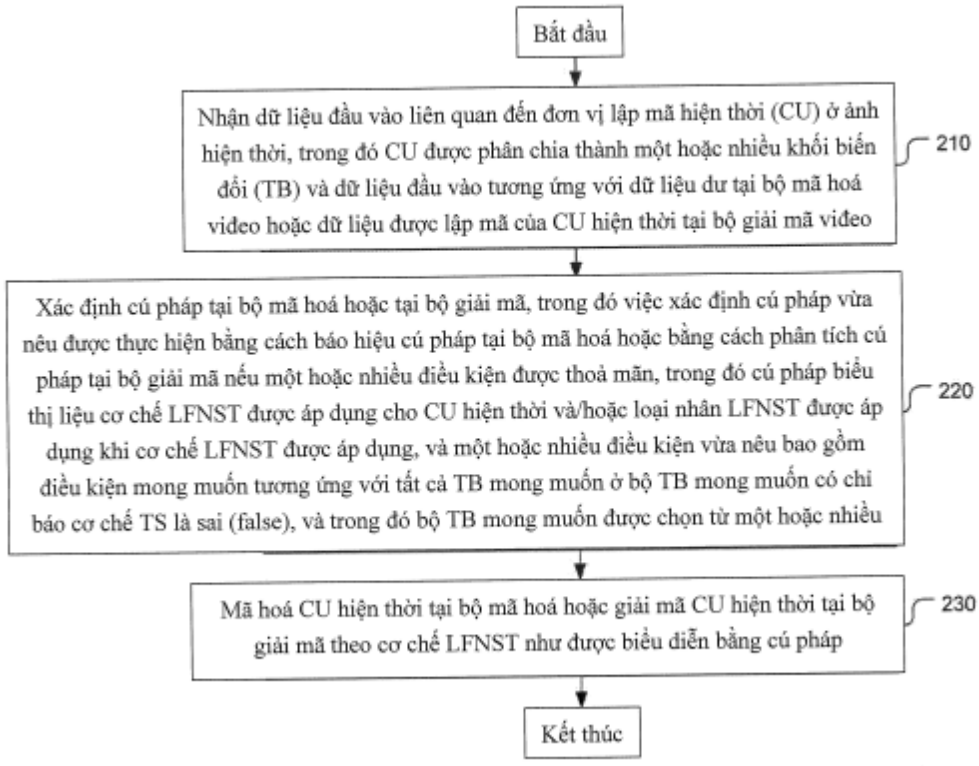
(21) 1-2022-04164
(22) 10/12/2020
(86) PCT/CN2020/135303 10/12/2020
(87) WO2021/115387 17/06/2021
(30) 62/947,057 12/12/2019 US; 62/949,505 18/12/2019 US; 62/959,938 11/01/2020 US
(45) 25/08/2025 449
(43) 26/09/2022 414A
(73) HFI INNOVATION INC. (CN)
3F.-7, No.5, Taiyuan 1st St., Zhubei City, Hsinchu County 302, Taiwan
(72) CHIANG, Man-Shu (TW); CHUANG, Tzu-Der (TW); HSU, Chih-Wei (TW); CHEN, Ching-Yeh (TW); LIN, Zhi-Yi (TW).
(74) Công ty TNHH Trường Xuân (AGELESS CO.,LTD.)

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ MÃ HOÁ HOẶC GIẢI MÃ CHUỖI VIĐEO

(21) 1-2022-04164

(57) Sáng chế đề cập đến phương pháp mã hoá hoặc giải mã chuỗi viđeo bằng cách sử dụng cơ chế biến đổi không phân tách tần số thấp (low-frequency non-separable transform: LFNST) và thiết bị mã hóa hoặc giải mã chuỗi viđeo. Đơn vị lập mã (coding unit: CU) được phân chia thành một hoặc nhiều khối biến đổi (transform block: TB). Cú pháp được xác định tại bộ mã hoá hoặc tại bộ giải mã, trong đó bước xác định được thực hiện bằng cách báo hiệu cú pháp tại bộ mã hoá hoặc bằng cách phân tích cú pháp tại bộ giải mã nếu một hoặc nhiều điều kiện được thoả mãn. Cú pháp biểu thị liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc loại nhân LFNST được áp dụng khi LFNST được áp dụng, và các điều kiện vừa nêu bao gồm điều kiện mong muốn tương ứng với tất cả TB mong muốn ở bộ TB mong muốn có chỉ báo cơ chế TS là sai (false), và bộ TB mong muốn được chọn từ các TB ở CU hiện thời. CU hiện thời được mã hoá tại bộ mã hoá hoặc được giải mã tại bộ giải mã theo cơ chế LFNST như được biểu diễn bằng cú pháp.

Fig. 2



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến lập mã viđeo. Cụ thể hơn là, sáng chế đề cập đến phương pháp và thiết bị mã hoá hoặc giải mã chuỗi viđeo.

Tình trạng kỹ thuật của sáng chế

Chuẩn lập mã viđeo hiệu suất cao HEVC (High-Efficiency Video Coding - HEVC) là một chuẩn lập viđeo quốc tế mới được phát triển bởi nhóm hợp tác chung về lập mã viđeo JCT-VC (Joint Collaborative Team on Video Coding - JCT-VC). HEVC dựa trên cấu trúc lập mã biến đổi tương tự với phép biến đổi DCT bù chuyển động dựa vào khối lai. Đơn vị cơ bản để nén, về mặt thuật ngữ được gọi là đơn vị lập mã CU (Coding Unit - CU), là khối vuông có kích thước 2Nx2N, và mỗi CU có thể được chia đệ quy thành bốn CU nhỏ hơn cho đến khi kích cỡ nhỏ nhất định trước đạt được. Mỗi CU bao gồm một hoặc nhiều đơn vị dự đoán PU (Prediction Unit - PU).

Để đạt được hiệu suất lập mã tốt nhất cho kiến trúc lập mã lai trong HEVC, có hai dạng cơ chế dự đoán cho mỗi PU, đó là dự đoán nội ảnh và dự đoán liên ảnh. Đối với các cơ chế dự đoán nội ảnh, các điểm ảnh được tái tạo lân cận theo không gian có thể được sử dụng để tạo ra các dự đoán hướng. Có tới 35 hướng trong HEVC. Đối với các cơ chế dự đoán liên ảnh, các khung hình tham chiếu được khôi phục theo thời gian có thể được sử dụng để tạo ra các dự đoán được bù chuyển động. Có ba loại cơ chế dự đoán liên ảnh, bao gồm cơ chế dự đoán véctơ tiên tiến AMVP (Advanced Motion vector Prediction - AMVP) bỏ qua (Skip), trộn (Merge) và liên ảnh (Inter).

Phép biến đổi

Sau khi dự đoán, các phần dư dự đoán cho một CU được chia thành các đơn vị biến đổi TU (Transform Unit - TU) và được lập mã bằng phép biến đổi và lượng tử hóa. Giống như nhiều tiêu chuẩn lập mã tiền lệ khác, HEVC sử dụng phép biến đổi côsin rời rạc dạng II (DCT-II) làm phép biến đổi nhân (biến đổi chính) bởi vì nó có đặc tính "nén năng lượng mạnh". Để nâng cao khả năng biến đổi, phép biến đổi sin rời rạc DST (Discrete Sine Transform - DST) cũng được đề xuất để được sử dụng thay phép DCT đối với những phương vị nội ảnh theo phương chéo. DCT-II là phép biến đổi duy nhất được sử dụng ở phiên bản HEVC hiện tại. Tuy nhiên, DCT-II không phải là phép biến

đổi tối ưu cho mọi trường hợp. Phép biến đổi sin rời rạc dạng VII (DST-VII) và phép biến đổi côsin rời rạc dạng VIII (DCT-VIII) được đề xuất để thay cho DCT-II trong một số trường hợp. Mô hình chọn phép biến đổi (MTS) cũng được sử dụng để lập mã phần dư cho các khối được lập mã nội ảnh và liên ảnh. Nó sử dụng nhiều phép biến đổi được chọn từ họ các phép DCT/DST khác với các phép biến đổi hiện tại ở HEVC. Các ma trận biến đổi mới nhất được giới thiệu là DCT-VIII. Theo VVC, mô hình chọn phép biến đổi (MTS) cho biến đổi nhân được mô tả như sau.

Ngoài DCT-II đã được sử dụng trong HEVC, mô hình chọn phép biến đổi (MTS) được sử dụng để lập mã phần dư cho các khối được lập mã nội ảnh và/hoặc liên ảnh. Nó sử dụng nhiều phép biến đổi được chọn từ DCT8 (DCT-VIII)/DST7(DST-VII). Các ma trận biến đổi mới nhất được đưa vào là DST-VII và DCT-VIII. Bảng 1 dưới đây thể hiện các hàm cơ bản của DST/DCT được chọn.

Bảng 1. Các hàm biến đổi cơ bản của DCT-II/VIII và DSTVII cho đầu vào N điểm

Dạng biến đổi	Hàm cơ bản Ti(j), i, j = 0, 1,…, N−1
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$ trong đó, $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

Để giữ tính trực giao của ma trận biến đổi, các ma trận biến đổi được lượng tử hóa chính xác hơn các ma trận biến đổi trong HEVC. Để giữ các giá trị trung gian của các hệ số biến đổi nằm trong phạm vi 16 bit, sau khi biến đổi ngang và sau khi biến đổi đứng, thì tất cả các hệ số đều phải có 10 bit.

Để điều khiển mô hình MTS, các cờ kích hoạt riêng được định ra tại phân mức SPS cho cơ chế nội ảnh và liên ảnh, một cách tương ứng. Khi MTS được kích hoạt tại SPS, chỉ số phân mức CU được báo hiệu để biểu thị cơ chế biến đổi có các dạng biến

đổi theo phương ngang và đứng cho CU hiện thời. Ở đây, MTS chỉ được áp dụng cho mẫu độ chói (luma). Chỉ số phân mức CU MTS (tức là, mts_idx) có thể được báo hiệu khi cả chiều rộng và chiều cao nhỏ hơn hoặc bằng 32 và cờ CBF bằng 1.

Nếu chỉ số CU MTS bằng 0, thì DCT2 được áp dụng theo cả hai phương. Tuy nhiên, nếu chỉ số CU MTS lớn hơn 0, thì các dạng biến đổi theo phương ngang và đứng được xác định theo Bảng 2.

Bảng 2. Bảng ánh xạ biến đổi và báo hiệu

mts_idx	Nội ảnh/liên ảnh	
	Ngang	Đứng
0	DCT2	
1	DST7	DST7
2	DCT8	DST7
3	DST7	DCT8
4	DCT8	DCT8

Biến đổi không phân tách tần số thấp (LFNST)

Để giảm độ phức tạp của DST-7 và DCT-8 kích cỡ lớn, các hệ số biến đổi tần số cao được đặt bằng không đối với các khối DST-7 và DCT-8 có kích cỡ (chiều rộng hoặc chiều cao, hoặc cả chiều rộng và chiều cao) bằng 32. Chỉ các hệ số nằm trong vùng tần số thấp hơn 16x16 được giữ lại.

Theo VVC, LFNST thuận (biến đổi không phân tách tần số thấp) 120, còn được hiểu là biến đổi phụ suy giảm, được áp dụng giữa phép biến đổi chính thuận 110 và phép lượng tử hóa 130 (tại bộ mã hoá) và LFNST ngược 150 được áp dụng giữa phép giải lượng tử 140 và phép biến đổi chính ngược 160 (tại bộ giải mã) như được thể hiện ở Fig. 1. Theo LFNST, phép biến đổi không phân tách 4x4 hoặc phép biến đổi không phân tách 8x8 được áp dụng theo kích cỡ khối. Ví dụ, LFNST 4x4 được áp dụng cho các khối nhỏ (tức là, min (width, height) < 8) và LFNST 8x8 được áp dụng cho các khối lớn hơn (tức là, min (width, height) > 4). Trên Fig. 1, vùng điền dấu chấm 122 tương ứng với 16 hệ số đầu vào cho LFNST thuận 4x4 hoặc 48 hệ số đầu vào cho LFNST thuận 8x8. Vùng điền dấu chấm 152 tương ứng với 8 hoặc 16 hệ số đầu vào cho LFNST ngược 4x4 hoặc 8 hoặc 16 hệ số đầu vào cho LFNST ngược 8x8. Trong trường hợp này, đầu vào

cho phép biến đổi chính thuận là các tín hiệu dư dự đoán và đầu ra từ phép biến đổi chính ngược là tín hiệu dư được khôi phục.

