



(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)



1-0049121

(51)^{2020.01} H04N 19/176

(13) B

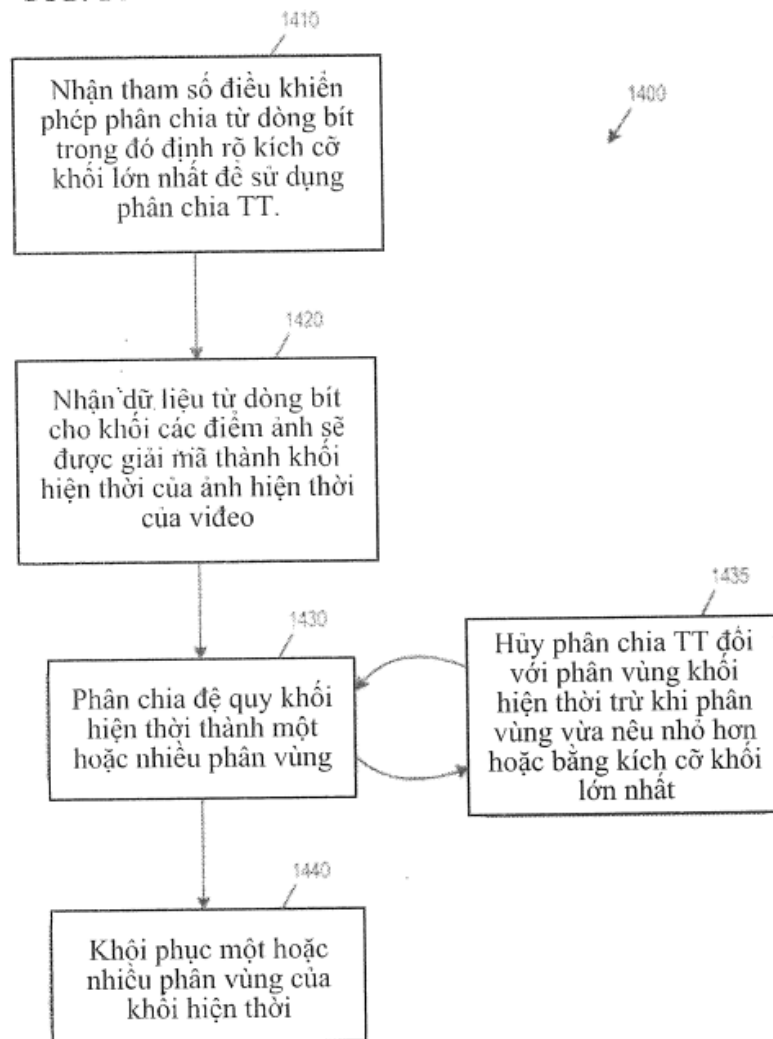
(21) 1-2022-02771 (22) 14/10/2020
(86) PCT/CN2020/120799 14/10/2020 (87) WO2021/073521 22/04/2021
(30) 62/914,577 14/10/2019 US; 62/915,049 15/10/2019 US; 62/916,891 18/10/2019 US;
62/930,084 04/11/2019 US; 62/956,359 02/01/2020 US; 17/068,817 12/10/2020 US
(45) 25/07/2025 448 (43) 25/07/2022 412A
(73) HFI INNOVATION INC. (CN)
3F.-7, No.5, Taiyuan 1st St., Zhubei City, Hsinchu County 302, Taiwan
(72) HSIANG, Shih-Ta (TW); LAI, Chen-Yen (TW); CHEN, Ching-Yeh (TW).
(74) Công ty TNHH Trường Xuân (AGELESS CO.,LTD.)

(54) PHƯƠNG PHÁP GIẢI MÃ VIDEO, PHƯƠNG PHÁP MÃ HÓA VIDEO VÀ
THIẾT BỊ ĐIỆN TỬ

(21) 1-2022-02771

(57) Sáng chế đề cập đến phương pháp giải mã video, phương pháp mã hóa video và thiết bị điện tử. Hệ thống này nhận tham số điều khiển phép phân chia từ dòng bit trong đó định rõ kích cỡ khối lớn nhất để sử dụng phân chia cây tam phân được ràng buộc bằng 64 hoặc nhỏ hơn. Hệ thống này nhận dữ liệu từ dòng bit cho khối các điểm ảnh sẽ được giải mã thành khối hiện thời của ảnh hiện thời của video. Hệ thống này phân chia đệ quy khối hiện thời thành một hoặc nhiều phân vùng, trong đó phân chia tam phân được hủy bỏ đối với phân vùng khối hiện thời trừ khi phân vùng vừa nêu nhỏ hơn hoặc bằng kích cỡ khối lớn nhất. Hệ thống này khôi phục một hoặc nhiều phân vùng khối hiện thời.

FIG. 14



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập chung đến phương pháp lập mã video và lập mã các ảnh. Cụ thể hơn là, sáng chế đề cập đến phương pháp giải mã video, phương pháp mã hóa video và thiết bị điện tử.

Tình trạng kỹ thuật của sáng chế

Trừ khi được chỉ rõ mang nghĩa khác ở bản mô tả này, nếu không thì những khía cạnh được mô tả ở phần tình trạng kỹ thuật của sáng chế này không phải là giải pháp kỹ thuật ưu tiên cho các điểm yêu cầu bảo hộ được liệt kê ở dưới đây cũng như không được coi là giải pháp kỹ thuật ưu tiên thuộc phần này.

Theo lập mã video hiệu suất cao HEVC (High Efficiency Video Coding - HEVC), ảnh được lập mã được phân chia thành các vùng khối vuông không chồng lấn được biểu diễn bằng các đơn vị cây lập mã CTU (Coding Tree Unit - CTU). Ảnh được lập mã có thể được biểu diễn bằng tập hợp các mảng (slice), mỗi chúng chứa một số nguyên các CTU. Các CTU riêng trong slice được xử lý theo lệnh quét mảng. Slice lưỡng dự đoán (Slice B) có thể được giải mã bằng dự đoán nội ảnh hoặc dự đoán liên ảnh bằng cách sử dụng tối đa hai vectơ chuyển động và các tham số tham chiếu để dự đoán các giá trị mẫu của từng khối. Slice nội ảnh (Slice I) được giải mã chỉ bằng dự đoán nội ảnh. Slice đơn dự đoán (Slice P) được giải mã bằng dự đoán nội ảnh hoặc dự đoán liên ảnh bằng cách sử dụng tối đa một vectơ chuyển động và tham số tham chiếu để dự đoán các giá trị mẫu của từng khối.

CTU có thể được phân chia tiếp thành nhiều đơn vị lập mã không chồng lấn (nhiều CU) để thích ứng với nhiều đặc tính cấu trúc và chuyển động cục bộ khác nhau. Cây lập mã được xây dựng theo cấu trúc phân chia cây tứ phân QT (QuadTree - QT) được sử dụng để biểu diễn các phân vùng CU được chia từ CTU. Xét CTU có kích cỡ $M \times M$ mẫu điểm ảnh (pixel), trong đó M là một trong các trị số 64, 32, và 16. CTU có thể hoặc được cấu tạo là một đơn vị đơn có cùng kích cỡ $M \times M$, hoặc được phân chia thành bốn đơn vị nhỏ hơn có cùng kích cỡ $M/2 \times M/2$. CTU tương ứng với nút gốc của cây lập mã và bốn đơn vị $M/2 \times M/2$ là các nút cây tứ phân con được phân chia từ nút gốc của cây lập mã. Mỗi một trong số bốn nút cây tứ phân vừa nêu có thể trở thành nút

cha đề được phân chia tiếp thành bốn nút con có kích cỡ được giảm một nửa theo từng chiều không gian. Phân chia cây tứ phân có thể được áp dụng đệ quy cho đến khi lá đạt được. Các nút lá của cây lập mã tương ứng với các CU. Các kích cỡ CU lớn nhất và nhỏ nhất có thể được xác định theo bộ tham số chuỗi SPS (Sequence Parameter Set - SPS).

Một hoặc nhiều đơn vị dự đoán (PU) có thể được xác định cho từng CU. Đơn vị dự đoán, cùng với cú pháp CU liên quan, đóng vai trò là đơn vị cơ bản để báo hiệu thông tin dự đoán. Quá trình dự đoán vừa nêu được áp dụng để dự đoán các trị số của các mẫu điểm ảnh liên quan bên trong PU. CU có thể được phân chia thành một, hai, hoặc bốn PU, phụ thuộc vào dạng PU được chọn.

CU có thể được phân chia tiếp bằng cách sử dụng cấu trúc cây tứ phân dư RQT (Residual QuadTree - RQT) để biểu diễn tín hiệu dư dự đoán liên quan. Các nút lá của RQT tương ứng với các đơn vị biến đổi TU (Transform Unit - TU). Đơn vị biến đổi được cấu tạo gồm khối biến đổi TB (Transform Block - TB) của các mẫu sáng (mẫu luma) có kích cỡ 8x8, 16x16, hoặc 32x32 hoặc bốn khối biến đổi của các mẫu màu (mẫu chroma) có kích cỡ 4x4, và hai khối biến đổi tương ứng của các mẫu màu (mẫu chroma) của ảnh có định dạng màu 4:2:0. Phép biến đổi số nguyên được áp dụng cho khối biến đổi và các trị số hệ số lượng tử được lập mã trong dòng bit. Các kích cỡ khối biến đổi lớn nhất và nhỏ nhất được xác định theo bộ tham số chuỗi.

Ở HEVC, các thuật ngữ khối cây lập mã (CTB), khối lập mã (CB), khối dự đoán (PB), và khối biến đổi (TB) được định nghĩa để xác định mảng mẫu 2-D của một thành phần màu từ CTU, CU, PU, và TU liên quan, một cách tương ứng. Do vậy, CTU bao gồm một CTB luma, hai CTB chroma, và các thành phần cú pháp liên quan trong ảnh màu không được lập mã bằng ba mặt phẳng màu riêng biệt. Cấu trúc phân chia cây lập mã được báo hiệu vừa nêu thường được áp dụng cho cả khối luma và khối chroma, song một số ngoại lệ được áp dụng khi những điều kiện ràng buộc kích cỡ nhất định được yêu cầu.

Bản chất kỹ thuật của sáng chế

Phần bản chất kỹ thuật dưới đây chỉ nhằm mục đích minh họa chứ không nhằm giới hạn sáng chế theo bất kỳ cách thức nào. Tức là, phần bản chất kỹ thuật dưới đây được đưa ra nhằm giới thiệu những ý tưởng, những điểm nổi bật, những lợi ích và những hiệu quả có lợi của những kỹ thuật mới và chưa tường minh được mô tả ở bản

mô tả này. Việc chọn một số chứ không phải toàn bộ các phương án được mô tả rõ hơn dưới đây ở phần mô tả chi tiết sáng chế. Do đó, phần bản chất kỹ thuật của sáng chế dưới đây không nhằm định rõ các dấu hiệu cơ bản của các đối tượng được bảo hộ, cũng như không nhằm xác định phạm vi của các đối tượng được bảo hộ.

Một số phương án của sáng chế đề xuất hệ thống video trong đó áp dụng các điều kiện ràng buộc đối với phép phân chia khối. Hệ thống video này nhận tham số điều khiển phép phân chia từ dòng bit trong đó định rõ kích cỡ khối lớn nhất để sử dụng phân chia TT được ràng buộc bằng 64 hoặc nhỏ hơn. Hệ thống video nhận dữ liệu từ dòng bit cho khối các điểm ảnh sẽ được giải mã thành khối hiện thời của ảnh hiện thời của video. Hệ thống video phân chia đệ quy khối hiện thời thành một hoặc nhiều phân vùng, trong đó phân chia TT được hủy bỏ đối với phân vùng khối hiện thời trừ khi phân vùng vừa nêu nhỏ hơn hoặc bằng kích cỡ khối lớn nhất. Hệ thống video khôi phục một hoặc nhiều phân vùng khối hiện thời.