Việc áp dụng phép biến đổi không phân tách, mà được sử dụng trong LFNST, được mô tả ở ví dụ dưới đây. Để áp dụng LFNST 4x4, khối đầu vào X 4x4,

$$X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix}$$

trước tiên được biểu diễn dưới dạng véctơ $\vec{X}$:

$$\vec{X} = [X_{00} X_{01} X_{02} X_{03} X_{10} X_{11} X_{12} X1_3 X_{20} X_{21} X_{22} X_{23} X_{30} X_{31} X_{32} X_{33}]^T$$

Biến đổi không phân tách được tính là $\vec{F} = T \cdot \vec{X}$, trong đó $\vec{F}$ biểu thị véctơ hệ số biến đổi, và T là ma trận biến đổi 16x16. Hệ số véctơ $\vec{F}$ 16x1 được tái tạo lại thành khối 4x4 bằng cách sử dụng lệnh quét (tức là, ngang, đứng hoặc chéo) đối với khối đó. Các hệ số có chỉ số nhỏ hơn sẽ được thay bằng chỉ số quét nhỏ hơn trong khối hệ số 4x4.

Biến đổi không phân tách suy giảm

Phép biến đổi không phân tách tần số thấp (low-frequency non-separable transform: LFNST) dựa theo phương pháp nhân ma trận trực tiếp để áp dụng phép biến đổi không phân tách sao cho nó được thực hiện trong một lần duy nhất mà không lặp lại nhiều lần. Tuy nhiên, kích thước ma trận biến đổi không phân tách cần được giảm để giảm thiểu mức độ phức tạp tính toán và không gian bộ nhớ để lưu các hệ số biến đổi. Vì vậy, phương pháp biến đổi không phân tách suy giảm RST (Reduced non-Separate Transform - RST) được sử dụng trong LFNST. Ý tưởng chính của phép biến đổi không phân tách suy giảm là ánh xạ véctơ N chiều thành véctơ R chiều ở không gian khác, trong đó N/R (R < N) là hệ số suy giảm và N thường bằng 64 đối với các biến đổi phụ không phân tách NSST 8x8 (Non-Separable Secondary Transform - NSST). Vì vậy, thay cho ma trận NxN, ma trận RST trở thành ma trận R×N sau:

$$T_{RxN} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{1N} \\ t_{21} & t_{22} & t_{23} & & t_{2N} \\ & \vdots & & \ddots & \vdots \\ t_{R1} & t_{R2} & t_{R3} & \cdots & t_{RN} \end{bmatrix}$$

trong đó R hàng của ma trận biến đổi là R chiều của không gian N chiều. Ma trận

biến đổi ngược cho RT là chuyển vị của phép biến đổi thuận. Đối với LFNST 8x8, hệ số suy giảm bằng 4 được áp dụng. Trong trường hợp này, ma trận trực tiếp 64x64, mà thường được sử dụng cho ma trận biến đổi không phân tách 8x8, được giảm xuống thành ma trận trực tiếp 16x48. Vì vậy, ma trận RST ngược 48×16 được sử dụng tại bộ giải mã để tạo ra các hệ số biến đổi nhân (chính) ở các vùng trên cùng-bên trái 8×8. Khi các ma trận 16x48 được áp dụng thay cho 16x64 với cùng cấu hình thiết lập biến đổi, mỗi chúng lấy 48 dữ liệu đầu vào từ ba khối 4x4 ở khối 8x8 trên cùng-bên trái không chứa khối 4x4 bên phải-dưới cùng.

Nhờ kích thước được giảm, nên việc sử dụng bộ nhớ để lưu toàn bộ các ma trận LFNST được giảm từ 10KB xuống 8KB với độ sụt hiệu suất chấp nhận được. Để giảm độ phức tạp tính toán, LFNST được ràng buộc sao cho được áp dụng chỉ khi toàn bộ các hệ số nằm ngoài nhóm con hệ số đầu tiên không có nghĩa. Vì vậy, toàn bộ các hệ số chỉ theo biến đổi chính phải bằng 0 khi LFNST được áp dụng. Điều này cho phép thiết đặt điều kiện đối với báo hiệu chỉ số LFNST ở vị trí có nghĩa cuối. Do vậy, nó giúp tránh phải quét hệ số phụ thêm trong thiết kế LFNST hiện tại, vốn là việc cần thiết để kiểm tra các hệ số có nghĩa chỉ ở các vị trí cụ thể. Phép xử lý trường hợp xấu nhất của LFNST, khi thực hiện nhiều phép nhân trên điểm ảnh, sẽ ràng buộc các biến đổi không phân tách cho các khối 4x4 và 8x8 thành các biến đổi 8x16 và 8x48, một cách tương ứng. Đối với những trường hợp này, vị trí quét có nghĩa cuối cùng phải nhỏ hơn 8 khi LFNST được áp dụng cho các kích cỡ khác nhỏ hơn 16. Đối với các khối có hình dạng 4xN và Nx4 và N >= 8, việc ràng buộc đã nêu có nghĩa là LFNST được áp dụng chỉ một lần và được áp dụng chỉ cho vùng 4x4 trên cùng-bên trái. Vì toàn bộ các hệ số chỉ theo biến đổi chính bằng 0 khi LFNST được áp dụng, nên số lượng các thuật toán dùng cho các biến đổi chính được giảm ở những trường hợp vừa nêu. Xét từ bộ mã hóa, việc lượng tử hóa các hệ số được đơn giản hóa đáng kể khi các biến đổi LFNST được kiểm tra. Phép lượng tử tối ưu độ méo phải được thực hiện nhiều nhất cho 8 hoặc 16 hệ số theo thứ tự quét, và các hệ số còn lại được thiết đặt bằng 0.

Chọn biến đổi cho LFNST

Có tổng cộng 4 thiết lập biến đổi và 2 ma trận (nhân) biến đổi không phân tách cho một thiết lập biến đổi được sử dụng ở LFNST. Ánh xạ từ cơ chế dự đoán nội ảnh thành thiết lập biến đổi được định trước như được thể hiện ở Bảng 3 dưới đây. Nếu một trong số ba cơ chế mô hình tuyến tính thành phần chéo CCLM (Cross-Component Linear

Model - CCLM) (tức là, INTRA_LT_CCLM, INTRA_T_CCLM hoặc INTRA_L_CCLM như được biểu diễn bằng 81 <= predModeIntra <= 83) được sử dụng cho khối hiện thời, thì thiết lập biến đổi 0 hoặc cơ chế dự đoán nội ảnh được chọn cho khối màu (chroma) hiện thời. Đối với mỗi thiết lập biến đổi, ứng viên biến đổi phụ không phân tách (hay còn được gọi là ma trận biến đổi không phân tách) được xác định tiếp bằng chỉ số LFNST được báo hiệu hiện. Chỉ số thiết lập biến đổi được báo hiệu trong luồng bit một lần cho mỗi CU nội ảnh sau các hệ số biến đổi.

Bảng 3. Bảng chọn biến đổi cho LFNST

IntraPredMode	Chỉ số thiết lập biến đổi
IntraPredMode < 0	1
0 <= IntraPredMode <= 1	0
2 <= IntraPredMode <= 12	1
13 <= IntraPredMode <= 23	2
24 <= IntraPredMode <= 44	3
45 <= IntraPredMode <= 55	2
56 <= IntraPredMode<= 80	1
81 = IntraPredMode<= 83	0

Báo hiệu chỉ số LFNST và tương tác với các công cụ khác

Vì LFNST được giới hạn sao cho được áp dụng chỉ khi toàn bộ các hệ số nằm ngoài nhóm con hệ số đầu tiên là không có nghĩa, nên lập mã chỉ số LFNST (phân mức CU) phụ thuộc vào vị trí của hệ số có nghĩa cuối cùng. Ngoài ra, chỉ số LFNST được lập mã thuộc tính (ngữ cảnh). Tuy nhiên, chỉ số LFNST không phụ thuộc vào cơ chế dự đoán nội ảnh và ít nhất một vùng trống được lập mã thuộc tính. Ngoài ra, LFNST được áp dụng cho CU nội ảnh ở cả lát (slice) nội ảnh và liên ảnh cho cả luma và/hoặc chroma. Nếu cây lưỡng được sử dụng, thì các tham số LFNST cho luma và chroma được báo hiệu riêng. Đối với slice liên ảnh (tức là, cây lưỡng được hủy), một chỉ số LFNST duy nhất được báo hiệu và được sử dụng cho cả luma và/hoặc chroma.

Giả sử rằng CU lớn lớn hơn 64x64 được phân chia ẩn (phân lát TU) do giới hạn kích cỡ biến đổi lớn nhất hiện có (tức là, kích cỡ 64x64 hoặc kích cỡ được thiết lập theo

cấu hình), phép tìm kiếm chỉ số LFNST có thể làm tăng bộ đệm dữ liệu lên bốn lần đối với số lượng các trạng thái truyền giải mã nhất định. Vì vậy, kích cỡ lớn nhất mà LFNST được phép được giới hạn là 64x64 hoặc kích cỡ biến đổi lớn nhất. Chú ý rằng MTS được sử dụng chỉ khi LFNST không được dùng.

Như được đề xuất trong JVET-P0058 (T. Tsukuba, và các cộng sự, "CE8-2.1: Bỏ qua biến đổi cho chroma bằng cách giới hạn số lượng lớn nhất vùng trống được lập mã thuộc tính khi lập mã phần dư TS," 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, Tài liệu: JVET-P0058), tài liệu này giới thiệu phép bỏ qua biến đổi TS (Transform Skip - TS) dùng cho chroma và áp dụng lập mã phần dư TS cho khối chroma được bỏ qua biến đổi. Ví dụ, TS được sử dụng cho chroma theo mọi định dạng mẫu chroma. Ngoài ra, vì phép điều chế mã xung vi sai dựa vào khối BDPCM (Block-based Delta Pulse Code Modulation - BDPCM) sử dụng TS, nên BDPCM có thể được sử dụng chỉ khi điều kiện sử dụng TS được thoả mãn. Điều kiện sử dụng TS chứa điều kiện ràng buộc kích cỡ mà nghĩa là khi chiều rộng khối nhỏ hơn hoặc bằng kích cỡ bỏ qua biến đổi lớn nhất (MaxTsSize) và chiều cao khối nhỏ hơn hoặc bằng MaxTsSize. Nếu điều kiện vừa nêu được thoả mãn, TS có thể được sử dụng. MaxTsSize là số nguyên hoặc biến cố định bằng 1 << (log2_transform_skip_max_size_minus2 + 2), trong đó log2_transform_skip_max_size_minus2 xác định kích cỡ khối lớn nhất được sử dụng cho phép bỏ qua biến đổi. log2_transform_skip_max_size_minus2 sẽ nằm trong khoảng từ 0 đến 3 và được xem là bằng 0 khi không xuất hiện.

Theo VVC, điều kiện ràng buộc kích cỡ đối với TS cho luma là nếu tbWidth <= MaxTsSize && tbHeight <= MaxTsSize, thì TS có thể được sử dụng.

Theo VVC, điều kiện ràng buộc kích cỡ đối với TS cho chroma là nếu wC <= MaxTsSize && hC <= MaxTsSize, thì TS có thể được sử dụng.

Ở những điều kiện ràng buộc được nêu ở trên, wC = tbWidth / SubWidthC và hC = tbHeight / SubHeightC. TbWidth là chiều rộng khối cho luma và tbHeight là chiều cao khối cho luma. Các biến SubWidthC và SubHeightC được xác định ở Bảng 4 dưới đây tuỳ theo cấu trúc mẫu định dạng chroma, mà được xác định bằng chroma_format_idc và separate_colour_plane_flag. Các trị số khác của chroma_format_idc, SubWidthC và SubHeightC có thể được xác định sau.

Bảng 4. Mô tả các biến SubWidthC và SubHeightC

chroma_format_idc	separate_colour_plane_flag	Định dạng chroma	SubWidthC	SubHeightC
0	0	Đơn sắc	1	1
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	1	1

Điều kiện báo hiệu chi tiết cho cơ chế bỏ qua biến đổi cho từng thành phần được thể hiện ở Bảng 5 dưới đây.