Theo một số phương án, tham số điều khiển phép phân chia được báo hiệu ở tiêu đề slice của slice mà chứa khối hiện thời. Tham số điều khiển phép phân chia có thể được báo hiệu ở tiêu đề ảnh của ảnh hiện thời. Tham số điều khiển phép phân chia có thể được báo hiệu theo bộ tham số chuỗi (SPS) của chuỗi các ảnh video mà chứa ảnh hiện thời. Theo một số phương án, SPS hoặc tiêu đề ảnh thiết lập giá trị mặc định cho tham số điều khiển phép phân chia có thể được ghi đè bởi tiêu đề slice mà chứa khối hiện thời.

Theo một số phương án, tham số điều khiển phép phân chia xác định kích cỡ khối lớn nhất bằng cách tham chiếu kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân, ví dụ, dưới dạng sai lệch giữa giữa (i) lôgarit cơ số 2 của kích cỡ khối lớn nhất theo đó khối có thể được phân chia tiếp bằng phân chia cây tam phân và (ii) lôgarit cơ số 2 của kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân. Kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân được ràng buộc là trị số nhỏ hơn giữa 64 và kích cỡ khối cây lập mã. Kích cỡ khối lớn nhất để sử dụng phân chia TT có thể được ràng buộc tiếp sao cho nhỏ hơn kích cỡ khối cây lập mã.

Kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân có thể được xác định bằng cách tham chiếu kích cỡ khối lập mã nhỏ nhất, và kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân được xác định bằng cách sử dụng sai lệch giữa (i) lôgarit cơ số 2 của kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân và (ii) lôgarit cơ số 2 của kích cỡ

khối lập mã nhỏ nhất.

Theo một số phương án, kích cỡ khối lớn nhất được xác định dựa vào kích cỡ của đơn vị dữ liệu đường truyền ảo VPDU (VPDU - virtual pipeline data unit), và kích cỡ khối lớn nhất được định nghĩa dựa vào việc đảm bảo sao cho mỗi VPDU có thể được xử lý hoàn toàn bằng trạng thái đường truyền giống nhau.

Mô tả vắn tắt các hình vẽ

Các hình vẽ đính kèm được sử dụng nhằm giúp hiểu rõ hơn sáng chế, và được đưa vào và cấu thành nên một phần của sáng chế. Các hình vẽ minh họa các phương án thực hiện của sáng chế và, cùng với phần mô tả, đóng vai trò giải thích các nguyên lý của sáng chế. Có thể thấy là các hình vẽ không nhất thiết phải đúng về mặt tỷ lệ vì một số bộ phận có thể được thể hiện không đúng với tỷ lệ kích thước trong thực tế nhằm minh họa rõ ý tưởng của sáng chế.

FIG. 1 minh họa Cấu trúc cây lập mã dùng để biểu diễn việc phân chia CTU thành các CU. Các đường nét liền biểu thị các biên giới CU và các đường nét đứt biểu thị các biên giới TU;

FIG. 2 minh họa tám dạng phân chia dùng để chia CU thành các PU được định nghĩa trong HEVC;

FIG. 3 minh họa một số dạng phân chia trong đó phân chia CU thành hai phân vùng, bao gồm các dạng phân chia đối xứng và không đối xứng khác nhau mà có thể phân chia CU theo phương đứng hoặc ngang;

FIG. 4 minh họa ví dụ về phân chia khối và cây nhị phân tương ứng của nó;

FIG. 5 minh họa ví dụ về phân chia khối và QTBT tương ứng của nó;

FIG. 6 minh họa CU được phân chia thành các CU nhỏ hơn bằng cách sử dụng một trong số năm dạng phân chia;

FIG. 7 minh họa các cấu trúc phân chia tương thích và không tương thích của khối các điểm ảnh khi các điều kiện ràng buộc theo VPDU được áp dụng;

FIG. 8 minh họa về mặt khái niệm các tham số ràng buộc dùng để điều khiển cấu trúc phân chia của khối điểm ảnh;

FIG. 9 minh họa bộ mã hóa video được lấy làm ví dụ để báo hiệu các điều kiện ràng buộc đối với phân chia khối;

FIG. 10 minh họa những bộ phận của bộ mã hóa video mà thực thi các điều kiện ràng buộc phân chia khối;

FIG. 11 minh họa về mặt khái niệm quy trình báo hiệu các điều kiện ràng buộc phân chia khối;

FIG. 12 minh họa bộ giải mã video được lấy làm ví dụ mà áp dụng các điều kiện ràng buộc đối với phép phân chia khối;

FIG. 13 minh họa những bộ phận của bộ giải mã video mà thực thi các điều kiện ràng buộc phân chia khối;

FIG. 14 minh họa về mặt khái niệm quy trình áp dụng các điều kiện ràng buộc phân chia khối; và

FIG. 15 minh họa về mặt ý tưởng hệ thống điện tử qua đó một số phương án của sáng chế được thực hiện.

Mô tả chi tiết sáng chế

Ở phần mô tả chi tiết sáng chế dưới đây, nhiều phương án chi tiết cụ thể được trình bày dưới dạng các ví dụ nhằm giúp hiểu toàn diện các nội dung sáng chế liên quan. Bất kỳ mọi sửa đổi, mọi suy dẫn và/hoặc mọi mở rộng dựa trên nội dung sáng chế được mô tả ở bản mô tả này đều nằm trong phạm vi bảo hộ của sáng chế. Trong một số trường hợp, các phương pháp, các quy trình, các bộ phận, và/hoặc hệ vi mạch vốn đã biết rõ mà liên quan đến một hoặc nhiều phương án thực hiện được lấy làm ví dụ được bộc lộ ở bản mô tả này có thể được mô tả ở mức độ vừa phải chứ không mô tả chi tiết quá, để tránh làm rối các khía cạnh nội dung của sáng chế.

I. Các cấu trúc phân chia khối

FIG. 1 minh họa cấu trúc cây lập mã dùng để biểu diễn việc phân chia CTU thành các CU. Các đường nét liền biểu thị các biên giới CU và các đường nét đứt biểu thị các biên giới TU. FIG. 2 minh họa tám dạng phân chia dùng để chia CU thành các PU được định nghĩa trong HEVC.

a. Phân chia khối cây nhị phân BT (Binary Tree - BT)

Theo một số phương án, các cấu trúc phân chia khối cây nhị phân (BT) được sử dụng để phân chia khối. Theo cấu trúc phân chia cây nhị phân, khối có thể được phân chia thành đệ quy hai khối hoặc hai phân vùng nhỏ hơn.

FIG. 3 minh họa một số dạng phân chia trong đó phân chia CU thành hai phân vùng, bao gồm các dạng phân chia đối xứng và không đối xứng khác nhau mà có thể phân chia CU theo phương đứng hoặc ngang. Theo một số phương án, các dạng phân chia đơn giản và hiệu quả nhất là các phân chia ngang và đứng đối xứng, do đó, chỉ hai

dạng phân chia này được sử dụng cho cấu trúc phân chia cây nhị phân.

Đối với khối cụ thể có kích cỡ $M \times N$, thành phần cú pháp `bt_split_flag` có thể được báo hiệu để biểu thị liệu có phân chia khối thành hai khối nhỏ hơn hay không. Nếu cờ này là đúng (true), thì thành phần cú pháp khác `bt_split_mode` được báo hiệu để cho biết loại phân chia nào được sử dụng (tức là, phân chia ngang hay đứng). Nếu phân chia ngang được sử dụng, thì khối đó được phân chia thành hai khối nhỏ hơn có kích cỡ $M \times N/2$. Ngược lại, nếu phân chia đứng được sử dụng, thì khối đó được phân chia thành hai khối nhỏ hơn có kích cỡ $M/2 \times N$. Phép phân chia cây nhị phân có thể được áp dụng đệ quy cho đến khi kích cỡ khối hiện thời (chiều rộng hoặc chiều cao) đạt kích cỡ khối (chiều rộng hoặc chiều cao) nhỏ nhất mà có thể được định nghĩa trong bộ cú pháp cấp cao chẳng hạn SPS. Vì cây nhị phân có hai dạng phân chia (tức là, phân chia ngang và đứng), nên chiều rộng và chiều cao khối nhỏ nhất có thể đều được biểu thị. Phân chia không theo phương ngang bị ẩn khi phân chia vừa nêu làm cho chiều cao khối nhỏ hơn nhỏ hơn giá trị nhỏ nhất đã biểu thị. Phân chia không theo phương đứng bị ẩn khi phân chia vừa nêu làm cho chiều rộng khối nhỏ hơn giá trị nhỏ nhất đã định.

FIG. 4 minh họa ví dụ về phân chia khối (bên trái) và cây nhị phân tương ứng của nó (bên phải). Đối với mỗi nút phân chia (tức là, nút không lá) của cây nhị phân, một cờ được báo hiệu để cho biết loại phân chia nào (ngang hay đứng) được sử dụng, “0” cho phân chia ngang và “1” cho phân chia đứng.

b. Phân chia khối cây tứ phân + cây nhị phân QTBT (Quad Tree Binary Tree - QTBT)

Theo một số phương án, cấu trúc cây tứ phân cộng cây nhị phân (QTBT) được sử dụng để phân chia CU thành các CU nhỏ hơn. Mỗi CTU có thể bao gồm một CU hoặc được phân chia thành nhiều CU nhỏ hơn bằng cách sử dụng cấu trúc phân chia cây tứ phân trước tiên. Phân chia cây tứ phân có thể được áp dụng đệ quy cho các CU cho đến khi kích cỡ CU hiện thời đạt kích cỡ nút lá cây tứ phân nhỏ nhất. Nếu khối cây tứ phân hiện thời không lớn hơn kích cỡ nút gốc cây nhị phân lớn nhất, thì nó có thể được phân chia tiếp bằng cách sử dụng cấu trúc phân chia cây nhị phân. Phân chia cây nhị phân có thể được áp dụng đệ quy cho đến khi kích cỡ CU hiện thời (chiều rộng hoặc chiều cao) đạt kích cỡ (chiều rộng hoặc chiều cao) nút lá cây nhị phân nhỏ nhất hoặc chiều sâu cây nhị phân hiện thời đạt chiều sâu cây nhị phân lớn nhất. Có hai dạng phân chia, phân chia ngang đối xứng và phân chia đứng đối xứng. Các nút lá cây nhị phân

tương ứng với các CU. Đối với mỗi nút phân chia (tức là, nút không lá) của cây nhị phân, một cờ được báo hiệu để cho biết loại phân chia nào (ngang hay đứng) được sử dụng. “0” biểu thị phân chia ngang và “1” biểu thị phân chia đứng. Kích cỡ nút lá cây tứ phân nhỏ nhất, kích cỡ nút gốc cây nhị phân lớn nhất, chiều rộng và chiều cao nút lá cây nhị phân nhỏ nhất, và chiều sâu cây nhị phân lớn nhất có thể được biểu thị trong các bộ cú pháp cấp cao. FIG. 5 minh họa ví dụ về phân chia khối (bên trái) và QTBT tương ứng của nó (bên phải). Các đường nét liền biểu diễn phân chia cây tứ phân và các đường nét đứt biểu diễn phân chia cây nhị phân.

c. Phân chia khối cây đa phân MTT (Multi Type Tree - MTT)

Theo cấu trúc cây đa phân (MTT), các cây tam phân TT (Ternary Tree – TT) ở chính giữa theo phương ngang và đứng được bổ sung thêm vào cấu trúc QTBT để phân chia các CU. FIG. 6 minh họa CU được phân chia thành các CU nhỏ hơn bằng cách sử dụng một trong số năm dạng phân chia. Các dạng phân chia (d) và (e) được gọi là các dạng phân chia cây bậc ba hoặc cây tam phân và có thể được sử dụng để chia khối thành ba khối nhỏ hơn, tất cả chúng đều có kích cỡ giảm theo một kích thước không gian trong khi vẫn giữ nguyên kích cỡ theo kích thước không gian còn lại. Chiều sâu cây nhị phân của CU hiện thời `btDepthC` trong cấu trúc MTT được định nghĩa theo sau:

$$\text{btDepthC} = \log_2((\text{widthR} * \text{heightR}) / (\text{widthC} * \text{heightC})),$$

trong đó `widthC` và `heightC` tương ứng là chiều rộng và chiều cao của CU hiện thời, và `widthR` và `heightR` tương ứng là chiều rộng và chiều cao của nút gốc MTT tương ứng, và hàm $\log_2(x)$ là loga cơ số 2 của x .

Theo một số phương án, mỗi đơn vị cây lập mã (CTU) có thể được phân chia thành một hoặc nhiều CU kích cỡ nhỏ hơn bằng cây tứ phân được lồng với cây đa phân bằng cách sử dụng phân chia nhị phân và tam phân. Thành phần cú pháp `split_cu_flag` được báo hiệu trước để biểu thị liệu đơn vị lập mã hiện thời có được phân chia tiếp thành nhiều hơn một đơn vị lập mã hay không. Khi `split_cu_flag` bằng 0, CU hiện thời là nút lá và không được phân chia tiếp. Nếu không thì, CU hiện thời được phân chia tiếp, và thành phần cú pháp `split_qt_flag` được báo hiệu tiếp để biểu thị liệu đơn vị lập mã hiện thời có được phân chia bằng phân chia cây tứ phân hay không. Khi `split_qt_flag` bằng 1, CU hiện thời được phân chia tiếp bằng phân chia cây tứ phân thành bốn CU có kích cỡ theo phương ngang và đứng giảm một nửa. Nếu không thì,

các thành phần cú pháp `mtt_split_cu_vertical_flag` và `mtt_split_cu_binary_flag` được báo hiệu để cho biết phương (ngang hay đứng) và dạng (cây nhị phân hay cây tam phân) của phân chia MTT được chọn.

II. Các điều kiện ràng buộc đối với kích cỡ khối dùng để phân chia

Các đơn vị dữ liệu đường truyền ảo (các VPDU) được định nghĩa dưới dạng các đơn vị $M \times M$ -luma(L)/ $N \times N$ -chroma(C) trong ảnh. Ở các bộ giải mã phần cứng, các VPDU liên tiếp nhau được xử lý theo nhiều trạng thái truyền một cách đồng thời và các trạng thái khác nhau sẽ xử lý các VPDU khác nhau một cách đồng thời. Kích cỡ VPDU về cơ bản tỷ lệ với kích cỡ bộ đệm ở hầu hết các trạng thái truyền. Ở các bộ giải mã phần cứng HEVC, kích cỡ VPDU thường được thiết lập sao cho tương hợp với kích cỡ khối biến đổi (TB) lớn nhất, còn được gọi là kích cỡ khối cho phép lớn nhất, hoặc `MaxTbSizeY`.

Theo một số phương án, để giảm kích cỡ VPDU trong video được lập mã, phân chia CU được ràng buộc theo điều kiện sau: (1) Đối với mỗi VPDU chứa một hoặc nhiều CU, các CU được chứa hoàn toàn trong VPDU. (2) Đối với mỗi CU chứa một hoặc nhiều VPDU, các VPDU được chứa hoàn toàn trong CU. Thứ tự xử lý các CU sẽ không bỏ qua VPDU và sẽ tái xử lý nó sau. Điều này đảm bảo rằng mỗi VPDU có thể được xử lý hoàn toàn bằng trạng thái đường truyền giống nhau.

FIG. 7 minh họa các cấu trúc phân chia tương thích và không tương thích của khối các điểm ảnh khi các điều kiện ràng buộc theo VPDU được áp dụng. Hình vẽ này minh họa khối 700 (ví dụ, CTU) mà có thể được phân chia tiếp để lập mã. Kích cỡ của khối 700 là các mẫu luma 128×128 . Khối 700 được xử lý thành bốn VPDU từ 701 đến 704 (được phân chia bằng các đường nét đứt), mỗi VPDU tương ứng với vùng các mẫu luma 64×64 .

Hình vẽ này thể hiện một số mô hình được lấy làm ví dụ từ (a) đến (l) dùng để phân chia khối 700 thành các CU. Một số trong số các mô hình phân chia vừa nêu tương thích với các VPDU trong khi các mô hình còn lại là không tương thích. Cụ thể là, các mô hình phân chia (a), (b), (c), và (d) tạo ra các CU thì tương thích với các VPDU, vì đối với mỗi VPDU chứa một hoặc nhiều CU, các CU được chứa hoàn toàn trong VPDU; và đối với mỗi CU chứa một hoặc nhiều VPDU, các VPDU được chứa hoàn toàn trong CU. Điều này đảm bảo rằng không có VPDU nào cần được xử lý riêng thành một phần của một CU và được tái xử lý sau đó thành một phần của CU khác.

Các mô hình phân chia (e), (f), (g), (h), (i), (j), (k), (l) tạo ra các CU thì không tương thích với các VPDU, vì mỗi một trong số các mô hình vừa nêu yêu cầu ít nhất một VPDU được xử lý riêng thành một phần của một CU và được tái xử lý sau đó thành một phần của CU khác. Nói cách khác, có ít nhất một VPDU liên đới nhiều trạng thái truyền khác nhau.

Theo một số phương án, liệu phân chia TT được phép cho khối hiện thời hay không được xác định dựa vào các điều kiện ràng buộc VPDU, ví dụ, kích cỡ khối cho phép lớn nhất theo các giới hạn đường truyền. Liệu phân chia TT được phép hay không có thể được xác định dựa vào các thông số sau đây:

- phân chia tam phân mode ttSplit
- chiều rộng khối lập mã cbWidth trong các mẫu luma,
- chiều cao khối lập mã cbHeight trong các mẫu luma,
- tọa độ (x0, y0) của mẫu luma trên cùng-bên trái của khối lập mã đang xét so với mẫu luma trên cùng-bên trái của ảnh,
- chiều sâu cây đa phân mttDepth
- chiều sâu cây đa phân lớn nhất maxMttDepth,
- kích cỡ cây tam phân lớn nhất maxTtSize,
- biến treeType xác định liệu cây đơn (SINGLE_TREE) hay cây lưỡng được sử dụng để phân chia nút cây lập mã và, khi cây lưỡng được sử dụng, thì xác định liệu thành phần luma (DUAL_TREE_LUMA) hay thành phần chroma (DUAL_TREE_CHROMA) đang được xử lý hay không,
- biến modeType xác định liệu các cơ chế lập mã nội ảnh (MODE_INTRA), IBC (MODE_IBC), và liên ảnh có thể được sử dụng (MODE_TYPE_ALL), hay chỉ các cơ chế lập mã nội ảnh và IBC có thể được sử dụng (MODE_TYPE_INTRA), hay chỉ các cơ chế lập mã liên ảnh có thể được sử dụng (MODE_TYPE_INTER) cho các đơn vị lập mã bên trong nút cây lập mã.

Các tham số vừa nêu được sử dụng để xác định biến “allowTtSplit”. Biến cbSize được suy dẫn dựa vào ttSplit theo điều kiện sau:

	ttSplit == SPLIT_TT_VER	ttSplit == SPLIT_TT_HOR
cbSize	cbWidth	cbHeight

Biến allowTtsplit được suy dẫn theo điều kiện sau: (với kích cỡ VPDU được biểu

diễn bằng MaxTbSizeY)

– Nếu một hoặc nhiều điều kiện trong số những điều kiện sau là true, thì allowTtSplit được thiết lập là FALSE:

- cbSize nhỏ hơn hoặc bằng $2 * \text{MinTtSizeY}$
- cbWidth lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
- cbHeight lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
- mttDepth lớn hơn hoặc bằng maxMttDepth
- $x0 + \text{cbWidth}$ lớn hơn pic_width_in_luma_samples
- $y0 + \text{cbHeight}$ lớn hơn pic_height_in_luma_samples
- treeType bằng DUAL_TREE_CHROMA và $(\text{cbWidth} / \text{SubWidthC}) * (\text{cbHeight} / \text{SubHeightC})$ nhỏ hơn hoặc bằng 32
- treeType bằng DUAL_TREE_CHROMA và $(\text{cbWidth} / \text{SubWidthC})$ bằng 8 và ttSplit bằng SPLIT_TT_VER
- treeType bằng DUAL_TREE_CHROMA và modeType bằng MODE_TYPE_INTRA
- $\text{cbWidth} * \text{cbHeight}$ bằng 64 và modeType bằng MODE_TYPE_INTER
- Nếu không thì, allowTtSplit được thiết lập là TRUE.

Theo một số phương án, các tham số cú pháp cấp cao liên quan đến cấu trúc phân chia CU được xác định dựa vào những giới hạn về xử lý đường truyền, tức là, các điều kiện ràng buộc liên quan đến VPDU (ví dụ, kích cỡ khối cho phép lớn nhất) được sử dụng khi phân chia CU. Theo một số phương án, hệ thống lập mã video có thể bao gồm một hoặc nhiều thành phần cú pháp dùng để suy diễn trị số kích cỡ khối cho phép lớn nhất để sử dụng phân chia TT theo cấu trúc dữ liệu liên quan.

Theo một số phương án, trị số kích cỡ khối cho phép lớn nhất để sử dụng phân chia TT được ràng buộc theo yêu cầu tương hợp với dòng bit. Do đó, để xác định liệu phân chia TT được phép cho CU hiện thời hay không, thì hệ thống lập mã video có thể chỉ kiểm tra xem liệu chiều rộng hoặc chiều cao của CU hiện thời lớn hơn kích cỡ khối cho phép lớn nhất để sử dụng phân chia TT và không kiểm tra tiếp điều đó nữa đối với kích cỡ VPDU cho từng CU hiện thời. Ví dụ, khi kích cỡ VPDU bằng 64, hệ thống lập mã video có thể ràng buộc kích cỡ khối cho phép lớn nhất để sử dụng phân chia TT được báo hiệu trong dòng bit sao cho nhỏ hơn hoặc bằng 64.

Theo một số phương án, tham số maxTtSize được ràng buộc sao cho nhỏ hơn

hoặc bằng 64, trong đó biến maxTtSize được suy dẫn từ một hoặc nhiều trị số thành phần cú pháp được lập mã để biểu diễn kích cỡ khối cho phép lớn nhất để sử dụng phân chia TT. Theo một số phương án, biến maxTtSize được suy dẫn từ thành phần cú pháp cấp cao được lập mã log2_diff_max_tt_min_tt là:

$\text{maxTtSize} = 1 \ll (\log2_diff_max_tt_min_tt + \log2MinMaxTT)$ theo điều kiện ràng buộc trong đó trị số lớn nhất cho phép của log2_diff_max_tt_min_tt bằng $\text{Min}(6, \text{CtbLog2SizeY}) - \log2MinMaxTT$. Biến log2MinMaxTT biểu thị lôgarit cơ số 2 của trị số nhỏ nhất cho phép của kích cỡ TT lớn nhất và CtbLog2SizeY biểu thị lôgarit cơ số 2 của kích cỡ CTB được xác định.

Quy trình xác định liệu phân chia TT được phép cho CU hiện thời hay không có thể được đơn giản hơn nữa. Do đó, biến allowTtSplit được suy dẫn như sau:

- Nếu một hoặc nhiều điều kiện trong số những điều kiện sau là true, allowTtSplit được thiết lập là FALSE:
 - cbSize nhỏ hơn hoặc bằng $2 * \text{MinTtSizeY}$
 - cbWidth lớn hơn maxTtSize
 - cbHeight lớn hơn maxTtSize
 - mttDepth lớn hơn hoặc bằng maxMttDepth
 - $x0 + \text{cbWidth}$ lớn hơn pic_width_in_luma_samples
 - $y0 + \text{cbHeight}$ lớn hơn pic_height_in_luma_samples
 - treeType bằng DUAL_TREE_CHROMA và $(\text{cbWidth} / \text{SubWidthC}) * (\text{cbHeight} / \text{SubHeightC})$ nhỏ hơn hoặc bằng 32
 - treeType bằng DUAL_TREE_CHROMA và $(\text{cbWidth} / \text{SubWidthC})$ bằng 8 và ttSplit bằng SPLIT_TT_VER
 - treeType bằng DUAL_TREE_CHROMA và modeType bằng MODE_TYPE_INTRA
 - $\text{cbWidth} * \text{cbHeight}$ bằng 64 và modeType bằng MODE_TYPE_INTER
 - Nếu không thì, allowTtSplit được thiết lập là TRUE.