Bảng 5. Điều kiện báo hiệu cho cơ chế bỏ qua biến đổi cho từng thành phần

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex, chType) {	**Bộ mô tả**
if(IntraSubPartitionsSplitType != ISP_NO_SPLIT && treeType == SINGLE_TREE && subTuIndex == NumIntraSubPartitions − 1) {	
xC = CbPosX[chType][x0][y0]	
yC = CbPosY[chType][x0][y0]	
wC = CbWidth[chType][x0][y0] / SubWidthC	
hC = CbHeight[chType][x0][y0] / SubHeightC	
} else {	
xC = x0	
yC = y0	
wC = tbWidth / SubWidthC	
hC = tbHeight / SubHeightC	
}	

chromaAvailable = treeType != DUAL_TREE_LUMA && ChromaArrayType != 0 && (IntraSubPartitionsSplitType == ISP_NO_SPLIT || (IntraSubPartitionsSplitType != ISP_NO_SPLIT && subTuIndex = = NumIntraSubPartitions − 1))	
if((treeType == SINGLE_TREE || treeType == DUAL_TREE_CHROMA) && ChromaArrayType != 0) {	
if((IntraSubPartitionsSplitType = = ISP_NO_SPLIT && !(cu_sbt_flag && ((subTuIndex = = 0 && cu_sbt_pos_flag) || (subTuIndex = = 1 && !cu_sbt_pos_flag))) || (IntraSubPartitionsSplitType != ISP_NO_SPLIT && (subTuIndex = = NumIntraSubPartitions − 1))) {	
tu_cbf_cb[xC][yC]	ae(v)
tu_cbf_cr[xC][yC]	ae(v)
}	
}	
if(treeType = = SINGLE_TREE | | treeType = = DUAL_TREE_LUMA) {	
if((IntraSubPartitionsSplitType = = ISP_NO_SPLIT && !(cu_sbt_flag && ((subTuIndex = = 0 && cu_sbt_pos_flag) || (subTuIndex = = 1 && !cu_sbt_pos_flag))) && (CuPredMode[chType][x0][y0] == MODE_INTRA || (chromaAvailable && (tu_cbf_cb[xC][yC] || tu_cbf_cr[xC][yC])) || CbWidth[chType][x0][y0] > MaxTbSizeY || CbHeight[chType][x0][y0] > MaxTbSizeY)) || (IntraSubPartitionsSplitType != ISP_NO_SPLIT && (subTuIndex < NumIntraSubPartitions − 1 | | !InferTuCbfLuma)))	
tu_cbf_luma[x0][y0]	ae(v)

if(IntraSubPartitionsSplitType != ISP_NO_SPLIT)	
InferTuCbfLuma = InferTuCbfLuma && !tu_cbf_luma[x0][y0]	
}	
if((CbWidth[chType][x0][y0] > 64 || CbHeight[chType][x0][y0] > 64 || tu_cbf_luma[x0][y0] || (chromaAvailable && (tu_cbf_cb[xC][yC] || tu_cbf_cr[xC][yC])) && treeType != DUAL_TREE_CHROMA) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_abs	ae(v)
if(cu_qp_delta_abs)	
cu_qp_delta_sign_flag	ae(v)
}	
}	
if((CbWidth[chType][x0][y0] > 64 || CbHeight[chType][x0][y0] > 64 || (chromaAvailable && (tu_cbf_cb[xC][yC] || tu_cbf_cr[xC][yC]))) && treeType != DUAL_TREE_LUMA) {	
if(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_offset_flag	ae(v)
if(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	
cu_chroma_qp_offset_idx	ae(v)
}	
}	
if(sps_joint_cbcr_enabled_flag && ((CuPredMode[chType][x0][y0] = = MODE_INTRA && (tu_cbf_cb[xC][yC] || tu_cbf_cr[xC][yC])) || (tu_cbf_cb[xC][yC] && tu_cbf_cr[xC][yC])) && chromaAvailable)	

tu_joint_cbcr_residual_flag[xC][yC]	ae(v)
if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA) {	
if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][0] && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && !cu_sbt_flag)	
transform_skip_flag[x0][y0][0]	ae(v)
if(!transform_skip_flag[x0][y0][0])	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
else	
residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	
if(tu_cbf_cb[xC][yC] && treeType != DUAL_TREE_LUMA) {	
if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][1] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag)	
transform_skip_flag[xC][yC][1]	ae(v)
if(!transform_skip_flag[xC][yC][1])	
residual_coding(xC, yC, Log2(wC), Log2(hC), 1)	
else	
residual_ts_coding(xC, yC, Log2(wC), Log2(hC), 1)	
}	

if(tu_cbf_cr[xC][yC] && treeType != DUAL_TREE_LUMA &&
!(tu_cbf_cb[xC][yC] && tu_joint_cbcr_residual_flag[xC][yC]))
{	

if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][2] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag)	
transform_skip_flag[xC][yC][2]	ae(v)

if(!transform_skip_flag[xC][yC][2])	
residual_coding(xC, yC, Log2(wC), Log2(hC), 2)	
else	
residual_ts_coding(xC, yC, Log2(wC), Log2(hC), 2)	
}	
}	

Ở bảng cú pháp nêu trên, **transform_skip_flag[x0][y0][cIdx]** xác định liệu phép biến đổi được áp dụng cho khối biến đổi liên quan hay không. Các chỉ số mảng x0, y0 xác định vị trí (x0, y0) của mẫu luma trên cùng-bên trái của khối biến đổi được xét so với mẫu luma trên cùng-bên trái của ảnh. Chỉ số mảng cIdx xác định chỉ số cho thành phần màu; chỉ số đó bằng 0 cho Y, 1 cho Cb, và 2 cho Cr. Transform_skip_flag[x0][y0][cIdx] bằng 1 xác định rằng không có phép biến đổi nào được áp dụng cho khối biến đổi liên quan. Transform_skip_flag[x0][y0][cIdx] bằng 0 xác định liệu có phép biến đổi được áp dụng cho khối biến đổi liên quan hay không phụ thuộc vào các thành phần cú pháp khác.

Khi transform_skip_flag[x0][y0][cIdx] không xuất hiện, nó được suy ra như sau:

–Nếu BdpcmFlag[x0][y0][cIdx] bằng 1, transform_skip_flag[x0][y0][cIdx] được xem là bằng 1.

–Nếu không thì (BdpcmFlag[x0][y0][cIdx] bằng 0), transform_skip_flag[x0][y0][cIdx] được xem là bằng 0.

Theo trên, BdpcmFlag[x0][y0][cIdx] là biến tương ứng với cờ BDPCM nội ảnh cho thành phần luma (tức là, cIdx = 0) hoặc cho thành phần chroma (tức là, cIdx = 1 hoặc 2).

Bản chất kỹ thuật của sáng chế

Phương pháp và thiết bị lập mã viđeo bằng cách sử dụng cơ chế biến đổi không phân tách tần số thấp (LFNST) và cơ chế bỏ qua biến đổi TS (Transform Skip - TS) được bộc lộ. Theo sáng chế, dữ liệu đầu vào liên quan đến đơn vị lập mã hiện thời (CU) ở khung hình hiện thời được nhận, trong đó CU được phân chia thành một hoặc nhiều

khối biến đổi TB (Transform Block - TB) và dữ liệu đầu vào tương ứng với dữ liệu dư tại bộ mã hoá viđeo và dữ liệu đầu vào tương ứng với dữ liệu được lập mã của CU hiện thời tại bộ giải mã viđeo. Cú pháp được xác định tại bộ mã hoá hoặc tại bộ giải mã, trong đó việc xác định cú pháp vừa nêu được thực hiện bằng cách báo hiệu cú pháp tại bộ mã hoá hoặc bằng cách phân tích cú pháp tại bộ giải mã nếu một hoặc nhiều điều kiện được thoả mãn. Cú pháp (ví dụ chỉ số LFNST) biểu thị liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc loại nhân LFNST được áp dụng khi cơ chế LFNST được áp dụng, và các điều kiện vừa nêu bao gồm điều kiện mong muốn tương ứng với tất cả TB mong muốn ở bộ TB mong muốn có chỉ báo cơ chế TS là sai (false), và bộ TB mong muốn được chọn từ các TB ở CU hiện thời. CU hiện thời được mã hoá tại bộ mã hoá hoặc được giải mã tại bộ giải mã theo cơ chế LFNST như được biểu diễn bằng cú pháp. Theo một phương án, cú pháp tham chiếu đến chỉ số LFNST.

Theo một phương án, ở cây phân chia luma, CU hiện thời tương ứng với khối lập mã luma (CB), và bộ TB mong muốn tương ứng với một hoặc nhiều TB luma.

Theo một phương án, ở cây phân chia chroma, CU hiện thời tương ứng với một hoặc nhiều khối lập mã chroma, và bộ TB mong muốn tương ứng với một hoặc nhiều TB chroma.

Theo một phương án, ở cây phân chia đơn, CU hiện thời tương ứng với một khối lập mã luma và một hoặc nhiều khối lập mã chroma, và bộ TB mong muốn tương ứng với một hoặc nhiều TB luma và một hoặc nhiều TB chroma.

Theo một phương án, bộ TB mong muốn bao gồm TB (các TB) biến đổi thứ nhất cho từng khối lập mã (CB) ở CU hiện thời. Ví dụ, ở cây phân chia luma, bộ TB mong muốn bao gồm TB luma thứ nhất cho CB luma ở CU hiện thời. Ở ví dụ khác, ở cây phân chia chroma, bộ TB mong muốn bao gồm TB Cb thứ nhất cho CB Cb ở CU hiện thời và TB Cr thứ nhất cho CB Cr ở CU hiện thời. Ở ví dụ khác, ở cây phân chia đơn, bộ TB mong muốn bao gồm TB luma thứ nhất cho CB luma ở CU hiện thời, TB Cb thứ nhất cho CB Cb ở CU hiện thời, và TB Cr thứ nhất cho CB Cr ở CU hiện thời.

Theo một phương án, biến được sử dụng để ghi nhận liệu có báo hiệu hoặc phân tích cú pháp cho cơ chế LFNST hay không. Theo phương án khác, cú pháp vừa nêu được suy ra dưới dạng cơ chế LFNST không được áp dụng cho CU hiện thời khi chỉ báo cơ chế TS là đúng (true).

Theo một phương án, cơ chế LFNST không được sử dụng cho TB ở CU hiện thời nếu chỉ số LFNST cho CU hiện thời bằng 0, và trong đó chỉ số LFNST có trị số lớn hơn 0 biểu thị một trong số các ma trận biến đổi tần số thấp được chọn.

Mô tả vắn tắt các hình vẽ

Fig. 1 minh hoạ ví dụ về lập mã viđeo tích hợp cơ chế LFNST (biến đổi không phân tách tần số thấp); và

Fig. 2 minh hoạ lưu đồ của hệ thống lập mã được lấy làm ví dụ có tích hợp báo hiệu LFNST (biến đổi không phân tách tần số thấp) được ràng buộc theo phương án của sáng chế.

Mô tả chi tiết sáng chế

Phần mô tả dưới đây là phần mô tả cách thức tốt nhất thực hiện sáng chế. Phần mô tả này được đưa ra nhằm mục đích minh họa các nguyên lý chung của sáng chế chứ không nhằm giới hạn sáng chế. Phạm vi của sáng chế được xác định đúng nhất bằng cách tham chiếu các điểm yêu cầu bảo hộ đính kèm.

Các kết hợp giữa LFNST với phép bỏ qua biến đổi sẽ không được phép vì khi phép bỏ qua biến đổi được áp dụng, không có phép biến đổi (biến đổi nhân/chính và/hoặc biến đổi phụ) sẽ được sử dụng. Theo VVC dự thảo 7 (B. Bross, và các cộng sự, "Lập mã viđeo đa năng (dự thảo 7)", Nhóm chuyên gia viđeo chung (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, Tài liệu: JVET-P2001), cú pháp cho cơ chế bỏ qua biến đổi được báo hiệu/được phân tích tại phân mức TB. Mặt khác, cú pháp cho LFNST được báo hiệu/được phân tích tại phân mức CU sau khi toàn bộ các TU/TB nằm trong các CU/CB được báo hiệu/được phân tích. Vì vậy, theo VVC dự thảo 7 (như được thể hiện ở Bảng 6A), các điều kiện báo hiệu/phân tích LFNST sẽ xem xét đến cờ bỏ qua biến đổi cho luma như sau. Như được thể hiện ở bảng cú pháp dưới đây, các điều kiện hiện có bao gồm kiểm tra phép bỏ qua biến đổi luma (tức là, transform_skip_flag[x0][y0][0] == 0) để ngăn không cho kết hợp đã nêu. Đối với kiểm tra vừa nêu, mô hình kiểm nghiệm VVC phiên bản 7 (VTM7, J. Chen, và các cộng sự, "Thuật toán mô tả cho lập mã viđeo đa năng và mô hình kiểm nghiệm 7 (VTM 7)", Nhóm chuyên gia viđeo chung (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, Tài liệu: JVET-P2002) đưa ra các mã

tương thích VVC dự thảo 7. Bảng cú pháp dùng để lập mã phần dư theo JVET-P2001 được thể hiện ở Bảng 6B.

Bảng 6A. Các điều kiện báo hiệu/phân tích LFNST theo VVC dự thảo 7

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	**Bộ mô tả**
chType = treeType = = DUAL_TREE_CHROMA ? 1 : 0	
...	
if(CuPredMode[chType][x0][y0] = = MODE_INTRA | | CuPredMode[chType][x0][y0] = = MODE_PLT) {	
if(treeType = = SINGLE_TREE | | treeType = = DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag) {	
palette_coding(x0, y0, cbWidth, cbHeight, treeType)	
} else {	
if(sps_bdpcm_enabled_flag && cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)	
intra_bdpcm_luma_flag	ae(v)
if(intra_bdpcm_luma_flag)	
intra_bdpcm_luma_dir_flag	ae(v)
else {	
...	
}	
}	
}	
if((treeType = = SINGLE_TREE | | treeType = = DUAL_TREE_CHROMA) && ChromaArrayType != 0) {	
if(pred_mode_plt_flag && treeType = = DUAL_TREE_CHROMA)	
palette_coding(x0, y0, cbWidth / SubWidthC, cbHeight / SubHeightC, treeType)	

else {	
if(!cu_act_enabled_flag) {	
if(cbWidth <= MaxTsSize && cbHeight <= MaxTsSize && sps_bdpcm_chroma_enabled_flag) {	
intra_bdpcm_chroma_flag	ae(v)
if(intra_bdpcm_chroma_flag)	
intra_bdpcm_chroma_dir_flag	ae(v)
} else {	
if(CclmEnabled)	
cclm_mode_flag	ae(v)
if(cclm_mode_flag)	
cclm_mode_idx	ae(v)
else	
intra_chroma_pred_mode	ae(v)
}	
}	
}	
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
…	
}	
if(CuPredMode[chType][x0][y0] != MODE_INTRA && !pred_mode_plt_flag && general_merge_flag[x0][y0] = = 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
…	
LfnstDcOnly = 1	
LfnstZeroOutSigCoeffFlag = 1	
MtsZeroOutSigCoeffFlag = 1	

transform_tree(x0, y0, cbWidth, cbHeight, treeType, chType)	
lfnstWidth = (treeType = = DUAL_TREE_CHROMA) ? cbWidth / SubWidthC : ((IntraSubPartitionsSplitType = = ISP_VER_SPLIT) ? cbWidth / NumIntraSubPartitions : cbWidth)	
lfnstHeight = (treeType = = DUAL_TREE_CHROMA) ? cbHeight / SubHeightC : ((IntraSubPartitionsSplitType = = ISP_HOR_SPLIT) ? cbHeight / NumIntraSubPartitions : cbHeight)	
if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag = = 1 && CuPredMode[chType][x0][y0] = = MODE_INTRA && transform_skip_flag[x0][y0][0] = = 0 && (treeType != DUAL_TREE_CHROMA || !intra_mip_flag[x0][y0] || Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) {	
if((IntraSubPartitionsSplitType != ISP_NO_SPLIT || LfnstDcOnly = = 0) && LfnstZeroOutSigCoeffFlag = = 1)	
lfnst_idx	ae(v)
}	
if(treeType != DUAL_TREE_CHROMA && lfnst_idx = = 0 && transform_skip_flag[x0][y0][0] = = 0 && Max(cbWidth, cbHeight) <= 32 && IntraSubPartitionsSplit[x0][y0] = = ISP_NO_SPLIT && cu_sbt_flag = = 0 && MtsZeroOutSigCoeffFlag = = 1 && tu_cbf_luma[x0][y0]) {	
if(((CuPredMode[chType][x0][y0] = = MODE_INTER && sps_explicit_mts_inter_enabled_flag) || (CuPredMode[chType][x0][y0] = = MODE_INTRA && sps_explicit_mts_intra_enabled_flag)))	
mts_idx	ae(v)

}	
}	

Bảng 6B. Bảng cú pháp dùng để lập mã phần dư theo VVC dự thảo 7

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	**Bộ mô tả**
if(sps_mts_enabled_flag && cu_sbt_flag && cIdx = = 0 && log2TbWidth = = 5 && log2TbHeight < 6)	
log2ZoTbWidth = 4	
else	
log2ZoTbWidth = Min(log2TbWidth, 5)	
if(sps_mts_enabled_flag && cu_sbt_flag && cIdx = = 0 && log2TbWidth < 6 && log2TbHeight = = 5)	
log2ZoTbHeight = 4	
else	
log2ZoTbHeight = Min(log2TbHeight, 5)	
if(log2TbWidth > 0)	
last_sig_coeff_x_prefix	ae(v)
if(log2TbHeight > 0)	
last_sig_coeff_y_prefix	ae(v)
if(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ae(v)
if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
log2TbWidth = log2ZoTbWidth	
log2TbHeight = log2ZoTbHeight	
remBinsPass1 = ((1 << (log2TbWidth + log2TbHeight)) * 7) >> 2	
log2SbW = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
log2SbH = log2SbW	
if(log2TbWidth + log2TbHeight > 3) {	
if(log2TbWidth < 2) {	
log2SbW = log2TbWidth	

log2SbH = 4 − log2SbW	
} else if(log2TbHeight < 2) {	
log2SbH = log2TbHeight	
log2SbW = 4 − log2SbH	
}	
}	
numSbCoeff = 1 << (log2SbW + log2SbH)	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight − (log2SbW + log2SbH))) − 1	
do {	
if(lastScanPos = = 0) {	
lastScanPos = numSbCoeff	
lastSubBlock− −	
}	
lastScanPos− −	
xS = DiagScanOrder[log2TbWidth − log2SbW][log2TbHeight − log2SbH] [lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth − log2SbW][log2TbHeight − log2SbH] [lastSubBlock][1]	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][1]	
} while((xC != LastSignificantCoeffX) | | (yC != LastSignificantCoeffY))	
if(lastSubBlock = = 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastScanPos > 0)	

LfnstDcOnly = 0	
if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) || (lastScanPos > 7 && (log2TbWidth = = 2 || log2TbWidth = = 3) && log2TbWidth = = log2TbHeight))	
LfnstZeroOutSigCoeffFlag = 0	
if((LastSignificantCoeffX > 15 || LastSignificantCoeffY > 15) && cIdx = = 0)	
MtsZeroOutSigCoeffFlag = 0	
...	
}	
}	
}	

Theo các điều kiện báo hiệu/phân tích cú pháp LFNST hiện đang có, hai vấn đề được phát hiện. Một vấn đề là khi luma và chroma sử dụng các cây phân chia khác nhau, nó không thể tìm ra cờ bỏ qua biến đổi cho luma (tức là, transform_skip_flag [x0][y0][0]) khi CU hiện thời nằm ở cây phân chia chroma (tức là, trường hợp có treeType == DUAL_TREE_CHROMA). Vấn đề khác gây ra bởi sự mở rộng của phép bỏ qua biến đổi cho chroma như được bộc lộ trong JVET-P0058. Phép kiểm tra được nêu ở trên sẽ được mở rộng để bao gồm các phép kiểm tra Cb và Cr. Một số phương pháp được đề xuất để giải quyết các vấn đề vừa nêu.

Phép kiểm tra được đề xuất sẽ tiến hành xem xét điều kiện của cờ (các cờ) bỏ qua biến đổi của M TB trong CU. Đối với TB có một hoặc nhiều mức hệ số biến đổi không bằng 0, cờ bỏ qua biến đổi cho TB được sử dụng để biểu thị liệu các thuật toán biến đổi được áp dụng cho TB hay không và phép kiểm tra được đề xuất được sử dụng để ngăn không cho TB có kết hợp giữa LFNST với phép bỏ qua biến đổi. Như đã nêu trước đó, ở cây phân chia tương ứng mà có thể là cây phân chia luma (DUAL_TREE_LUMA), cây phân chia chroma (DUAL_TREE_CHROMA), hoặc cây phân chia đơn (SINGLE_TREE), có một hoặc nhiều TB ở CU hiện thời. M TB tương ứng với bộ các TB được chọn, được gọi là các TB mong muốn. Điều kiện của cờ (các cờ) bỏ qua biến đổi của bộ TB mong muốn được kiểm tra. Việc đạt yêu cầu kiểm tra nghĩa là cờ (các

cờ) bỏ qua biến đổi cho toàn bộ các M TB là sai (false) (tức là, cờ (các cờ) bỏ qua biến đổi 0); nói cách khác, việc đạt yêu cầu kiểm tra nghĩa là điều kiện mong muốn (tương ứng với tất cả TB mong muốn ở bộ TB mong muốn có chỉ báo cơ chế TS là false) được thoả mãn. Nói cách khác, điều kiện cờ (các cờ) cơ chế bỏ qua biến đổi được thoả mãn nếu không có TB (các TB) được chọn nào sử dụng cơ chế bỏ qua biến đổi. Sau khi đạt yêu cầu kiểm tra (tức là, điều kiện cờ (các cờ) cơ chế bỏ qua biến đổi được thoả mãn), các điều kiện báo hiệu/phân tích LFNST liên quan đến cơ chế bỏ qua biến đổi được thoả mãn và cú pháp cho LFNST có thể được báo hiệu/được phân tích nếu các điều kiện báo hiệu/phân tích LFNST khác cũng được thoả mãn.

Theo một phương án, M TB chỉ chứa thành phần thứ nhất ở từng cây phân chia luma/chroma. Ví dụ của bảng cú pháp được đề xuất được thể hiện ở Bảng 7.

Bảng 7. Bảng cú pháp được lấy làm ví dụ về điều kiện báo hiệu/phân tích LFNST theo một phương án của sáng chế.

If(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag = = 1 &&
CuPredMode[chType][x0][y0] = = MODE_INTRA &&
transform_skip_flag[x0][y0][chType] = = 0 &&
(treeType != DUAL_TREE_CHROMA | | !intra_mip_flag[x0][y0] | |
Min(lfnstWidth, lfnstHeight) >= 16) &&
Max(cbWidth, cbHeight) <= MaxTbSizeY) {	

if((IntraSubPartitionsSplitType != ISP_NO_SPLIT | | LfnstDcOnly = = 0) && LfnstZeroOutSigCoeffFlag = = 1)	
lfnst_idx	ae(v)
}	

Theo phương án khác, để cây phân chia đơn được sử dụng cho cả thành phần luma và thành phần chroma, M TB chứa một hoặc nhiều thành phần.

Theo một phương án phụ, M TB tham chiếu đến một thành phần được chọn. Ví dụ, M TB tham chiếu đến thành phần thứ nhất. Theo ví dụ khác, M TB tham chiếu đến Y (tức là, thành phần luma). Bảng cú pháp được lấy làm ví dụ theo một phương án được thể hiện ở Bảng 8. Theo ví dụ khác, M TB có thể là một thành

phần bất kỳ trong cây phân chia.

Bảng 8. Bảng cú pháp được lấy làm ví dụ về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag = = 1 && CuPredMode[chType][x0][y0] = = MODE_INTRA && (treeType = = SINGLE_TREE && transform_skip_flag[x0][y0][0] = = 0) && (treeType != DUAL_TREE_CHROMA || !intra_mip_flag[x0][y0] || Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) {	
if((IntraSubPartitionsSplitType != ISP_NO_SPLIT || LfnstDcOnly = = 0) && LfnstZeroOutSigCoeffFlag = = 1)	
lfnst_idx	ae(v)
}	

Theo phương án khác, khi cây phân chia không là cây chroma (tức là, cây phân chia chứa thành phần Y (tức là, luma)), M TB tham chiếu đến TB (các TB) Y (tức là, luma). Bảng cú pháp được lấy làm ví dụ theo một phương án được thể hiện ở Bảng 9.

Bảng 9. Bảng cú pháp được lấy làm ví dụ về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag = = 1 && CuPredMode[chType][x0][y0] = = MODE_INTRA && (treeType != DUAL_TREE_CHROMA && transform_skip_flag[x0][y0][0] = = 0) && (treeType != DUAL_TREE_CHROMA || !intra_mip_flag[x0][y0] || Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) {	

if((IntraSubPartitionsSplitType != ISP_NO_SPLIT | | LfnstDcOnly == = 0) && LfnstZeroOutSigCoeffFlag = = 1)	
lfnst_idx	ae(v)
}	

Theo phương án khác, đối với cây phân chia chroma, M TB chứa toàn bộ các thành phần chroma (ví dụ Cb và Cr). Nếu cờ bỏ qua biến đổi bất kỳ cho các thành phần chroma là false (tức là, cờ bỏ qua biến đổi bằng 0), thì phép kiểm tra đó đạt.

Theo phương án khác, đối với cây phân chia luma, M TB bao gồm tất cả các thành phần (ví dụ Y). Nếu cờ bỏ qua biến đổi bất kỳ cho các thành phần này là false (tức là, cờ bỏ qua biến đổi bằng 0), thì phép kiểm tra đó đạt. Bảng cú pháp được lấy làm ví dụ theo phương án này được thể hiện ở Bảng 10.

Theo phương án khác, đối với cây đơn được sử dụng cho các thành phần luma và chroma, M TB bao gồm tất cả các thành phần (ví dụ Y, Cb, và Cr). Nếu cờ bỏ qua biến đổi bất kỳ cho các thành phần này là false (tức là, cờ bỏ qua biến đổi bằng 0), thì phép kiểm tra đó đạt. Bảng cú pháp được lấy làm ví dụ theo phương án này được thể hiện ở Bảng 10.