Để thỏa mãn điều kiện ràng buộc VDP, theo một số phương án, các biến minQtSizeY và minQtSizeC được suy dẫn từ một hoặc nhiều trị số thành phần cú pháp được lập mã để biểu diễn kích cỡ khối cho phép nhỏ nhất cho nút lá QT cho luma và chroma, một cách tương ứng, và các giá trị của các biến minQtSizeY và minQtSizeC được ràng buộc nhỏ hơn hoặc bằng 64 trong slice nội ảnh khi cơ chế cây lưỡng nội ảnh

được sử dụng. Theo một số phương án, các trị số của biến minQtSizeY và minQtSizeC được ràng buộc nhỏ hơn hoặc bằng $\text{Min}(64, \text{CtbSizeY})$ trong slice nội ảnh khi cơ chế cây lưỡng nội ảnh được sử dụng, trong đó các biến minQtSizeY và minQtSizeC được suy dẫn từ một hoặc nhiều trị số thành phần cú pháp được lập mã để biểu thị kích cỡ khối cho phép nhỏ nhất trong các mẫu luma cho nút lá QT cho luma và chroma, một cách tương ứng, và biến CtbSizeY biểu thị kích cỡ CTB trong các mẫu luma trong chuỗi video được lập mã.

Theo một số phương án, khi cơ chế cây lưỡng nội ảnh được sử dụng, CU hiện thời có kích cỡ lớn hơn 64 có thể được phân chia tiếp thành các CU nhỏ hơn trong đó mỗi CU được chứa bên trong các vùng VPDU bằng phân chia cây tứ phân.

Theo một số phương án, các trị số của biến minQtSizeY và minQtSizeC được ràng buộc nhỏ hơn hoặc bằng $\text{Min}(64, \text{CtbSizeY})$ cho tất cả các dạng slice. Theo đó, khi CU hiện thời có kích cỡ lớn hơn 64 vượt qua một phần (hoặc mở rộng qua) biên giới ảnh bên phải hoặc dưới cùng, thì CU hiện thời có thể được phân chia tiếp thành các CU nhỏ hơn bằng phân chia cây tứ phân.

III. Các điều kiện ràng buộc đối với chiều sâu phân chia

Theo một số phương án, các tham số ràng buộc đối với chiều sâu MTT lớn nhất được xác định. Các tham số ràng buộc này dùng để điều khiển cấu trúc phân chia CU dựa vào cây tứ phân được lồng với cây đa phân bằng cách sử dụng phân chia nhị phân và/hoặc phân chia tam phân. Theo một khía cạnh của sáng chế, chiều sâu MTT lớn nhất được ràng buộc sao cho lớn hơn 0 khi kích cỡ QT nhỏ nhất lớn hơn kích cỡ khối lập mã nhỏ nhất. Theo một số phương án, chiều sâu MTT lớn nhất được ràng buộc sao cho lớn hơn 0 khi kích cỡ QT nhỏ nhất lớn hơn kích cỡ khối lập mã nhỏ nhất.

Theo một số phương án, thành phần cú pháp $\text{log2_ctu_size_minus5}$ cộng 5 xác định (lôgarit cơ số 2 của) kích cỡ khối cây lập mã luma của từng CTU. Thành phần cú pháp $\text{log2_ctu_size_minus5}$ có thể được sử dụng để suy dẫn các biến CtbLog2SizeY và CtbSizeY theo điều kiện sau:

$$\text{CtbLog2SizeY} = \text{log2_ctu_size_minus5} + 5$$

$$\text{CtbSizeY} = 1 \ll \text{CtbLog2SizeY}$$

Theo một số phương án, thành phần cú pháp SPS $\text{log2_min_luma_coding_block_size_minus2}$ cộng 2 xác định lôgarit cơ số 2 của kích cỡ khối lập mã luma nhỏ nhất. Thành phần cú pháp

$\log2_min_luma_coding_block_size_minus2$ có thể được sử dụng để suy dẫn kích cỡ nhỏ nhất của CB trong các mẫu luma, được biểu diễn dưới dạng $MinCbLog2SizeY$ hoặc $MinCbSizeY$. Theo một số phương án, $MinCbSizeY$ được ràng buộc sao cho nhỏ hơn hoặc bằng $Min(64, CtbSizeY)$. $MinCbLog2SizeY$ và $MinCbSizeY$ có thể được suy dẫn theo điều kiện sau:

$$MinCbLog2SizeY = \log2_min_luma_coding_block_size_minus2 + 2$$

$$MinCbSizeY = 1 \ll MinCbLog2SizeY$$

Theo một số phương án, thành phần cú pháp tiêu đề slice (SH) $slice_log2_diff_min_qt_min_cb_luma$ xác định sai lệch giữa lôgarit cơ số 2 của kích cỡ nhỏ nhất trong các mẫu luma của khối lá luma thu được từ phân chia QT của CTU và lôgarit cơ số 2 của kích cỡ khối lập mã nhỏ nhất trong các mẫu luma cho các CU luma ở slice hiện thời.

Theo một số phương án, thành phần cú pháp tiêu đề slice (SH) $slice_max_mtt_hierarchy_depth_luma$ xác định chiều sâu phân cấp lớn nhất cho các đơn vị lập mã thu được từ phân chia cây đa phân của lá cây tứ phân ở slice hiện thời. Khi $slice_max_mtt_hierarchy_depth_luma$ khác không, các thành phần cú pháp $slice_log2_diff_max_bt_min_qt_luma$ và $slice_log2_diff_max_tt_min_qt_luma$ được lập mã ở tiêu đề slice hiện thời. Thành phần cú pháp tiêu đề slice $slice_log2_diff_max_bt_min_qt_luma$ xác định sai lệch giữa (lôgarit cơ số 2 của) kích cỡ lớn nhất (chiều rộng hoặc chiều cao trong các mẫu luma) của khối lập mã luma mà có thể được phân chia bằng phân chia nhị phân và (lôgarit cơ số 2 của) kích cỡ nhỏ nhất (chiều rộng hoặc chiều cao trong các mẫu luma) của khối lá luma thu được từ phân chia cây tứ phân của CTU ở slice hiện thời. Thành phần cú pháp SH $slice_log2_diff_max_tt_min_qt_luma$ xác định sai lệch giữa (lôgarit cơ số 2 của) kích cỡ lớn nhất (chiều rộng hoặc chiều cao trong các mẫu luma) của khối lập mã luma mà có thể được phân chia bằng phân chia tam phân và (lôgarit cơ số 2 của) kích cỡ nhỏ nhất (chiều rộng hoặc chiều cao trong các mẫu luma) của khối lá luma thu được từ phân chia cây tứ phân của CTU ở slice hiện thời.

Biến $MinQtLog2SizeY$ hoặc $MinQtSizeY$ biểu thị kích cỡ nhỏ nhất trong các mẫu luma của khối lá luma thu được từ phân chia QT của CTU. Các trị số $slice_log2_diff_max_bt_min_qt_luma$ và $slice_log2_diff_max_tt_min_qt_luma$ được ràng buộc sao cho nằm trong khoảng từ 0 đến $CtbLog2SizeY - MinQtLog2SizeY$. Trị

số MinQtSizeY được ràng buộc sao cho nằm trong phạm vi (MinCbSizeY, Min(64,CtbSizeY)).

Thành phần cú pháp slice_log2_diff_min_qt_min_cb_luma có thể được sử dụng để suy dẫn biến MinQtLog2SizeY và MinQtSizeY theo điều kiện sau:

$$\begin{aligned} \text{MinQtLog2SizeY} &= \text{MinCbLog2SizeY} + \text{slice_log2_diff_min_qt_min_cb_luma} \\ \text{MinQtSizeY} &= 1 \ll \text{MinQtLog2SizeY} \end{aligned}$$

Vì trị số MinQtSizeY được ràng buộc nằm trong phạm vi (MinCbSizeY, Min(64,CtbSizeY)), phạm vi của slice_log2_diff_min_qt_min_cb_luma là từ 0 đến Min(6,CtbLog2SizeY) – MinCbLog2SizeY.

Biến MaxBtSizeY biểu thị kích cỡ lớn nhất trong các mẫu luma của khối lập mã luma mà có thể được phân chia bằng phân chia nhị phân. Biến MaxBtSizeY được suy dẫn như sau:

$$\begin{aligned} \text{MaxBtSizeY} &= \\ 1 \ll & (\text{MinQtLog2SizeY} + \text{slice_log2_diff_max_bt_min_qt_luma}) \end{aligned}$$

Theo một số phương án, kích cỡ BT lớn nhất được ràng buộc sao cho lớn hơn hoặc bằng kích cỡ QT nhỏ nhất. Theo cách này, khi khối lập mã hiện thời (là kết quả của phép phân chia) đạt kích cỡ QT nhỏ nhất, thì khối lập mã có thể được phân chia đệ quy tiếp bằng phân chia nhị phân cho đến khi khối lập mã hiện thời đạt kích cỡ khối lập mã nhỏ nhất. Theo một số phương án, kích cỡ TT lớn nhất được ràng buộc sao cho lớn hơn hoặc bằng kích cỡ QT nhỏ nhất. Theo cách này, khi khối lập mã hiện thời (là kết quả của phép phân chia) đạt kích cỡ QT nhỏ nhất, thì khối lập mã có thể được phân chia đệ quy tiếp bằng phân chia tam phân cho đến khi khối lập mã hiện thời đạt kích cỡ khối lập mã nhỏ nhất.

Biến MaxTtSizeY biểu thị kích cỡ lớn nhất trong các mẫu luma của khối lập mã luma mà có thể được phân chia bằng phân chia tam phân. Biến MaxTtSizeY có thể được suy dẫn từ thành phần cú pháp slice_log2_diff_max_tt_min_qt_luma như sau:

$$\begin{aligned} \text{MaxTtSizeY} &= \\ 1 \ll & (\text{MinQtLog2SizeY} + \text{slice_log2_diff_max_tt_min_qt_luma}) \end{aligned}$$

Vì MaxTtSizeY nằm trong khoảng từ MinQtSize đến Min (64, CtbSizeY), nên slice_log2_diff_max_tt_min_qt_luma nằm trong khoảng từ 0 đến Min(6,CtbLog2SizeY) – MinQtLog2SizeY.

Theo một số phương án, các thành phần cú pháp tiêu đề slice

`slice_log2_diff_min_qt_min_cb_chroma`, `slice_log2_diff_max_bt_min_qt_chroma`, và `slice_log2_diff_max_tt_min_qt_chroma` tương ứng xác định kích cỡ QT nhỏ nhất, kích cỡ BT lớn nhất, và kích cỡ TT lớn nhất (trong các mẫu luma) của khối lập mã chroma. Theo một số phương án, các giá trị mặc định của các thành phần cú pháp SH được xác định bằng các thành phần cú pháp tương ứng liên quan trong SPS. Ví dụ, theo một số phương án, có thể là các tham số SPS tương ứng `sps_log2_diff_min_qt_min_cb_chroma`, `sps_log2_diff_max_bt_min_qt_chroma`, và `sps_log2_diff_max_tt_min_qt_chroma` tương ứng xác định kích cỡ QT nhỏ nhất, kích cỡ BT lớn nhất, và kích cỡ TT lớn nhất (trong các mẫu luma) của khối lập mã chroma tại phân mức SPS.

Theo một số phương án, kích cỡ BT lớn nhất và kích cỡ TT lớn nhất được suy dẫn có thể lớn hơn hoặc bằng kích cỡ QT nhỏ nhất. Do đó, khi khối lập mã hiện thời đạt kích cỡ QT nhỏ nhất và phân chia nhị phân hoặc tam phân được sử dụng, thì khối lập mã hiện thời có thể được phân chia đệ quy tiếp bằng phân chia nhị phân hoặc tam phân cho đến khi khối lập mã hiện thời bằng kích cỡ khối lập mã nhỏ nhất. Tuy nhiên, khi `slice_max_mtt_hierarchy_depth_luma` bằng 0, thì cả phân chia nhị phân và phân chia tam phân được hủy bỏ. Theo một số phương án, không có khối lập mã nào có thể được phân chia thành khối lập mã nhỏ nhất nếu kích cỡ QT nhỏ nhất lớn hơn kích cỡ khối lập mã nhỏ nhất, vì điều đó có thể gây ra vấn đề về phân chia các CU dọc các biên giới ảnh, trong đó mỗi CU nằm ngoài biên giới được yêu cầu được phân chia đệ quy tiếp cho đến khi toàn bộ CU nằm trong các biên giới ảnh. Theo một số phương án, cú pháp `slice_max_mtt_hierarchy_depth_luma` được ràng buộc sao cho lớn hơn 0 khi `slice_log2_diff_min_qt_min_cb_luma` khác không.