Bảng 10. Bảng cú pháp được lấy làm ví dụ về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag = = 1 &&
CuPredMode[chType][x0][y0] = = MODE_INTRA &&
(treeType == DUAL_TREE_CHROMA?
(transform_skip_flag[x0][y0][1] = = 0 ||
transform_skip_flag[x0][y0][2] = = 0) : (treeType ==
DUAL_TREE_LUMA ? transform_skip_flag[x0][y0][0] = = 0:
(transform_skip_flag[x0][y0][0] = = 0 ||
transform_skip_flag[x0][y0][1] = = 0 ||
transform_skip_flag[x0][y0][2] = = 0))) &&
(treeType != DUAL_TREE_CHROMA | | !intra_mip_flag[x0][y0] | |
Min(lfnstWidth, lfnstHeight) >= 16) &&
Max(cbWidth, cbHeight) <= MaxTbSizeY) {	

if((IntraSubPartitionsSplitType != ISP_NO_SPLIT | | LfnstDcOnly = = 0) && LfnstZeroOutSigCoeffFlag = = 1)	
lfnst_idx	ae(v)
}	

Theo phương án khác, đối với cây phân chia chroma, M TB chứa toàn bộ các thành phần chroma (ví dụ Cb và Cr). Nếu toàn bộ các cờ bỏ qua biến đổi cho các thành phần chroma là false (tức là, cờ bỏ qua biến đổi bằng 0), thì phép kiểm tra đó đạt.

Theo phương án khác, đối với cây phân chia luma, M TB bao gồm toàn bộ các thành phần luma (ví dụ Y). Nếu toàn bộ các cờ bỏ qua biến đổi cho các thành phần này là false (tức là, cờ bỏ qua biến đổi bằng 0), thì phép kiểm tra đó đạt. Bảng cú pháp được lấy làm ví dụ theo phương án này được thể hiện ở Bảng 11.

Theo phương án khác, đối với cây đơn được sử dụng cho các thành phần luma và chroma, M TB bao gồm tất cả các thành phần (ví dụ Y, Cb, và Cr). Nếu toàn bộ các cờ bỏ qua biến đổi cho các thành phần này là false (tức là, cờ bỏ qua biến đổi bằng 0), thì phép kiểm tra đó đạt. Bảng cú pháp được lấy làm ví dụ theo phương án này được thể hiện ở Bảng 11.

Theo phương án khác nữa, hai hoặc nhiều hơn hai trong số ba phương án nêu trên có thể được kết hợp. Ví dụ, phương án kết hợp vừa nêu có thể chỉ kiểm tra transform_skip_flag luma khi cây phân chia không là cây phân chia chroma (ví dụ không dùng cho DUAL_TREE_CHROMA) và chỉ kiểm tra transform_skip_flag chroma khi cây phân chia không là cây phân chia luma (ví dụ không dùng cho DUAL_TREE_LUMA). Bảng cú pháp được lấy làm ví dụ theo phương án này được thể hiện ở Bảng 11.

Bảng 11. Bảng cú pháp được lấy làm ví dụ về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

If(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag = = 1 &&
CuPredMode[chType][x0][y0] = = MODE_INTRA &&
(treeType == DUAL_TREE_CHROMA?
(transform_skip_flag[x0][y0][1] = = 0 &&
transform_skip_flag[x0][y0][2] = = 0) : (treeType ==
DUAL_TREE_LUMA ? transform_skip_flag[x0][y0][0] = = 0:
(transform_skip_flag[x0][y0][0] = = 0 &&
transform_skip_flag[x0][y0][1] = = 0 &&
transform_skip_flag[x0][y0][2] = = 0))) &&
(treeType != DUAL_TREE_CHROMA | | !intra_mip_flag[x0][y0] | |
Min(lfnstWidth, lfnstHeight) >= 16) &&
Max(cbWidth, cbHeight) <= MaxTbSizeY) {	

if((IntraSubPartitionsSplitType != ISP_NO_SPLIT | | LfnstDcOnly = = 0) && LfnstZeroOutSigCoeffFlag = = 1)	
lfnst_idx	ae(v)
}	

Bảng cú pháp được lấy làm ví dụ khác để kết hợp ba phương án nêu trên được thể hiện ở Bảng 12.

Bảng 12. Bảng cú pháp được lấy làm ví dụ về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 &&CuPredMode[chType][x0][y0] = = MODE_INTRA && (treeType == DUAL_TREE_CHROMA || transform_skip_flag[x0][y0][0] = = 0) && (treeType == DUAL_TREE_LUMA || (transform_skip_flag[x0][y0][1] = = 0 && transform_skip_flag[x0][y0][2] == 0)) && (treeType == DUAL_TREE_CHROMA || !intra_mip_flag[x0][y0] ||Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) {	
if((IntraSubPartitionsSplitType != ISP_NO_SPLIT || LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag = = 1)	
lfnst_idx	ae(v)
}	

Theo phương án khác, khi phép kiểm tra không đạt, thì cú pháp cho LFNST (ví dụ, chỉ số LFNST) không được báo hiệu/được phân tích.

Theo một phương án phụ, cú pháp cho LFNST (chỉ số LFNST) được xem là 0 (tức là, LFNST không được áp dụng).

Theo phương án khác, một tương hợp luồng bit được yêu cầu để xử lý trường hợp khi phép kiểm tra không đạt. Tương hợp luồng bit vừa nêu là như sau. Có yêu cầu tương hợp luồng bit đó là trị số của lfnst_index sẽ không lớn hơn 0 khi phép kiểm tra không đạt. Dưới đây, ví dụ về tương hợp luồng bit được minh hoạ cho trường hợp kiểm tra "cờ (các cờ) bỏ qua biến đổi cho M TB, trong đó M TB chỉ chứa thành phần thứ nhất ở từng cây phân chia luma/chroma". Yêu cầu tương hợp luồng bit tương ứng với trị số của lfnst_index sẽ không lớn hơn 0 khi trị số của cờ bỏ qua biến đổi cho thành phần thứ nhất ở từng cây phân chia luma/chroma (ví dụ, transform_skip_flag[x0][y0][chType], trong

đó nếu treeType == DUAL_TREE_CHROMA, chType biểu thị 1 (tức là, Cb); nếu không thì, chType biểu thị 0 (tức là, Y)) lớn hơn 1.

Theo phương án khác, biến có thể được tạo ra ở câu lệnh hoặc phần mềm để biểu thị liệu có báo hiệu/phân tích cú pháp cho LFNST. Trị số của biến này được cập nhật theo một hoặc nhiều điều kiện báo hiệu/phân tích LFNST hiện có và/hoặc một hoặc nhiều phép kiểm tra được đề xuất ở sáng chế này. Ví dụ, biến này được khởi tạo bằng 1 và nếu bất kỳ điều kiện báo hiệu/phân tích LFNST và/hoặc một hoặc nhiều phép kiểm tra được đề xuất ở sáng chế này không được thoả mãn, thì biến này được chuyển thành 0 và cú pháp cho LFNST không được báo hiệu/được phân tích.

Theo phương án khác, cơ chế kiểm tra hợp nhất được sử dụng cho các cây phân chia khác nhau cho luma và chroma. Ví dụ, khi luma và chroma sử dụng các cây lưỡng (tức là, các cây phân chia riêng), CU luma tuân theo cây phân chia luma và CU chroma tuân theo cây phân chia chroma. Cơ chế kiểm tra hợp nhất vừa nêu là cơ chế mà LFNST được huỷ bỏ nếu bất kỳ một trong số các cờ bỏ qua biến đổi cho tất cả các thành phần ở CU hiện thời đang sử dụng phép bỏ qua biến đổi.

Do điều kiện ràng buộc kích cỡ hiện tại đối với LFNST, LFNST có thể được áp dụng khi một CU/CB chứa một TU/TB. Phép kiểm tra có thể xem xét cờ bỏ qua biến đổi cho một TU/TB thay vì nhiều TU/TB. Khi một CU/CB chứa nhiều TU/TB, phép kiểm tra được đề xuất ở trên tuân theo một hoặc nhiều TU/TB ở CU/CB đó. Theo một phương án, phép kiểm tra được đề xuất ở trên tuân theo toàn bộ các TU/TB ở CU/CB đó. Theo phương án khác, phép kiểm tra tuân theo bất kỳ một trong số các TU/TB ở CU/CB đó (ví dụ TU/TB đầu tiên hoặc TU/TB cuối). Ví dụ, ở cây phân chia luma, bộ TB mong muốn bao gồm TB luma thứ nhất cho CB luma ở CU hiện thời. Ở ví dụ khác, ở cây phân chia chroma, bộ TB mong muốn bao gồm TB Cb thứ nhất cho CB Cb ở CU hiện thời và TB Cr thứ nhất cho CB Cr ở CU hiện thời. Ở ví dụ khác, ở cây phân chia đơn, bộ TB mong muốn bao gồm TB luma thứ nhất cho CB luma ở CU hiện thời, TB Cb thứ nhất cho CB Cb ở CU hiện thời, và TB Cr thứ nhất cho CB Cr ở CU hiện thời. Theo phương án khác, phép kiểm tra tuân theo tập con bất kỳ tập con trong số các TU/TB ở CU/CB đó.

Phép kiểm tra tràn khi biến đổi Cb/Cr hợp nhất JCCR (Joint Cb/Cr Transform - JCCR) được đề xuất lần đầu tiên trong JVET-N0054 (J. Lainema, “CE7: Lập mã hợp

nhất các phần dư màu (CE7-1),” ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 14: Geneva, CH, ngày 19 đến 27 tháng 3 năm 2019, Tài liệu: JVET-N0054). Ở TrQuant.cpp, invTransformCbCr( ), trị số cb[x] = -32768 sẽ tràn dạng dữ liệu 16 bit nếu biến đổi Cr = -Cb được áp dụng. Cụ thể là, dòng mã sau:

else if (signedMode == -2) { cr[x] = -cb[x]; }.

Một cách khắc phục đó là có thể bổ sung điều kiện phụ. Ví dụ:

```
else if ( signedMode == -2 ) { cr[x] = (cb[x] == -32768) ? 32767 : -cb[x];
}
```

Theo đó, thay vì biến đổi thông thường, phương án thực hiện sáng chế này sẽ không gây tràn bằng cách thiết đặt Cr = +32767 chứ không phải +32768 khi cb bằng -32768

Việc thiết đặt phần dư Cr bằng 32767 chứ không phải 32768 sẽ không tạo ra sai số sau khi phần dư được cộng vào dự đoán và được cắt thành chiều sâu bit điểm ảnh. Đây là cách áp dụng ít nhất cho các chiều sâu bit lên tới 14 bit.

Một cách khác để khắc phục hiện tượng tràn trong JCCR là thiết đặt CoeffMin thành –(1 << 15) + 1.

Các biến CoeffMin và CoeffMax xác định các giá trị hệ số biến đổi nhỏ nhất và lớn nhất được suy dẫn như sau:

CoeffMin = −(1 << 15)

CoeffMax = (1 << 15) – 1

Ngoài ra, việc sử dụng LFNST có thể bị giới hạn theo một số điều kiện. Ở thiết kế hiện tại, LFNST được áp dụng cho CU nội ảnh ở cả slice nội ảnh và liên ảnh, và luma và/hoặc chroma. Nếu cây lưỡng được sử dụng, thì các tham số LFNST cho luma và chroma được báo hiệu/được phân tích riêng. Đối với slice liên ảnh, trong đó cây lưỡng được huỷ bỏ, chỉ số LFNST duy nhất được báo hiệu/được phân tích và được sử dụng cho Luma và/hoặc chroma. Ở sáng chế này, LFNST chroma được huỷ bỏ theo một số trường hợp.

Theo một phương án, đối với cây đơn, LFNST chroma được huỷ bỏ.

Theo một phương án phụ, khi LFNST chroma được huỷ bỏ, chỉ số LFNST vẫn

được báo hiệu/được phân tích và có thể được sử dụng cho luma.

Theo phương án khác, LFNST chroma được huỷ bỏ.

Theo một phương án phụ, khi LFNST chroma được huỷ bỏ, chỉ số LFNST không được báo hiệu/được phân tích ở cây lưỡng chroma.

Theo phương án khác, LFNST không thể được sử dụng cho TB mặc dù chỉ số LFNST cho CU chứa TB lớn hơn 0. Biến, applyLfnstFlag, được tạo ra để biểu thị liệu LFNST có thể được sử dụng hay không. Nếu applyLfnstFlag bằng 0, LFNST không thể được sử dụng. Nếu applyLfnstFlag bằng 1, LFNST có thể được sử dụng.