FIG. 8 minh họa về mặt khái niệm các tham số ràng buộc dùng để điều khiển cấu trúc phân chia của khối điểm ảnh 800. Khối điểm ảnh 800 là CTU mà được phân chia bằng cách sử dụng cả QT và MTT. Hình vẽ này minh họa CTU 800 cùng với cây 810 thể hiện những bộ phận của cấu trúc phân chia của CTU 800.

Kích cỡ của CTU 800 là 128x128. CTU 800 được ràng buộc để hủy bỏ phân chia MTT (ví dụ, TT) đối với những phân vùng có kích cỡ lớn hơn 64. Nói cách khác, phân chia MTT không được phép cho CTU 800, nhưng phân chia MTT được phép cho các phân vùng QT của CTU 800 mà có kích cỡ 64x64 hoặc nhỏ hơn. Kích cỡ khối cho phép lớn nhất cho MTT có thể được xác định dựa vào kích cỡ của VPDU để đảm bảo

rằng mỗi VPDU có thể được xử lý hoàn toàn bằng trạng thái đường truyền giống nhau.

Kích cỡ cho phép lớn nhất cho MTT (ví dụ, các biến `MaxBtSizeY` hoặc `MaxTtSizeY`) có thể còn được báo hiệu bằng cách tham chiếu kích cỡ nhỏ nhất của lá QT (biến `minQtSizeY`). Ví dụ, kích cỡ cho phép lớn nhất cho MTT có thể được biểu diễn dưới dạng sai lệch giữa lôgarit cơ số hai của `MaxTtSizeY` và lôgarit cơ số hai của `MinQtSizeY`, ví dụ, dưới dạng `slice_log2_diff_max_tt_min_qt_luma` hoặc tín hiệu tương ứng trong tiêu đề ảnh hoặc SPS. Kích cỡ nhỏ nhất của lá QT có thể được báo hiệu bằng cách tham chiếu kích cỡ nhỏ nhất của CB (biến `MinCbSizeY`). Ví dụ, kích cỡ nhỏ nhất của lá QT có thể được biểu diễn dưới dạng sai lệch giữa lôgarit cơ số hai của `MinQtSizeY` và lôgarit cơ số hai của `MinCbSizeY`, ví dụ, dưới dạng `slice_log2_diff_min_qt_min_cb_luma` hoặc tín hiệu tương ứng trong tiêu đề ảnh hoặc SPS.

IV. Báo hiệu cấu trúc phân chia khối

Theo một số phương án, thông tin về chiều sâu MTT lớn nhất được báo hiệu dựa vào sai lệch giữa lôgarit cơ số 2 của kích cỡ QT nhỏ nhất và lôgarit cơ số 2 của kích cỡ khối lập mã nhỏ nhất. Theo một số phương án, biến `MaxMttDepthY` biểu thị chiều sâu phân cấp lớn nhất cho các đơn vị lập mã thu được từ phân chia cây đa phân của lá cây tứ phân, nằm trong phạm vi $(0, 2 * (CtbLog2SizeY - MinCbLog2SizeY))$. Biến `MaxMttDepthY` có thể được suy dẫn như sau:

$$\text{MaxMttDepthY} = \text{slice_max_mtt_hierarchy_depth_luma} + ((1 + \text{slice_log2_diff_min_qt_min_cb_luma}) \gg 1).$$

Theo một số phương án, các tham số ràng buộc dùng để điều khiển cấu trúc phân chia CU được xác định. Theo một số phương án, khi khối lập mã hiện thời đạt (hoặc có) kích cỡ QT nhỏ nhất, thì nó được phép được phân chia tiếp một cách đệ quy cho đến khi đạt kích cỡ khối lập mã nhỏ nhất miễn là ít nhất một trong số các dạng phân chia MTT (BT và TT) vẫn được sử dụng. Bộ lập mã video có thể báo hiệu kích cỡ BT lớn nhất hoặc kích cỡ TT lớn nhất nhỏ hơn kích cỡ QT nhỏ nhất, trong đó ít nhất một trong số kích cỡ BT lớn nhất và kích cỡ TT lớn nhất được ràng buộc sao cho lớn hơn hoặc bằng kích cỡ QT nhỏ nhất khi chiều sâu MTT hiện thời khác không. Theo đó, cấu trúc MTT được xác định có thể hủy bỏ một trong số các dạng phân chia MTT cho đối với khối lập mã lớn hơn điều kiện ràng buộc kích cỡ mà nhỏ hơn kích cỡ QT nhỏ nhất. Cụ thể, bộ lập mã video có thể hủy bỏ chỉ một trong số của các dạng MTT bằng cách

xác định điều kiện ràng buộc kích cỡ tương ứng bằng kích cỡ khối lập mã nhỏ nhất.

Theo một số phương án, thành phần cú pháp thứ nhất mà báo hiệu kích cỡ lớn nhất cho dạng MTT thứ nhất sẽ xác định sai lệch giữa (lôgarit cơ số 2 của) kích cỡ lớn nhất trong các mẫu luma cho dạng MTT thứ nhất và (lôgarit cơ số 2 của) kích cỡ khối lập mã nhỏ nhất trong các mẫu luma cho các CU luma ở slice hiện thời. Khi kích cỡ lớn nhất cho dạng MTT thứ nhất nhỏ hơn kích cỡ QT nhỏ nhất và kích cỡ QT nhỏ nhất lớn hơn kích cỡ khối lập mã nhỏ nhất, thì thành phần cú pháp thứ hai mà báo hiệu kích cỡ lớn nhất cho dạng MTT thứ hai sẽ xác định sai lệch giữa (lôgarit cơ số 2 của) kích cỡ lớn nhất trong các mẫu luma cho dạng MTT thứ nhất và (lôgarit cơ số 2 của) kích cỡ nhỏ nhất (chiều rộng hoặc chiều cao) trong các mẫu luma của khối lá luma thu được từ phân chia cây tứ phân của CTU ở slice hiện thời. Nếu không thì, thành phần cú pháp thứ hai mà báo hiệu kích cỡ lớn nhất cho dạng MTT thứ hai sẽ xác định sai lệch giữa (lôgarit cơ số 2 của) kích cỡ lớn nhất trong các mẫu luma cho dạng MTT thứ hai và (lôgarit cơ số 2 của) kích cỡ khối lập mã nhỏ nhất trong các mẫu luma cho các CU luma ở slice hiện thời.

Bảng 1 đưa ra bảng cú pháp được lấy làm ví dụ của tiêu đề slice (SH) dùng để xác định cấu trúc phân chia khối. Cú pháp vừa nêu cho phép một trong số các dạng MTT có kích cỡ lớn nhất được xác định nhỏ hơn kích cỡ QT nhỏ nhất.

Bảng 1:

slice_header() {	Biến mô tả
....	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	ue(v)
.if(partition_constraints_override_flag) {	
slice_log2_diff_min_qt_min_cb_luma	ue(v)
slice_max_mtt_hierarchy_depth_luma	ue(v)
if(slice_max_mtt_hierarchy_depth_luma != 0) {	
slice_log2_max_bt_luma	ue(v)
slice_log2_max_tt_luma	ue(v)
}	
if(slice_type == I && qtbtt_dual_tree_intra_flag) {	
slice_log2_diff_min_qt_min_cb_chroma	ue(v)
slice_max_mtt_hierarchy_depth_chroma	ue(v)

if(slice_max_mtt_hierarchy_depth_chroma != 0) {	
slice_log2_max_bt_chroma	uc(v)
slice_log2_max_tt_chroma	ue(v)
}	
}	
}	
}	
....	
}	

Các biến MaxBtSizeY và MaxTtSizeY được suy dẫn như sau:

$$\text{MaxBtSizeY} = 1 \ll (\text{MinCbLog2SizeY} + \text{slice_log2_max_bt_luma})$$

$$\text{MaxTtSizeY} = 1 \ll$$

$$((\text{MaxBtSizeY} < \text{MinQtSizeY} \ \&\& \ \text{MinQtLog2SizeY} > \text{MinCbLog2SizeY}) ?$$

$$\text{MinQtLog2SizeY} : \text{MinCbLog2SizeY}) + \text{slice_log2_max_tt_luma}).$$

Theo một số phương án khác, các biến MaxBtSizeY và MaxTtSizeY được suy dẫn như sau:

$$\text{MaxBtSizeY} = 1 \ll (\text{MinCbLog2SizeY} + \text{slice_log2_max_bt_luma})$$

$$\text{MaxTtSizeY} = 1 \ll ((\text{MaxBtSizeY} < \text{MinQtSizeY}) ?$$

$$\text{MinQtLog2SizeY} : \text{MinCbLog2SizeY}) + \text{slice_log2_max_tt_luma}).$$

Theo một số phương án khác, thành phần cú pháp thứ nhất mà báo hiệu kích cỡ lớn nhất cho dạng MTT thứ nhất sẽ xác định sai lệch giữa lôgarit cơ số 2 của kích cỡ lớn nhất trong các mẫu luma cho dạng MTT thứ nhất và lôgarit cơ số 2 của kích cỡ khối lập mã nhỏ nhất trong các mẫu luma cho các CU luma ở slice hiện thời. Thành phần cú pháp thứ hai mà báo hiệu kích cỡ lớn nhất cho dạng MTT thứ hai sẽ xác định sai lệch giữa lôgarit cơ số 2 của kích cỡ lớn nhất trong các mẫu luma cho dạng MTT thứ hai và lôgarit cơ số 2 của kích cỡ khối lập mã nhỏ nhất trong các mẫu luma cho các CU luma ở slice hiện thời. Các biến MaxBtSizeY và MaxTtSizeY được suy dẫn như sau:

$$\text{MaxBtSizeY} = 1 \ll (\text{MinCbLog2SizeY} + \text{slice_log2_max_bt_luma})$$

$$\text{MaxTtSizeY} = 1 \ll (\text{MinCbLog2SizeY} + \text{slice_log2_max_tt_luma}).$$

Theo một số phương án, ít nhất một trị số của các biến MaxBtSizeY và MaxBtSizeY được ràng buộc sao cho lớn hơn hoặc bằng MinQtSizeY.

Cần được lưu ý rằng phương pháp đề xuất và các phương án liên quan có thể được áp dụng tiếp xác định các thành phần cú pháp liên quan khác để báo hiệu kích cỡ BT lớn nhất và kích cỡ TT lớn nhất để phân chia các khối lập mã chroma ở tiêu đề slice cho slice nội ảnh khi cơ chế phân chia cây lưỡng được sử dụng (qtbtt_dual_tree_intra_flag bằng 1). Phương pháp đề xuất và các phương án liên quan có thể được áp dụng tiếp xác định các thành phần cú pháp liên quan khác để báo hiệu kích cỡ BT lớn nhất mặc định và kích cỡ TT lớn nhất mặc định trong SPS.

Theo một số phương án, khi khối lập mã hiện thời đạt kích cỡ QT nhỏ nhất, thì nó được phép được phân chia tiếp một cách đệ quy cho đến khi kích cỡ khối lập mã nhỏ nhất đạt được (ví dụ, bằng cách sử dụng phân chia nhị phân hoặc tam phân cho kích cỡ khối lập mã nhỏ hơn hoặc bằng kích cỡ QT nhỏ nhất mà không có giới hạn bất kỳ nào về kích cỡ lớn nhất cho phân chia nhị phân hoặc tam phân). Theo một số phương án, bộ lập mã video có thể báo hiệu kích cỡ TT lớn nhất nhỏ hơn kích cỡ QT nhỏ nhất theo điều kiện ràng buộc trong đó kích cỡ BT lớn nhất lớn hơn hoặc bằng kích cỡ QT nhỏ nhất. Theo đó, cấu trúc MTT được xác định ở trên có thể hủy phân chia tam phân cho các khối lập mã có các kích cỡ lớn hơn điều kiện ràng buộc kích cỡ được xác định mà nhỏ hơn kích cỡ QT nhỏ nhất. Cụ thể là, bộ lập mã video có thể hủy một cách hữu hiệu phân chia tam phân bằng cách xác định điều kiện ràng buộc kích cỡ tương ứng có bằng kích cỡ khối lập mã nhỏ nhất không.