Ví dụ, đối với cây đơn, LFNST chroma được huỷ bỏ. Biến applyLfnstFlag được suy dẫn như sau: (trong đó xTbY và yTbY là vị trí mẫu luma tương ứng cho TB, cIdx tham chiếu đến thành phần cho TB (ví dụ cIdx bằng 0 tham chiếu đến thành phần luma, cIdx bằng 1 tham chiếu đến thành phần Cb, và cIdx bằng 2 tham chiếu đến thành phần Cr), lfnst_idx là chỉ số LFNST cho CU, và nTbW và nTbH là chiều rộng và chiều cao của TB)

–Nếu (1) treeType bằng SINGLE_TREE, (2) lfnst_idx không bằng 0, (3) transform_skip_flag[xTbY][yTbY][cIdx] bằng 0, (4) cIdx bằng 0 và (5) cả nTbW và nTbH lớn hơn hoặc bằng 4, thì applyLfnstFlag được thiết lập bằng 1. (Bất kỳ tập con trong số (1) đến (5) có thể được sử dụng theo điều kiện này)

–Nếu không thì, nếu (1) treeType không bằng SINGLE_TREE, (2) lfnst_idx không bằng 0, (3) transform_skip_flag[xTbY][yTbY][cIdx] bằng 0 và (4) both nTbW và nTbH lớn hơn hoặc bằng 4, thì applyLfnstFlag được thiết lập bằng 1. (Bất kỳ tập con trong số (1) đến (4) có thể được sử dụng theo điều kiện này)

–Nếu không thì, applyLfnstFlag được thiết lập bằng 0.

Ở ví dụ khác, LFNST chroma được huỷ bỏ. Biến applyLfnstFlag được suy dẫn như sau:

–Nếu (1) lfnst_idx không bằng 0, (2) transform_skip_flag[xTbY][yTbY][cIdx] bằng 0, (3) cIdx bằng 0 và (4) cả nTbW và nTbH lớn hơn hoặc bằng 4, thì applyLfnstFlag được thiết lập bằng 1. (Bất kỳ tập con trong số (1) đến (4) có thể được sử dụng theo điều kiện này)

–Nếu không thì, applyLfnstFlag được thiết lập bằng 0.

Theo phương án phụ khác, applyLfnstFlag có thể được sử dụng ở một hoặc nhiều phân mục liên quan đến LFNST. Ví dụ, chỉ số LFNST được tham chiếu ở phân mục tương ứng ở tiêu chuẩn dự thảo.

8.7.4 Phép biến đổi cho các hệ số biến đổi được phóng tỷ lệ

….Khi applyLfnstFlag bằng 1 //lfnst_idx không bằng 0 và transform_skip_flag[xTbY][yTbY][cIdx] bằng 0 và cả nTbW và nTbH lớn hơn hoặc bằng 4//, điều sau được áp dụng:…

Ở các câu lệnh được sửa đổi ở trên dựa vào tiêu chuẩn dự thảo, các ký tự chữ nằm giữa cặp dấu "//" biểu thị các ký tự chữ được xoá.

8.7.3 Phép phóng tỷ lệ cho các hệ số biến đổi

…Để suy dẫn các hệ số biến đổi được phóng tỷ lệ d[x][y] với x = 0..nTbW − 1, y = 0..nTbH − 1, điều sau được áp dụng:

–Hệ số phóng tỷ lệ trung gian m[x][y] được suy dẫn như sau:

–Nếu một hoặc nhiều điều kiện sau là đúng (true), m[x][y] được thiết lập bằng 16:

–sps_scaling_list_enabled_flag bằng 0.

–pic_scaling_list_present_flag bằng 0.

–transform_skip_flag[xTbY][yTbY][cIdx] bằng 1.

–scaling_matrix_for_lfnst_disabled_flag bằng 1 và applyLfnstFlag bằng 1 //lfnst_idx[xTbY][yTbY] không bằng 0//….

Ở các câu lệnh được sửa đổi ở trên dựa vào tiêu chuẩn dự thảo, các ký tự chữ nằm giữa cặp dấu "//" biểu thị các ký tự chữ được xoá.

Theo phương án khác, khi LFNST chroma được huỷ bỏ ở một số trường hợp, LfnstDcOnly, mà được khởi tạo bằng 1 trước khi phân tích từng TB trong một CU và được chuyển thành 0 nếu bất kỳ TB trong CU đó có các hệ số có nghĩa (hoặc hệ số có nghĩa cuối) được nằm tại vị trí lớn hơn vị trí DC, không được cập nhật ở các TB không LFNST. Ví dụ, LFNST chroma được huỷ bỏ đối với cây đơn. Thì khi đó, các TB không LFNST chứa các TB chroma đối với cây đơn. Ví dụ về những thay đổi tương ứng ở bảng cú pháp được thể hiện ở Bảng 13.

Bảng 13. Bảng cú pháp được lấy làm ví dụ về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

Residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {…	**Bộ mô tả**
if(lastSubBlock = = 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastSPos > 0 && ((cIdx == 0) || (treeType != SINGLE_TREE)))	
LfnstDcOnly = 0	
if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2)	| (lastSPos > 7 && (log2TbWidth = = 2 | | log2TbWidth = = 3) && log2TbWidth = = log2TbHeight))	
LfnstZeroOutSigCoeffFlag = 0	
…}	

Theo ví dụ khác, LFNST chroma được huỷ bỏ và các TB không LFNST chứa các TB chroma. Ví dụ về những thay đổi tương ứng ở bảng cú pháp được thể hiện ở Bảng 14.

Bảng 14. Bảng cú pháp được lấy làm ví dụ về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {…	**Bộ mô tả**
if(lastSubBlock = = 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastScanPos > 0 && (cIdx = 0))	
LfnstDcOnly = 0	

if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) | | (lastScanPos > 7 && (log2TbWidth = = 2 | | log2TbWidth = = 3) && log2TbWidth = = log2TbHeight))	
LfnstZeroOutSigCoeffFlag = 0	
…}	

Theo phương án khác, khi LFNST chroma được huỷ bỏ ở một số trường hợp, LfnstZeroOutSigCoeffFlag, mà được khởi tạo bằng 1 trước khi phân tích từng TB trong một CU và được chuyển thành 0 nếu bất kỳ TB trong CU đó có các hệ số có nghĩa (hoặc hệ số có nghĩa cuối) được nằm tại vùng LFNST khác không, không được cập nhật ở các TB không LFNST. Ví dụ, LFNST chroma được huỷ bỏ đối với cây đơn. Thì khi đó, các TB không LFNST chứa các TB chroma cho cây đơn. Ví dụ về những thay đổi tương ứng ở bảng cú pháp được thể hiện ở Bảng 15.

Bảng 15. Bảng cú pháp được lấy làm ví dụ về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {…	**Bộ mô tả**
if(lastSubBlock = = 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastSPos > 0)	
LfnstDcOnly = 0	
if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) | | (lastScanPos > 7 && (log2TbWidth = = 2 | | log2TbWidth = = 3) && log2TbWidth = = log2TbHeight) && ((cIdx == 0) || (treeType != SINGLE_TREE)))	
LfnstZeroOutSigCoeffFlag = 0	
…}	

Theo ví dụ khác, LFNST chroma được huỷ bỏ và các TB không LFNST chứa các TB chroma. Ví dụ về những thay đổi tương ứng ở bảng cú pháp được thể hiện ở Bảng 16.

Bảng 16. Bảng cú pháp được lấy làm ví dụ về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

Residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {…	**Bộ mô tả**
if(lastSubBlock = = 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastSPos > 0)	
LfnstDcOnly = 0	
if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) || (lastScanPos > 7 && (log2TbWidth = = 2 || log2TbWidth = = 3) && log2TbWidth = = log2TbHeight) &&(cIdx = 0))	
LfnstZeroOutSigCoeffFlag = 0	
…}	

Dựa vào Bảng 11, là bảng dùng để chỉ kiểm tra transform_skip_flag luma khi lập mã các TU luma (ví dụ không dùng cho DUAL_TREE_CHROMA) và chỉ kiểm tra transform_skip_flag chroma khi lập mã các TU chroma (ví dụ không dùng cho DUAL_TREE_LUMA), LFNST chroma được huỷ bỏ theo một số trường hợp. Ví dụ, đối với cây đơn, LFNST chroma được huỷ bỏ. Ví dụ của bảng cú pháp được đề xuất được thể hiện ở Bảng 17.

Bảng 17. Bảng cú pháp được lấy làm ví dụ về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == = 1 && CuPredMode[chType][x0][y0] = = MODE_INTRA && (treeType == DUAL_TREE_CHROMA? (transform_skip_flag[x0][y0][1] = = 0 && transform_skip_flag[x0][y0][2] = = 0) : (transform_skip_flag[x0][y0][0] = = 0)) && (treeType != DUAL_TREE_CHROMA | | !intra_mip_flag[x0][y0] | | Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) {	
if((IntraSubPartitionsSplitType != ISP_NO_SPLIT | | LfnstDcOnly = = 0) && LfnstZeroOutSigCoeffFlag = = 1)	
lfnst_idx	ae(v)
}	

Ví dụ khác của bảng cú pháp được đề xuất dựa vào Bảng 12 còn được thể hiện ở Bảng 18.

Bảng 18. Bảng cú pháp được lấy làm ví dụ về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 &&CuPredMode[chType][x0][y0] = = MODE_INTRA && (treeType == DUAL_TREE_CHROMA | | transform_skip_flag[x0][y0][0] = = 0) && (treeType != DUAL_TREE_CHROMA | | (transform_skip_flag[x0][y0][1] = = 0 && transform_skip_flag[x0][y0][2] = = 0)) && (treeType = = DUAL_TREE_CHROMA | | !intra_mip_flag[x0][y0] | | Min(lfnstWidth, lfnstHeight) >= 16) &&	

Max(cbWidth, cbHeight) <= MaxTbSizeY) {	
if((IntraSubPartitionsSplitType != ISP_NO_SPLIT || LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag = = 1)	
lfnst_idx	ae(v)
}	

Theo ví dụ khác, LFNST chroma được huỷ bỏ. Bảng cú pháp được đề xuất được thể hiện như sau. Ví dụ của bảng cú pháp được đề xuất được thể hiện ở Bảng 19.

Bảng 19. Bảng cú pháp được lấy làm ví dụ về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag = = 1 &&
CuPredMode[chType][x0][y0] = = MODE_INTRA &&
(treeType != DUAL_TREE_CHROMA &&
(transform_skip_flag[x0][y0][0] = = 0)) &&
(treeType != DUAL_TREE_CHROMA || !intra_mip_flag[x0][y0] ||
Min(lfnstWidth, lfnstHeight) >= 16) &&
Max(cbWidth, cbHeight) <= MaxTbSizeY) {	

if((IntraSubPartitionsSplitType != ISP_NO_SPLIT || LfnstDcOnly = = 0) && LfnstZeroOutSigCoeffFlag = = 1)	
lfnst_idx	ae(v)
}	

Theo VVC dự thảo 7, các chỉ số LFNST và MTS được lập mã ở phần cuối của CU, điều này gây ra các vấn đề về độ trễ và bộ nhớ đệm mà bộ giải mã cần để nhớ đệm tất cả các hệ số của toàn bộ ba thành phần màu trước khi nhận các chỉ số LFNST và MTS. Để giảm các vấn đề về độ trễ và bộ nhớ đệm, đã có đề xuất việc gửi cờ kích hoạt hoặc chỉ số LFNST và/hoặc MTS (ví dụ 0 là huỷ bỏ, 1 và 2 là kích hoạt sử dụng. 1 và 2 nghĩa là sử dụng phép biến đổi chính khác nhau hoặc ma trận LFNST khác nhau) ở phần cuối

của TB đầu tiên trong CU hoặc ở phần cuối của một hoặc nhiều TB của thành phần màu thứ nhất, hoặc ở phần cuối của TB khác không thứ nhất (và TB không bỏ qua biến đổi) trong CU hoặc ở phần cuối của một hoặc nhiều TB của thành phần màu thứ nhất (và TB không bỏ qua biến đổi).

Theo phương án khác, đề xuất việc gửi cờ kích hoạt hoặc chỉ số LFNST và/hoặc MTS ở phần cuối của TB khác không thứ nhất (và TB không bỏ qua biến đổi) trong CU hoặc ở phần cuối của một hoặc nhiều TB của thành phần màu thứ nhất (và TB không bỏ qua biến đổi). Theo một ví dụ, nó có thể được áp dụng chỉ cho cây đơn. Ở cây đơn, các chỉ số LFNST và MTS được báo hiệu/được phân tích sau TB luma (hoặc trước các TB chroma). Nếu cây đơn và ISP được áp dụng, các chỉ số LFNST và MTS được báo hiệu/được phân tích sau khi TB luma cuối (hoặc trước các TB chroma). Ví dụ, subTuIndex có thể được sử dụng. Khi subTuIndex bằng NumIntraSubPartitions – 1, TB hiện thời là TB luma, và dạng cây hiện thời là cây đơn, các chỉ số LFNST và MTS được báo hiệu/được phân tích (nếu một hoặc nhiều điều kiện được thoả mãn).