Theo một số phương án, thành phần cú pháp mà báo hiệu kích cỡ lớn nhất (chiều rộng hoặc chiều cao) cho dạng phân chia nhị phân sẽ xác định sai lệch giữa (lôgarit cơ số 2 của) kích cỡ lớn nhất (trong các mẫu luma) của khối lập mã luma mà có thể được phân chia bằng phân chia nhị phân và (lôgarit cơ số 2 của) kích cỡ khối lập mã nhỏ nhất (chiều rộng hoặc chiều cao trong các mẫu luma) của khối lá luma (mà tạo thành từ phân chia cây tứ phân của CTU ở slice hiện thời). Thành phần cú pháp khác mà báo hiệu kích cỡ lớn nhất cho dạng phân chia tam phân sẽ xác định sai lệch giữa (lôgarit cơ số 2 của) kích cỡ lớn nhất (chiều rộng hoặc chiều cao trong các mẫu luma) của khối lập mã luma mà có thể được phân chia bằng phân chia tam phân và (lôgarit cơ số 2 của) kích cỡ khối lập mã nhỏ nhất trong các mẫu luma cho các CU luma ở slice hiện thời. Bảng 2 đưa ra bảng cú pháp được lấy làm ví dụ của tiêu đề slice có thể hỗ trợ dạng phân chia tam phân có kích cỡ lớn nhất được xác định nhỏ hơn kích cỡ QT nhỏ nhất.

Bảng 2:

slice_header() {	Biến mô tả
....	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	ue(v)
if(partition_constraints_override_flag) {	
slice_log2_diff_min_qt_min_cb_luma	ue(v)
slice_max_mtt_hierarchy_depth_luma	ue(v)
if(slice_max_mtt_hierarchy_depth_luma != 0) {	
slice_log2_diff_max_bt_min_qt_luma	ue(v)
slice_log2_diff_max_tt_min_cb_luma	ue(v)
}	
if(slice_type == I && qtbtt_dual_tree_intra_flag) {	
slice_log2_diff_min_qt_min_cb_chroma	ue(v)
slice_max_mtt_hierarchy_depth_chroma	ue(v)
if(slice_max_mtt_hierarchy_depth_chroma != 0) {	
slice_log2_diff_max_bt_min_qt_chroma	ue(v)
slice_log2_diff_max_tt_min_cb_chroma	ue(v)
}	
}	
}	
}	
}	
....	
}	

Các biến MaxBtSizeY biểu thị kích cỡ lớn nhất của CB luma cho phép phân chia BT. Biến MaxTtSizeY biểu thị kích cỡ lớn nhất của CB luma cho phép phân chia TT. MaxBtSizeY và MaxTtSizeY nằm trong khoảng từ MinQtSize đến Min (64, CtbSizeY). MaxBtSizeY và MaxTtSizeY được suy dẫn như sau:

MaxBtSizeY =

$1 \ll (\text{MinQtLog2SizeY} + \text{slice_log2_diff_max_bt_min_qt_luma})$

MaxTtSizeY =

$1 \ll (\text{MinCbLog2SizeY} + \text{slice_log2_diff_min_tt_min_cb_luma})$.

Các biến MaxBtSizeC biểu thị kích cỡ lớn nhất của CB chroma cho phép phân chia

BT. Biến MaxTtSizeY biểu thị kích cỡ lớn nhất của CB chroma cho phép phân chia

TT. Các biến MaxBtSizeC và MaxTtSizeC được suy dẫn như sau:

MaxBtSizeC =

$$1 \ll (\text{MinQtLog2SizeC} + \text{slice_log2_diff_max_bt_min_qt_chroma})$$

MaxTtSizeC =

$$1 \ll (\text{MinCbLog2SizeY} + \text{slice_log2_diff_min_tt_min_cb_chroma}).$$

Cần được lưu ý rằng các phương pháp đề xuất và các phương án liên quan được mô tả ở các phần từ phần II đến phần IV có thể được áp dụng tiếp cho các thành phần cú pháp liên quan khác để báo hiệu thông tin nhằm suy dẫn chiều sâu MTT lớn nhất để phân chia các khối lập mã chroma ở tiêu đề slice cho slice nội ảnh khi cơ chế phân chia cây lưỡng được sử dụng (qtbtt_dual_tree_intra_flag bằng 1). Các phương pháp đề xuất có thể còn được sử dụng để báo hiệu các tham số ràng buộc liên quan ở tiêu đề ảnh. Phương pháp đề xuất và các phương án liên quan có thể được áp dụng tiếp xác định các thành phần cú pháp liên quan khác để báo hiệu thông tin nhằm suy dẫn chiều sâu MTT lớn nhất mặc định trong SPS. Ví dụ, có thể là các tham số tiêu đề ảnh và các tham số SPS tương ứng với các tham số tiêu đề slice chẳng hạn slice_log2_diff_max_tt_min_qt_luma và slice_log2_diff_min_qt_min_cb_luma thiết lập các giá trị mặc định cho các tham số nêu trên ở phân mức ảnh hoặc chuỗi ảnh.

Bất kỳ phương pháp nào trong số các phương pháp được đề xuất đã nêu 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 được đề xuất có thể được thực hiện ở môđun mã hóa entropy (hoặc mã hóa cú pháp cấp cao) của bộ mã hóa, và/hoặc môđun giải mã entropy (hoặc giải mã cú pháp cấp cao) của bộ giải mã. Ngoài ra, bất kỳ phương pháp nào trong số các phương pháp được đề xuất có thể được thực hiện dưới dạng mạch được tích hợp vào môđun mã hóa entropy (hoặc mã hóa cú pháp cấp cao) của bộ mã hóa và/hoặc môđun giải mã entropy (hoặc giải mã cú pháp cấp cao) của bộ giải mã. Bất kỳ phương pháp nào trong số các phương pháp được đề xuất ở trên có thể còn được thực hiện ở các bộ mã hóa và/hoặc các bộ giải mã hình ảnh, trong đó dòng bit được tạo ra tương ứng là một khung hình được lập mã bằng cách chỉ sử dụng dự đoán nội ảnh

V. Bộ mã hóa video được lấy làm ví dụ

FIG. 9 minh họa bộ mã hóa video được lấy làm ví dụ 900 để báo hiệu các điều

kiện ràng buộc đối với phân chia khối. Như được minh họa, bộ mã hóa video 900 nhận tín hiệu video đầu vào từ nguồn video 905 và mã hóa tín hiệu vừa nêu thành dòng bit 995. Bộ mã hóa video 900 có một số bộ phận hoặc môđun dùng để mã hóa tín hiệu từ nguồn video 905, ít nhất bao gồm một số bộ phận được chọn từ môđun biến đổi 910, môđun lượng tử hóa 911, môđun lượng tử hóa ngược 914, môđun biến đổi ngược 915, môđun ước lượng nội ảnh 920, môđun dự đoán nội ảnh 925, môđun bù chuyển động 930, môđun ước lượng chuyển động 935, bộ lọc vòng lặp kín 945, bộ đệm hình ảnh tái tạo 950, bộ đệm MV 965, và môđun dự đoán MV 975, và bộ mã hóa entropy 990. Môđun bù chuyển động 930 và môđun ước lượng chuyển động 935 là một phần của môđun dự đoán liên ảnh 940.

Theo một số phương án, các môđun từ 910 đến 990 là các môđun có các lệnh phần mềm được xử lý bởi một hoặc nhiều đơn vị xử lý (ví dụ, bộ xử lý) của thiết bị tính toán hoặc thiết bị điện tử. Theo một số phương án, các môđun từ 910 đến 990 là các môđun có các mạch phần cứng được thực hiện bởi một hoặc nhiều mạch tích hợp IC (Integrated Circuit - IC) của thiết bị điện tử. Mặc dù các môđun từ 910 đến 990 được minh họa dưới dạng các môđun riêng biệt, song một số trong số các môđun đó có thể được kết hợp thành một môđun duy nhất.

Nguồn video 905 cấp tín hiệu video thô thể hiện dữ liệu điểm ảnh của từng khung hình video chưa nén. Bộ trừ 908 tính sai lệch giữa dữ liệu điểm ảnh video thô của nguồn video 905 và dữ liệu điểm ảnh được dự đoán 913 từ môđun bù chuyển động 930 hoặc môđun dự đoán nội ảnh 925. Môđun biến đổi 910 biến sai lệch (hoặc phần dư dữ liệu điểm ảnh hoặc tín hiệu dư thừa 909) thành các hệ số biến đổi (ví dụ, bằng phép biến đổi cosin rời rạc, hay gọi là phép DCT). Môđun lượng tử hóa 911 lượng tử hóa các hệ số biến đổi thành dữ liệu được lượng tử hóa (hoặc các hệ số được lượng tử hóa) 912, được mã hóa thành dòng bit 995 bởi bộ mã hóa entropy 990.

Môđun lượng tử hóa ngược 914 giải lượng tử dữ liệu được lượng tử hóa (hoặc các hệ số được lượng tử hóa) 912 để thu được các hệ số biến đổi, và môđun biến đổi ngược 915 thực hiện phép biến đổi ngược đối với các hệ số biến đổi để tạo ra phần dư tái tạo 919. Phần dư tái tạo 919 được cộng với dữ liệu điểm ảnh được dự đoán 913 để tạo ra dữ liệu điểm ảnh được tái tạo 917. Theo một số phương án, dữ liệu điểm ảnh được tái tạo 917 được lưu tạm thời ở bộ đệm dòng (không được minh họa) để dự đoán nội ảnh và dự đoán MV không gian. Các điểm ảnh được tái tạo được lọc bởi bộ lọc

vòng lặp kín 945 và được lưu ở bộ đệm hình ảnh tái tạo 950. Theo một số phương án, bộ đệm hình ảnh tái tạo 950 là bộ nhớ nằm ngoài bộ mã hóa video 900. Theo một số phương án, bộ đệm hình ảnh tái tạo 950 là bộ nhớ nằm trong bộ mã hóa video 900.

Môđun ước lượng nội ảnh 920 thực hiện dự đoán nội ảnh dựa vào dữ liệu điểm ảnh được tái tạo 917 để tạo ra dữ liệu dự đoán nội ảnh. Dữ liệu dự đoán nội ảnh được cấp đến bộ mã hóa entropy 990 để được mã hóa thành dòng bit 995. Dữ liệu dự đoán nội ảnh còn được sử dụng bởi môđun dự đoán nội ảnh 925 để tạo ra dữ liệu điểm ảnh được dự đoán 913.

Môđun ước lượng chuyển động 935 thực hiện dự đoán liên ảnh bằng cách tạo ra các MV để tham chiếu dữ liệu điểm ảnh của các khung hình được giải mã trước đó được lưu ở bộ đệm hình ảnh tái tạo 950. Những MV này được cấp đến môđun bù chuyển động 930 để tạo ra dữ liệu điểm ảnh được dự đoán.

Thay vì mã hóa các MV thực hoàn toàn trong dòng bit, bộ mã hóa video 900 sử dụng dự đoán MV để tạo ra các MV được dự đoán, và sai lệch giữa các MV được sử dụng để bù chuyển động và các MV được dự đoán được mã hóa dưới dạng dữ liệu chuyển động dư thừa và được lưu trong dòng bit 995.

Môđun dự đoán MV 975 tạo ra các MV được dự đoán dựa vào các MV tham chiếu mà được tạo ra để lập mã các khung hình video trước đó, tức là, các MV bù chuyển động được sử dụng để thực hiện bù chuyển động. Môđun dự đoán MV 975 gọi các MV tham chiếu từ các khung hình video trước đó từ bộ đệm MV 965. Bộ mã hóa video 900 lưu các MV được tạo ra cho khung hình video hiện thời ở bộ đệm MV 965 dưới dạng các MV tham chiếu để tạo ra các MV được dự đoán.

Môđun dự đoán MV 975 sử dụng các MV tham chiếu để tạo ra các MV được dự đoán. Các MV được dự đoán có thể được tính theo dự đoán dự đoán MV không gian hoặc dự đoán dự đoán MV thời gian. Sai lệch giữa các MV được dự đoán và các MV bù chuyển động (các MC MV) của khung hình hiện thời (dữ liệu chuyển động dư thừa) được mã hóa thành dòng bit 995 bởi bộ mã hóa entropy 990.

Bộ mã hóa entropy 990 mã hóa nhiều thông số và dữ liệu khác nhau thành dòng bit 995 bằng cách sử dụng các phương pháp lập mã entropy chẳng hạn mã hóa số học nhị phân thích ứng ngữ cảnh CABAC (Context-adaptive Binary Arithmetic Coding - CABAC) hoặc mã hóa Huffman. Bộ mã hóa entropy 990 mã hóa nhiều thành phần tiêu đề, nhiều cờ, cùng với các hệ số chuyển đổi đã lượng tử hóa 912, và dữ liệu chuyển

động dư thừa dưới dạng các thành phần cú pháp thành dòng bit 995. Dòng bit 995 sau đó được lưu trong thiết bị lưu trữ hoặc được truyền đến bộ giải mã thông qua các phương tiện truyền thông chẳng hạn mạng.

Bộ lọc vòng lặp kín 945 thực hiện lọc hoặc làm mượt những phần tử trên dữ liệu điểm ảnh được tái tạo 917 để giảm độ giả tạo của lập mã, đặc biệt là tại các biên giới của các khối điểm ảnh. Theo một số phương án, phép lọc được thực hiện gồm có phép giải khối hoặc phép bù thích ứng mẫu SAO (Sample Adaptive Offset - SAO). Theo một số phương án, những phép lọc bao gồm phép lọc vòng lặp thích ứng ALF (Adaptive Loop Filter - ALF).

FIG. 10 minh họa những bộ phận của bộ mã hóa video 900 mà thực thi các điều kiện ràng buộc phân chia khối. Cụ thể là, bộ phân chia 1010 tạo ra bộ các điều kiện phân chia 1020 cho bộ mã hóa entropy 990 dựa vào bộ các điều kiện ràng buộc phân cứng (ví dụ, kích cỡ VPDU) và thông tin tỷ lệ/độ méo 1015. Bộ mã hóa entropy 990 mã hóa hoặc báo hiệu bộ các điều kiện phân chia 1020 thành dòng bit 995. Các điều kiện phân chia 1020 có thể bao gồm các điều kiện mà có thể áp dụng được tại các phân mức khác nhau của phân cấp video (ví dụ, chuỗi ảnh, ảnh, slice, khối) và được báo hiệu ở các tiêu đề slice, các tiêu đề ảnh, SPS, v.v., bởi bộ mã hóa entropy 990. Các điều kiện phân chia 1020 có thể bao gồm kích cỡ khối MTT lớn nhất, kích cỡ lá QT nhỏ nhất, v.v. The bộ phân chia 1010 còn cấp cấu trúc phân chia 1030 đến môđun biến đổi 910 để cho môđun biến đổi có thể thực hiện các phép biến đổi đối với khối hiện thời của các điểm ảnh được lập mã (ví dụ, CTU) theo cấu trúc phân chia 1030.

FIG. 11 minh họa về mặt khái niệm quy trình 1100 báo hiệu các điều kiện ràng buộc phân chia khối. Theo một số phương án, một hoặc nhiều đơn vị xử lý (ví dụ, bộ xử lý) của thiết bị tính toán điều khiển bộ mã hóa 900 thực hiện quy trình 1100 bằng cách xử lý các lệnh được lưu ở phương tiện đọc được bằng máy tính. Theo một số phương án, thiết bị điện tử điều khiển bộ giải mã 900 thực hiện quy trình 1100.

Bộ mã hóa báo hiệu (tại khối 1110) tham số điều khiển phép phân chia trong dòng bit trong đó định rõ kích cỡ khối lớn nhất để sử dụng phân chia cây tam phân được ràng buộc bằng 64 hoặc nhỏ hơn.

Theo một số phương án, tham số điều khiển phép phân chia được báo hiệu ở tiêu đề slice của slice mà chứa khối hiện thời. Tham số điều khiển phép phân chia có thể được báo hiệu ở tiêu đề ảnh của ảnh hiện thời. Tham số điều khiển phép phân chia có

thể được báo hiệu theo bộ tham số chuỗi (SPS) của chuỗi các ảnh video mà chứa ảnh hiện thời. Theo một số phương án, SPS hoặc tiêu đề ảnh thiết lập giá trị mặc định cho tham số điều khiển phép phân chia có thể được ghi đè bởi tiêu đề slice mà chứa khối hiện thời.

Theo một số phương án, tham số điều khiển phép phân chia xác định kích cỡ khối lớn nhất bằng cách tham chiếu kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân, ví dụ, dưới dạng sai lệch giữa giữa (i) lôgarit cơ số 2 của kích cỡ khối lớn nhất theo đó khối có thể được phân chia tiếp bằng phân chia cây tam phân và (ii) lôgarit cơ số 2 của kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân. Kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân được ràng buộc là trị số nhỏ hơn giữa 64 và kích cỡ khối cây lập mã. Kích cỡ khối lớn nhất để sử dụng phân chia TT được ràng buộc tiếp sao cho nhỏ hơn kích cỡ khối cây lập mã.

Kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân có thể được xác định bằng cách tham chiếu kích cỡ khối lập mã nhỏ nhất, và kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân được xác định bằng cách sử dụng sai lệch giữa (i) lôgarit cơ số 2 của kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân và (ii) lôgarit cơ số 2 của kích cỡ khối lập mã nhỏ nhất.

Theo một số phương án, kích cỡ khối lớn nhất được xác định dựa vào kích cỡ của đơn vị dữ liệu đường truyền ảo (VPDU), và kích cỡ khối lớn nhất được định nghĩa dựa vào việc đảm bảo sao cho mỗi VPDU có thể được xử lý hoàn toàn bằng trạng thái đường truyền giống nhau.

Bộ mã hóa nhận (tại khối 1120) dữ liệu điểm ảnh thô sẽ được mã hóa dưới dạng khối hiện thời của ảnh hiện thời của video thành dòng bit.

Bộ mã hóa phân chia đệ quy (tại khối 1130) khối hiện thời thành một hoặc nhiều phân vùng. Bộ mã hóa hủy (tại khối 1135) phân chia TT đối với phân vùng khối hiện thời trừ khi phân vùng vừa nêu nhỏ hơn hoặc bằng kích cỡ khối lớn nhất.

Bộ mã hóa mã hóa (tại khối 1140) một hoặc nhiều phân vùng khối hiện thời thành dòng bit.

VI. Bộ giải mã video được lấy làm ví dụ

FIG. 12 minh họa bộ giải mã video được lấy làm ví dụ 1200 mà áp dụng các điều kiện ràng buộc đối với phép phân chia khối. Như được minh họa, bộ giải mã video 1200 là mạch giải mã ảnh hoặc giải mã videomà nhận dòng bit 1295 và giải mã nội

dung của dòng bit thành dữ liệu điểm ảnh của các khung hình video để hiển thị. Bộ giải mã video 1200 có một số bộ phận hoặc các môđun dùng để giải mã dòng bit 1295, bao gồm một số bộ phận được chọn từ môđun lượng tử hóa ngược 1211, môđun biến đổi ngược 1210, môđun dự đoán nội ảnh 1225, môđun bù chuyển động 1230, bộ lọc vòng lặp kín 1245, bộ đệm hình ảnh được giải mã 1250, bộ đệm MV 1265, môđun dự đoán MV 1275, và bộ phân tích 1290. Môđun bù chuyển động 1230 là một phần của môđun dự đoán liên ảnh 1240.

Theo một số phương án, các môđun từ 1210 đến 1290 là các môđun có các lệnh phần mềm được xử lý bằng một hoặc nhiều đơn vị xử lý (ví dụ, bộ xử lý) của thiết bị tính toán. Theo một số phương án, các môđun từ 1210 đến 1290 là các môđun có các mạch phần cứng được thực hiện bởi một hoặc nhiều IC của thiết bị điện tử. Mặc dù các môđun từ 1210 đến 1290 được minh họa dưới dạng các môđun riêng biệt, song một số trong số các môđun đó có thể được kết hợp thành một môđun duy nhất.

Bộ phân tích 1290 (hoặc bộ giải mã entropy) nhận dòng bit 1295 và thực hiện phép phân tích ban đầu theo cú pháp được định nghĩa bởi tiêu chuẩn lập mã video hoặc tiêu chuẩn lập mã ảnh. Thành phần cú pháp được phân tích bao gồm nhiều các thành phần tiêu đề, các cờ, cũng như dữ liệu được lượng tử hóa (hoặc các hệ số lượng tử) 1212. Bộ phân tích 1290 phân tích nhiều các thành phần cú pháp bằng cách sử dụng các phương pháp lập mã entropy chẳng hạn mã hóa số học nhị phân thích ứng ngữ cảnh CABAC (Context-adaptive Binary Arithmetic Coding - CABAC) hoặc mã hóa Huffman.

Môđun lượng tử hóa ngược 1211 giải lượng tử dữ liệu được lượng tử hóa (hoặc các hệ số được lượng tử hóa) 1212 để thu được các hệ số biến đổi, và môđun biến đổi ngược 1210 thực hiện phép biến đổi ngược đối với các hệ số biến đổi 1216 để tạo ra tín hiệu dư được tái tạo 1219. Tín hiệu dư tái tạo 1219 được cộng với dữ liệu điểm ảnh được dự đoán 1213 từ môđun dự đoán nội ảnh 1225 hoặc môđun bù chuyển động 1230 để tạo ra dữ liệu điểm ảnh được giải mã 1217. Dữ liệu điểm ảnh được giải mã được lọc bởi bộ lọc vòng lặp kín 1245 và được lưu ở bộ đệm hình ảnh được giải mã 1250. Theo một số phương án, bộ đệm hình ảnh được giải mã 1250 là bộ nhớ nằm ngoài bộ giải mã video 1200. Theo một số phương án, bộ đệm hình ảnh được giải mã 1250 là bộ nhớ nằm trong bộ giải mã video 1200.

Môđun dự đoán nội ảnh 1225 nhận dữ liệu dự đoán nội ảnh từ dòng bit 1295 và

theo đó, tạo ra dữ liệu điểm ảnh được dự đoán 1213 từ dữ liệu điểm ảnh được giải mã 1217 được lưu ở bộ đệm hình ảnh được giải mã 1250. Theo một số phương án, dữ liệu điểm ảnh được giải mã 1217 còn được lưu ở bộ đệm dòng (không được minh họa) để dự đoán nội ảnh và dự đoán MV không gian.

Theo một số phương án, nội dung của bộ đệm hình ảnh được giải mã 1250 được sử dụng để hiển thị. Thiết bị hiển thị 1255 hoặc truy xuất nội dung của bộ đệm hình ảnh được giải mã 1250 để hiển thị trực tiếp, hoặc gọi nội dung của bộ đệm hình ảnh được giải mã cho bộ đệm hiển thị. Theo một số phương án, thiết bị hiển thị nhận các giá trị điểm ảnh bộ đệm hình ảnh được giải mã 1250 thông qua bộ truyền điểm ảnh.

Môđun bù chuyển động 1230 tạo ra dữ liệu điểm ảnh được dự đoán 1213 từ dữ liệu điểm ảnh được giải mã 1217 được lưu ở bộ đệm hình ảnh được giải mã 1250 theo các MV bù chuyển động (các MC MV). Các MV bù chuyển động này được giải mã bằng cách cộng dữ liệu chuyển động dư thừa được nhận từ dòng bit 1295 với các MV được dự đoán được nhận từ môđun dự đoán MV 1275.

Môđun dự đoán MV tạo ra các MV được dự đoán dựa vào các MV tham chiếu mà được tạo ra để giải mã các khung hình video trước đó, ví dụ, các MV bù chuyển động được sử dụng để thực hiện bù chuyển động n. Môđun dự đoán MV 1275 gọi các MV tham chiếu của các khung hình video trước đó từ bộ đệm MV 1265. Bộ giải mã video 1200 lưu các MV bù chuyển động được tạo ra để giải mã khung hình video hiện thời ở bộ đệm MV 1265 thành các MV tham chiếu để tạo ra các MV được dự đoán.