Theo phương án khác, đề xuất việc gửi cờ kích hoạt hoặc chỉ số LFNST và/hoặc MTS ở phần cuối của một hoặc nhiều TB luma trong CU (hoặc trước các TB chroma) ở trường hợp cây đơn; trong khi ở cây lưỡng luma, chỉ số LFNST và/hoặc MTS được báo hiệu/được phân tích ở phần cuối của một hoặc nhiều TB luma trong CU (hoặc ở phần cuối của CU); trong khi ở cây lưỡng chroma, chỉ số LFNST và/hoặc MTS được báo hiệu/được phân tích sau khi ở phần cuối của các TB Cr trong CU (hoặc ở phần cuối của CU). Nếu cây đơn và ISP được áp dụng, các chỉ số LFNST và MTS được báo hiệu/được phân tích sau khi TB luma cuối (hoặc trước các TB chroma). Ví dụ, subTuIndex có thể được sử dụng. Khi subTuIndex bằng NumIntraSubPartitions – 1, TB hiện thời là TB luma và dạng cây hiện thời là cây đơn, các chỉ số LFNST và MTS được báo hiệu/được phân tích (nếu một hoặc nhiều điều kiện được thoả mãn).

Theo phương án khác, cờ kích hoạt hoặc chỉ số LFNST và/hoặc MTS được báo hiệu/được phân tích ở TB đầu tiên (ví dụ ở phần cuối của TB đầu tiên) khi cơ chế ISP được sử dụng. Phương pháp được đề xuất có thể chỉ được áp dụng cho cây đơn (ví dụ vẫn báo hiệu/phân tích chỉ số MTS/LFNST ở phần cuối của CU trong cây lưỡng luma hoặc cây lưỡng chroma).

Ở phương pháp đã nêu ở trên, chỉ số MTS có thể được báo hiệu/được phân tích

sau khi chỉ số LFNST. Nếu LFNST được sử dụng (ví dụ chỉ số LFNST không bằng 0), chỉ số MTS được xem là bằng 0. Theo phương án khác, LFNST có thể được báo hiệu/được phân tích sau khi báo hiệu/phân tích chỉ số MTS. Nếu MTS được sử dụng (ví dụ chỉ số MTS không bằng 0), chỉ số LFNST được xem là bằng 0.

Bất kỳ trong số các phương pháp được đề xuất ở trên có thể được kết hợp.

Bất kỳ biến thể kết hợp của các phương pháp nêu trên có thể được quyết định ẩn bằng chiều rộng khối hoặc chiều cao khối hoặc diện tích khối, hoặc được quyết định hiện bằng cờ được báo hiệu/được phân tích tại phân mức CU, CTU, lát ảnh (slice), khối ảnh (tile), nhóm khối ảnh, SPS, PPS, hoặc phân mức khung hình. "Khối" ở sáng chế này có thể có nghĩa là TU/TB/CU/CB/PU/PB.

Bất kỳ trong số các phương pháp được đề xuất ở trên có thể được thực hiện ở các bộ mã hoá và/hoặc các bộ giải mã. Ví dụ, bất kỳ trong số các phương pháp được đề xuất ở trên có thể được thực hiện ở môđun lập mã liên ảnh/nội ảnh/biến đổi của bộ mã hoá, môđun bù chuyển động, môđun suy dẫn ứng viên merge của bộ giải mã. Ngoài ra, bất kỳ trong số các phương pháp được đề xuất ở trên có thể được thực hiện dưới dạng mạch được liên kết với môđun lập mã liên ảnh/nội ảnh/biến đổi của bộ mã hoá và/hoặc môđun bù chuyển động, môđun suy dẫn ứng viên merge của bộ giải mã.

Fig. 2 minh hoạ lưu đồ của hệ thống mã hoá/giải mã được lấy làm ví dụ mà tích hợp báo hiệu/phân tích LFNST được ràng buộc theo phương án của sáng chế. Các bước được thể hiện trên lưu đồ này có thể được thực hiện dưới dạng các mã chương trình có thể chạy bằng một hoặc nhiều bộ xử lý (ví dụ, một hoặc nhiều CPU) tại bộ mã hoá. Các bước được thể hiện trên lưu đồ này cũng có thể được thực hiện dưới dạng phần cứng chẳng hạn một hoặc nhiều mạch hoặc bộ xử lý điện tử được sắp xếp để thực hiện các bước trên lưu đồ. Theo phương pháp này, dữ liệu đầu vào liên quan đến đơn vị lập mã hiện thời (CU) ở khung hình hiện thời được nhận ở bước 210, trong đó CU được phân chia thành một hoặc nhiều khối biến đổi (nhiều TB) và dữ liệu đầu vào tương ứng với dữ liệu dư tại bộ mã hoá viđeo và dữ liệu đầu vào tương ứng với dữ liệu được lập mã của CU hiện thời tại bộ giải mã viđeo. Cú pháp được xác định tại bộ mã hoá hoặc tại bộ giải mã ở bước 220, trong đó việc xác định cú pháp vừa nêu được thực hiện bằng cách báo hiệu cú pháp tại bộ mã hoá hoặc bằng cách phân tích cú pháp tại bộ giải mã nếu một hoặc nhiều điều kiện được thoả mãn. Cú pháp biểu thị liệu cơ chế LFNST được áp dụng

cho CU hiện thời và/hoặc loại nhân LFNSF được áp dụng khi cơ chế LFNST được áp dụng, và các điều kiện vừa nêu bao gồm điều kiện mong muốn tương ứng với tất cả TB mong muốn ở bộ TB mong muốn có chỉ báo cơ chế TS là false, và bộ TB mong muốn được chọn từ các TB ở CU hiện thời. CU hiện thời được mã hoá tại bộ mã hoá hoặc được giải mã tại bộ giải mã theo cơ chế LFNST như được biểu diễn bằng cú pháp ở bước 230.

Lưu đồ được thể hiện là nhằm minh hoạ ví dụ về hệ thống mã hoá/giải mã theo sáng chế. Người có hiểu biết trung bình trong lĩnh vực liên quan 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 trệch khỏi nội dung kỹ thuật 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 hiểu biết trung bình trong lĩnh vực liên quan có thể thực hiện sáng chế bằng cách thay thế cú pháp và các ngôn ngữ vừa nêu bằng cú pháp và các ngôn ngữ tương đương mà không trệch khỏi nội dung kỹ thuật của sáng chế.

Phần mô tả vừa nêu được đưa ra nhằm cho phép người có hiểu biết trung bình trong lĩnh vực này thực hiện sáng chế như được trình bày trong phần mô tả cụ thể và yêu cầu bảo hộ của nó. Nhiều sửa đổi khác nhau cho các phương án đã được mô tả sẽ là điều hiển nhiên đối với những người có hiểu biết trung bình trong lĩnh vực này, và các 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 có ý định bị giới hạn ở các phương án cụ thể đã được thể hiện và được mô tả, mà được ý định là tuân theo phạm vị rộng nhất phù hợp với những nguyên lý và dấu hiệu mới được bộc lộ ở bản mô tả này. Ở phần mô tả chi tiết sáng chế ở trên, nhiều phần chi tiết cụ thể khác nhau được minh họa nhằm giúp hiểu toàn bộ sáng chế. Tuy nhiên, sẽ được hiểu bởi những người có hiểu biết trung bình trong lĩnh vực này rằng 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 lập trình phần cứng, phần mềm khác nhau hoặc kết hợp cả phần cứng và phần mềm. Ví dụ, phương án của sáng chế có thể là một hoặc nhiều vi mạch điện tử được tích hợp vào chíp nén viđeo hoặc mã chương trình được tích hợp vào phần mềm nén viđeo để thực hiện quá trình xử lý đã được mô tả ở bản mô tả này. Phương án của sáng chế có thể còn là mã chương trình được xử lý trên bộ xử lý tín hiệu số DSP (Digital Signal Processor -

DSP) để thực hiện quá trình xử lý đã được mô tả ở bản mô tả này. Sáng chế có thể còn liên quan đến số lượng chức năng 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 mã phần mềm hoặc mã vi chương trình có thể đọc được bằng máy mà định ra các phương pháp thực hiện của sáng chế. Mã phần mềm hoặc mã vi chương trình có thể được phát triển theo các ngôn ngữ lập trình khác và các định dạng hoặc cách thức khác. Mã phần mềm có thể còn được tương thích với nhiều nền tảng khác nhau. Tuy nhiên, các định dạng, cách thức và ngôn ngữ mã khác nhau của các mã chương trình và các phương thức cấu hình mã khác để thực hiện những nhiệm vụ theo sáng chế sẽ không trệch khỏi nội dung và phạm vi của sáng chế. Phương án của sáng chế như được mô tả ở trên có thể được thực hiện ở bộ mã hoá viđeo và bộ giải mã viđeo. Các thành phần của bộ mã hoá viđeo và bộ giải mã viđeo có thể được xử lý bằng các phần cứng, một hoặc nhiều bộ xử lý được cấu hình để xử lý các lệnh chương trình được lưu ở bộ nhớ, hoặc kết hợp giữa phần cứng và bộ xử lý. Ví dụ, bộ xử lý xử lý các lệnh chương trình để điều khiển việc nhận dữ liệu đầu vào liên quan đến chuỗi viđeo chứa khối hiện thời ở khung hình hiện thời. Bộ xử lý được trang bị một hoặc nhiều nhân xử lý. Ở một số ví dụ, bộ xử lý xử lý các lệnh chương trình để thực hiện các chức năng ở một số bộ phận bộ mã hoá và bộ giải mã, và bộ nhớ được kết nối điện với bộ xử lý được sử dụng để lưu các lệnh chương trình, thông tin tương ứng với các ảnh được khôi phục của các khối, và/hoặc dữ liệu trung gian trong quá trình mã hoá hoặc giải mã. Bộ nhớ theo một số phương án bao gồm phương tiện có thể đọc bằng máy tính không chuyển tiếp, chẳng hạn bộ nhớ bán dẫn hoặc bộ nhớ trạng thái rắn, bộ nhớ truy cập ngẫu nhiên (RAM), bộ nhớ chỉ đọc (ROM), ổ cứng, ổ quang, hoặc phương tiện lưu thích hợp khác. Bộ nhớ vừa nêu có thể còn là kết hợp giữa hai hoặc nhiều hơn hai trong số các phương tiện có thể đọc bằng máy tính không chuyển tiếp vừa nêu ở trên.

Sáng chế có thể được thực hiện theo các dạng cụ thể khác mà không lệch khỏi nội dung và các đặc tính cơ bản của nó. Các ví dụ đã mô tả được xem xét theo toàn bộ các khía cạnh chỉ là nhằm minh họa chứ không nhằm giới hạn. Vì vậy, phạm vi của sáng chế được thể hiện theo các điểm yêu cầu bảo hộ chứ không phải phần mô tả đã nêu ở trên. Mọi thay đổi mà nằm trong nội dung ý nghĩa và phạm vi nghĩa tương đương của các điểm yêu cầu bảo hộ đều được xem là thuộc phạm vi của sáng chế.

YÊU CẦU BẢO HỘ

1. Phương pháp mã hoá hoặc giải mã chuỗi viđeo, trong đó cơ chế biến đổi không phân tách tần số thấp (low-frequency non-separable transform: LFNST) và cơ chế bỏ qua biến đổi (transform skip: TS) được hỗ trợ, phương pháp này bao gồm:

nhận dữ liệu đầu vào liên quan đến đơn vị lập mã (coding unit: CU) hiện thời ở khung hình hiện thời, trong đó CU được phân chia thành một hoặc nhiều khối biến đổi (transform block: TB) và dữ liệu đầu vào tương ứng với dữ liệu dư tại bộ mã hoá viđeo và dữ liệu đầu vào tương ứng với dữ liệu được lập mã của CU hiện thời tại bộ giải mã viđeo;

xác định cú pháp tại bộ mã hoá hoặc tại bộ giải mã, trong đó việc xác định cú pháp vừa nêu được thực hiện bằng cách báo hiệu cú pháp đó tại bộ mã hoá hoặc bằng cách phân tích cú pháp tại bộ giải mã nếu một hoặc nhiều điều kiện được thoả mãn, trong đó cú pháp biểu thị liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc loại nhân LFNST được áp dụng khi cơ chế LFNST được áp dụng, và một hoặc nhiều điều kiện vừa nêu bao gồm điều kiện mong muốn tương ứng với tất cả TB mong muốn ở bộ TB mong muốn có chỉ báo cơ chế TS là sai (false), và trong đó bộ TB mong muốn được chọn từ một hoặc nhiều TB, trong đó bộ TB mong muốn tương ứng với một hoặc nhiều TB thứ nhất cho mỗi khối lập mã trong CU hiện thời; và

mã hoá CU hiện thời tại bộ mã hoá hoặc giải mã CU hiện thời tại bộ giải mã theo cơ chế LFNST như được chỉ ra bằng cú pháp.

2. Phương pháp theo điểm 1, trong đó ở cây phân chia luma, CU hiện thời tương ứng với khối lập mã luma, và bộ TB mong muốn tương ứng với một hoặc nhiều TB luma.

3. Phương pháp theo điểm 1, trong đó ở cây phân chia chroma, CU hiện thời tương ứng với một hoặc nhiều khối lập mã chroma, và bộ TB mong muốn tương ứng với một hoặc nhiều TB chroma.

4. Phương pháp theo điểm 1, trong đó ở cây phân chia đơn, CU hiện thời tương ứng với một khối lập mã luma và một hoặc nhiều khối lập mã chroma, và bộ TB mong muốn tương ứng với một hoặc nhiều TB luma và một hoặc nhiều TB chroma.

5. Phương pháp theo điểm 1, trong đó CU hiện thời tương ứng với một khối lập mã luma hoặc một hoặc nhiều khối lập mã chroma, và bộ TB mong muốn tương ứng với một hoặc nhiều TB cho tất cả các thành phần ở CU hiện thời.

6.Phương pháp theo điểm 1, trong đó biến được sử dụng để ghi nhận liệu có báo hiệu hoặc phân tích cú pháp cho cơ chế LFNST hay không.

7. Phương pháp theo điểm 1, trong đó cú pháp được suy ra dưới dạng cơ chế LFNST không được áp dụng cho CU hiện thời khi chỉ báo cơ chế TS là đúng (true).

8. Phương pháp theo điểm 1, trong đó cơ chế LFNST không được sử dụng cho TB ở CU hiện thời nếu cú pháp được thiết kế dưới dạng chỉ số LFNST cho CU hiện thời bằng 0, và trong đó chỉ số LFNST có trị số lớn hơn 0 biểu thị một trong số các ma trận biến đổi tần số thấp được chọn.

9. Thiết bị mã hoá hoặc giải mã chuỗi viđeo, trong đó cơ chế biến đổi không phân tách tần số thấp (low-frequency non-separable transform: LFNST) và cơ chế bỏ qua biến đổi (transform skip: TS) được hỗ trợ, thiết bị này bao gồm một hoặc nhiều mạch hoặc bộ xử lý điện tử được sắp xếp để:

nhận dữ liệu đầu vào liên quan đến đơn vị lập mã (coding unit: CU) hiện thời ở khung hình hiện thời, trong đó CU được phân chia thành một hoặc nhiều khối biến đổi (transform block: TB) và dữ liệu đầu vào tương ứng với dữ liệu dư tại bộ mã hoá viđeo hoặc dữ liệu đầu vào tương ứng với dữ liệu được lập mã của CU hiện thời tại bộ giải mã viđeo;

xác định cú pháp tại bộ mã hoá hoặc tại bộ giải mã, trong đó việc xác định cú pháp vừa nêu được thực hiện bằng cách báo hiệu cú pháp tại bộ mã hoá hoặc bằng cách phân tích cú pháp tại bộ giải mã nếu một hoặc nhiều điều kiện được thoả mãn, trong đó cú pháp biểu thị liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc loại nhân LFNST được áp dụng khi cơ chế LFNST được áp dụng, và một hoặc nhiều điều kiện vừa nêu bao gồm điều kiện mong muốn tương ứng với tất cả TB mong muốn ở bộ TB mong muốn có chỉ báo cơ chế TS là sai (false), và trong đó bộ TB mong muốn được chọn từ một hoặc nhiều TB, trong đó bộ TB mong muốn tương ứng với một hoặc nhiều TB thứ nhất cho mỗi khối lập mã trong CU hiện thời; và

mã hoá CU hiện thời tại bộ mã hoá hoặc giải mã CU hiện thời tại bộ giải mã theo cơ chế LFNST như được biểu diễn bằng cú pháp.

10. Thiết bị theo điểm 9, trong đó ở cây phân chia luma, CU hiện thời tương ứng với khối lập mã luma, và bộ TB mong muốn tương ứng với một hoặc nhiều TB luma.

11. Thiết bị theo điểm 9, trong đó ở cây phân chia chroma, CU hiện thời tương ứng với một hoặc nhiều khối lập mã chroma, và bộ TB mong muốn tương ứng với một hoặc

nhiều TB chroma.

12. Thiết bị theo điểm 9, trong đó ở cây phân chia đơn, CU hiện thời tương ứng với một khối lập mã luma và một hoặc nhiều khối lập mã chroma, và bộ TB mong muốn tương ứng với một hoặc nhiều TB luma và một hoặc nhiều TB chroma.

13. Thiết bị theo điểm 9, trong đó CU hiện thời tương ứng với một khối lập mã luma hoặc một hoặc nhiều khối lập mã chroma, và bộ TB mong muốn tương ứng với một hoặc nhiều TB cho tất cả các thành phần ở CU hiện thời.

14.Thiết bị theo điểm 9, trong đó biến được sử dụng để ghi nhận liệu có báo hiệu hoặc phân tích cú pháp cho cơ chế LFNST hay không.

15. Thiết bị theo điểm 9, trong đó cú pháp vừa nêu được suy ra dưới dạng cơ chế LFNST không được áp dụng cho CU hiện thời khi chỉ báo cơ chế TS là đúng (true). Thiết bị theo điểm 9, trong đó cơ chế LFNST không được sử dụng cho TB ở CU hiện thời nếu cú pháp được thiết kế dưới dạng chỉ số LFNST cho CU hiện thời bằng 0, và trong đó chỉ số LFNST có trị số lớn hơn 0 biểu thị một trong số các ma trận biến đổi tần số thấp được chọn.

1/2

Fig. 1

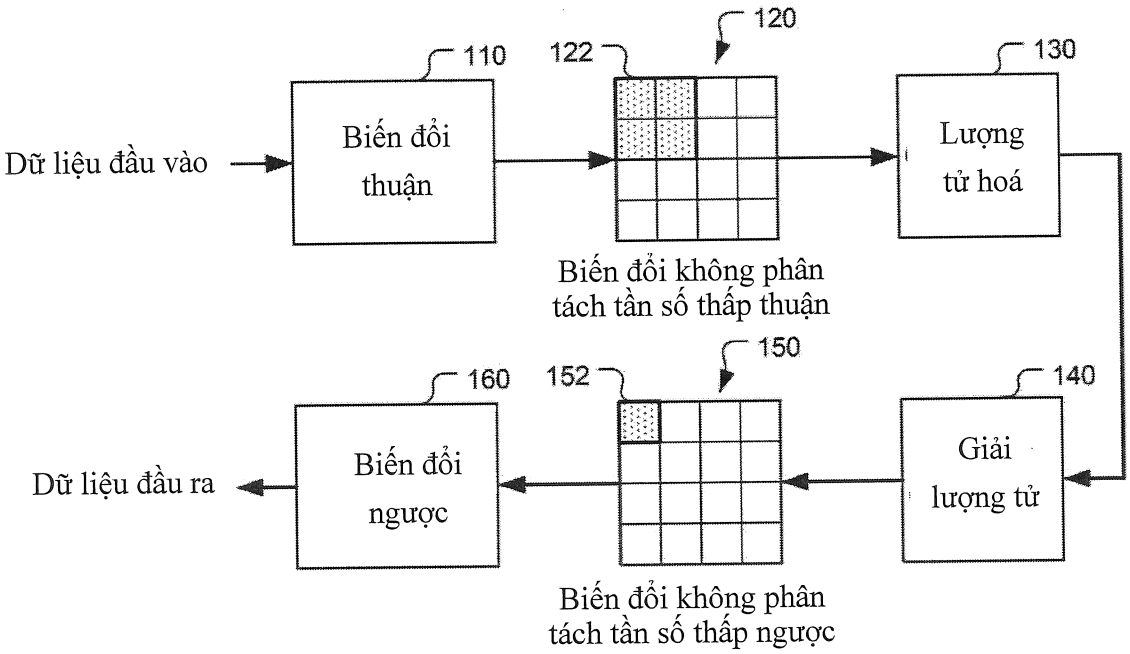
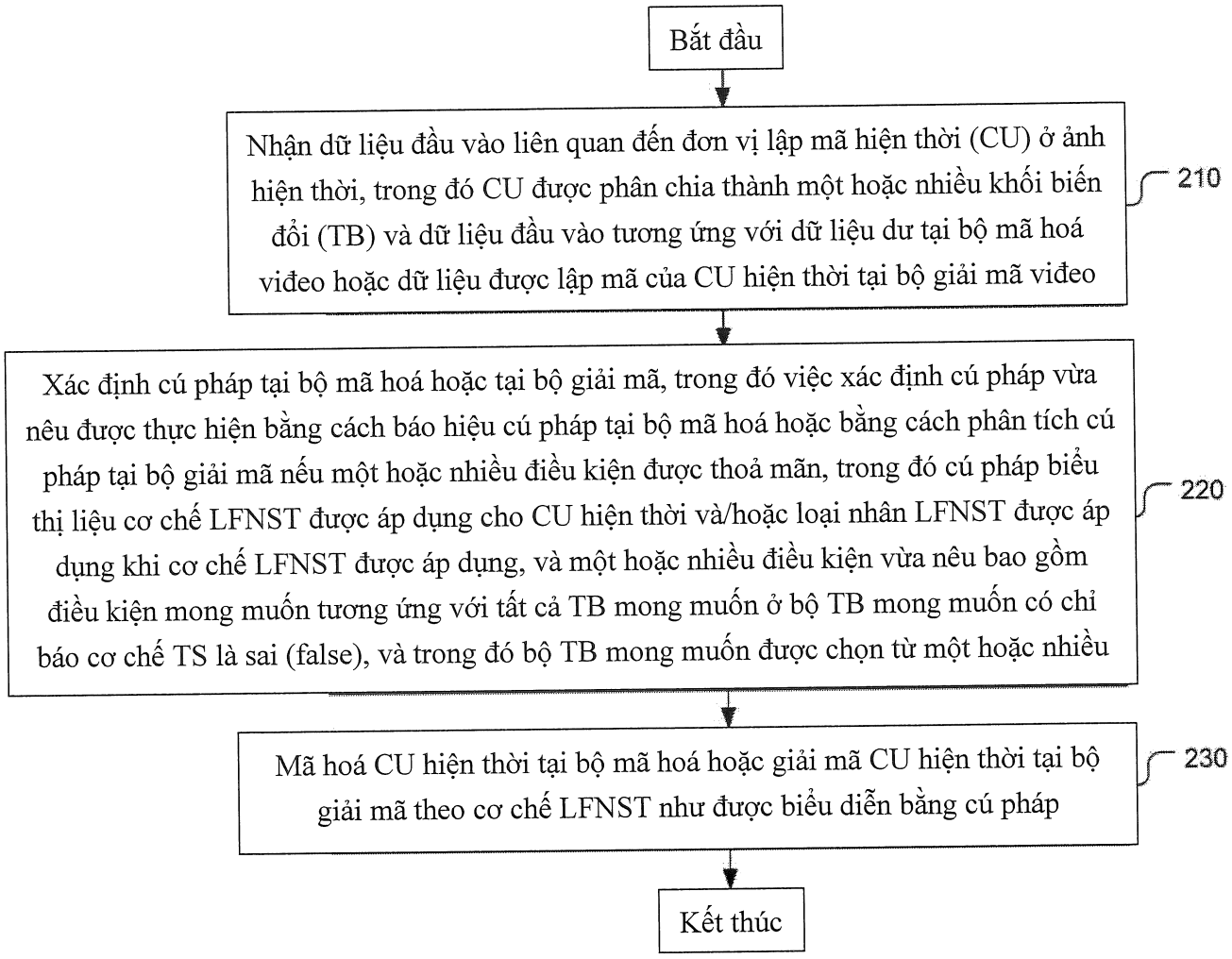

120
110
122
130
Dữ liệu đầu vào
Biến đổi thuận
Lượng tử hoá
Biến đổi không phân tách tần số thấp thuận
150
160
152
140
Dữ liệu đầu ra
Biến đổi ngược
Giải lượng tử
Biến đổi không phân tách tần số thấp ngược

Fig. 2


Bắt đầu
Nhận dữ liệu đầu vào liên quan đến đơn vị lập mã hiện thời (CU) ở ảnh hiện thời, trong đó CU được phân chia thành một hoặc nhiều khối biến đổi (TB) và dữ liệu đầu vào tương ứng với dữ liệu dư tại bộ mã hoá viđeo hoặc dữ liệu được lập mã của CU hiện thời tại bộ giải mã viđeo
210
Xác định cú pháp tại bộ mã hoá hoặc tại bộ giải mã, trong đó việc xác định cú pháp vừa nêu được thực hiện bằng cách báo hiệu cú pháp tại bộ mã hoá hoặc bằng cách phân tích cú pháp tại bộ giải mã nếu một hoặc nhiều điều kiện được thoả mãn, trong đó cú pháp biểu thị liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc loại nhân LFNST được áp dụng khi cơ chế LFNST được áp dụng, và một hoặc nhiều điều kiện vừa nêu bao gồm điều kiện mong muốn tương ứng với tất cả TB mong muốn ở bộ TB mong muốn có chỉ báo cơ chế TS là sai (false), và trong đó bộ TB mong muốn được chọn từ một hoặc nhiều
220
Mã hoá CU hiện thời tại bộ mã hoá hoặc giải mã CU hiện thời tại bộ giải mã theo cơ chế LFNST như được biểu diễn bằng cú pháp
230
Kết thúc