Bộ lọc vòng lặp kín 1245 thực hiện các thuật toán lọc hoặc làm mượt dữ liệu điểm ảnh được giải mã 1217 để giảm độ giả tạo của lặp mã, đặc biệt là tại các biên giới của các khối điểm ảnh. Theo một số phương án, phép lọc được thực hiện gồm có phép giải khối hoặc phép bù thích ứng mẫu SAO (Sample Adaptive Offset - SAO). Theo một số phương án, những phép lọc bao gồm phép lọc vòng lặp thích ứng ALF (Adaptive Loop Filter - ALF).

FIG. 13 minh họa những bộ phận của bộ giải mã video 1200 mà thực thi các điều kiện ràng buộc phân chia khối. Cụ thể là, bộ giải mã entropy 1290 phân tích từ dòng bit 1295 các thành phần cú pháp liên quan đến việc phân chia khối và tạo ra bộ các điều kiện phân chia 1320. Các điều kiện phân chia vừa nêu có thể bao gồm các điều kiện mà có thể áp dụng được tại các phân mức khác nhau của phân cấp video (ví dụ, chuỗi ảnh, ảnh, slice, khối) và được báo hiệu ở các tiêu đề slice, các tiêu đề ảnh, SPS,

v.v. Các điều kiện phân chia 1320 có thể bao gồm kích cỡ khối MTT lớn nhất, kích cỡ lá QT nhỏ nhất, v.v. Bộ giải mã có thể hủy các dạng phân chia nhất định mà vượt quá kích cỡ khối lớn nhất theo các điều kiện phân chia 1320.

Các điều kiện phân chia được tạo ra 1320 được áp dụng cho bộ phân chia 1310, tạo ra các cấu trúc phân chia 1330 cho môđun biến đổi ngược 1210 để nó có thể thực hiện các phép biến đổi ngược đối với các phân vùng riêng (ví dụ, các CU) theo cấu trúc phân chia.

FIG. 14 minh họa về mặt khái niệm quy trình 1400 áp dụng các điều kiện ràng buộc phân chia khối. Theo một số phương án, một hoặc nhiều đơn vị xử lý (ví dụ, bộ xử lý) của thiết bị tính toán điều khiển bộ giải mã 1200 thực hiện quy trình 1400 bằng cách xử lý các lệnh được lưu ở phương tiện đọc được bằng máy tính. Theo một số phương án, thiết bị điện tử điều khiển bộ giải mã 1200 thực hiện quy trình 1400.

Bộ giải mã nhận (tại khối 1410) tham số điều khiển phép phân chia từ dòng bit trong đó định rõ kích cỡ khối lớn nhất để sử dụng phân chia cây tam phân TT (Ternary-Tree TT). Theo một số phương án, kích cỡ khối lớn nhất được ràng buộc sao cho bằng 64 hoặc nhỏ hơn.

Theo một số phương án, tham số điều khiển phép phân chia được báo hiệu ở tiêu đề slice của slice mà chứa khối hiện thời. Tham số điều khiển phép phân chia có thể được báo hiệu ở tiêu đề ảnh của ảnh hiện thời. Tham số điều khiển phép phân chia có thể được báo hiệu theo bộ tham số chuỗi (SPS) của chuỗi các ảnh video mà chứa ảnh hiện thời. Theo một số phương án, SPS hoặc tiêu đề ảnh thiết lập giá trị mặc định cho tham số điều khiển phép phân chia có thể được ghi đè bởi tiêu đề slice mà chứa khối hiện thời.

Theo một số phương án, tham số điều khiển phép phân chia xác định kích cỡ khối lớn nhất bằng cách tham chiếu kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân, ví dụ, bằng cách sử dụng sai lệch giữa (i) lôgarit cơ số 2 của kích cỡ khối lớn nhất theo đó khối có thể được phân chia tiếp bằng phân chia cây tam phân và (ii) lôgarit cơ số 2 của kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân. Kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân được ràng buộc là trị số nhỏ hơn giữa 64 và kích cỡ khối cây lập mã. Kích cỡ khối lớn nhất để sử dụng phân chia TT có thể được ràng buộc tiếp sao cho nhỏ hơn kích cỡ khối cây lập mã.

Kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân có thể được xác định bằng

cách tham chiếu kích cỡ khối lập mã nhỏ nhất, và kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân được xác định bằng cách sử dụng sai lệch giữa (i) lôgarit cơ số 2 của kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân và (ii) lôgarit cơ số 2 của kích cỡ khối lập mã nhỏ nhất.

Theo một số phương án, kích cỡ khối lớn nhất được xác định dựa vào kích cỡ của đơn vị dữ liệu đường truyền ảo (VPDU), và kích cỡ khối lớn nhất được định nghĩa dựa vào việc đảm bảo sao cho mỗi VPDU có thể được xử lý hoàn toàn bằng trạng thái đường truyền giống nhau.

Theo một số phương án, tham số điều khiển phép phân chia được báo hiệu ở tiêu đề slice của slice mà chứa khối hiện thời. Tham số điều khiển phép phân chia có thể được báo hiệu ở tiêu đề ảnh của ảnh hiện thời. Tham số điều khiển phép phân chia có thể được báo hiệu theo bộ tham số chuỗi (SPS) của chuỗi các ảnh video mà chứa ảnh hiện thời. Theo một số phương án, SPS hoặc tiêu đề ảnh thiết lập giá trị mặc định cho tham số điều khiển phép phân chia có thể được ghi đè bởi tiêu đề slice mà chứa khối hiện thời.

Bộ giải mã nhận (tại khối 1420) dữ liệu từ dòng bit cho khối các điểm ảnh sẽ được giải mã thành khối hiện thời của ảnh hiện thời của video.

Bộ giải mã phân chia đệ quy (tại khối 1430) khối hiện thời thành một hoặc nhiều phân vùng. Bộ giải mã hủy (tại khối 1435) phân chia TT đối với phân vùng khối hiện thời trừ khi phân vùng vừa nêu nhỏ hơn hoặc bằng kích cỡ khối lớn nhất.

Bộ giải mã khôi phục (tại khối 1440) một hoặc nhiều phân vùng khối hiện thời.

Theo một số phương án, bộ mã hóa có thể báo hiệu (hoặc tạo ra) một hoặc nhiều thành phần cú pháp trong dòng bit, sao cho bộ giải mã có thể phân tích một hoặc nhiều thành phần cú pháp vừa nêu từ dòng bit.

VII. Hệ thống điện tử được lấy làm ví dụ

Rất nhiều các dấu hiệu và ứng dụng được mô tả ở trên được thực hiện dưới dạng các xử lý phần mềm mà được chỉ dẫn dưới dạng tập hợp các lệnh được ghi vào phương tiện lưu trữ có thể đọc được bằng máy tính (còn được gọi là phương tiện có thể đọc được bằng máy tính). Khi các lệnh này được xử lý bởi một hoặc nhiều bộ xử lý hoặc bộ tính toán (ví dụ, một hoặc nhiều bộ xử lý, lõi của các bộ xử lý, hoặc các bộ xử lý khác), chúng giúp cho bộ xử lý (các bộ xử lý) thực hiện những chức năng được chỉ thị trong những câu lệnh. Những ví dụ về các phương tiện đọc được bằng máy tính

bao gồm, nhưng không bị giới hạn chỉ ở, các đĩa CD-ROM, các ổ đĩa nhanh, các vi mạch bộ nhớ ngẫu nhiên RAM (Random-Access Memory - RAM), các ổ cứng, các bộ nhớ chỉ đọc chương trình có thể xóa được (Erasable Programmable Read Only Memory - EPROM), các bộ nhớ chỉ đọc chương trình có thể xóa được bằng điện (Electrically Erasable Programmable Read-Only Memories - EEPROM), .v.v. Các phương tiện đọc được bằng máy tính không bao gồm các sóng truyền thông và các tín hiệu điện truyền kết nối có dây hoặc không dây.

Ở bản mô tả này, thuật ngữ “phần mềm” được hiểu là bao gồm phần vi chương trình nằm trong bộ nhớ chỉ đọc hoặc các ứng dụng được lưu được lưu trong bộ nhớ từ mà có thể có thể được ghi vào bộ nhớ để xử lý bởi khối xử lý. Tương tự, theo một số phương án, nhiều phát minh phần mềm có thể còn được sử dụng dưới dạng các chương trình con của một chương trình lớn hơn trong khi giữ lại các kỹ thuật phần mềm riêng. Theo một số phương án, nhiều kỹ thuật phần mềm có thể cũng được sử dụng dưới dạng các chương trình riêng biệt. Cuối cùng, bất kỳ sự kết hợp nào giữa các chương trình riêng biệt mà cùng thực hiện kỹ thuật phần mềm được mô tả ở đây đều nằm trong phạm vi của sáng chế. Theo một số phương án, các chương trình phần mềm, khi được cài đặt để vận hành một hoặc nhiều hệ thống điện tử, định ra một hoặc nhiều thiết bị thực thi chuyên biệt để xử lý và thực hiện các phép toán của các chương trình phần mềm.

FIG. 15 minh họa về mặt ý tưởng hệ thống điện tử 1500 qua đó một số phương án của sáng chế được thực hiện. Hệ thống điện tử 1500 có thể là máy tính (ví dụ, máy tính để bàn, máy tính cá nhân, máy tính bảng, .v.v), điện thoại, PDA, hoặc bất kỳ loại thiết bị điện tử nào. Hệ thống điện tử vừa nêu gồm có nhiều loại phương tiện có thể đọc được bằng máy tính và nhiều giao diện dùng cho nhiều loại phương tiện có thể đọc được bằng máy tính đó. Hệ thống điện tử 1500 bao gồm bus 1505, bộ xử lý (các bộ xử lý) 1510, bộ xử lý đồ họa GPU (Graphic Processing Unit - GPU) 1515, bộ nhớ hệ thống 1520, mạng 1525, bộ nhớ chỉ đọc 1530, thiết bị lưu trữ cố định 1535, các thiết bị nhập 1540, và các thiết bị xuất 1545.

Bus 1505 là thuật ngữ chỉ tập hợp toàn bộ hệ thống, thiết bị ngoại vi, và các bus vi mạch mà kết nối truyền thông nhiều thiết bị nội bộ của hệ thống điện tử 1500. Ví dụ, bus 1505 kết nối truyền thông bộ xử lý (các bộ xử lý) 1510 với GPU 1515, bộ nhớ chỉ đọc 1530, bộ nhớ hệ thống 1520, và thiết bị lưu trữ cố định 1535.

Từ những bộ nhớ vừa nêu, bộ xử lý (các bộ xử lý) 1510 gọi ra các lệnh xử lý và dữ liệu nhằm xử lý các quá trình xử lý của sáng chế. Bộ xử lý (các bộ xử lý) có thể là bộ xử lý đơn nhân hoặc bộ xử lý đa nhân theo các phương án khác nhau. Một số lệnh được phân tích và được xử lý bởi GPU 1515. GPU 1515 có thể có thể nạp ngoại tuyến nhiều phép tính toán hoặc bổ sung phép xử lý ảnh được cấp bởi bộ xử lý (các bộ xử lý) 1510.

Bộ nhớ chỉ đọc ROM (Read Only Memory - ROM) 1530 lưu các dữ liệu cố định và các lệnh được sử dụng bởi bộ xử lý (các bộ xử lý) 1510 và các môđun khác của hệ thống điện tử. Mặt khác, thiết bị lưu trữ cố định 1535 là thiết bị nhớ đọc và ghi được. Thiết bị này là bộ nhớ không khả biến để lưu các lệnh và dữ liệu ngay cả khi hệ thống điện tử 1500 không hoạt động. Một số phương án của sáng chế sử dụng bộ nhớ dung lượng cao (chẳng hạn đĩa từ hoặc đĩa quang và ổ đĩa tương ứng của nó) làm thiết bị lưu trữ cố định 1535.

Các phương án khác sử dụng thiết bị lưu di động (chẳng hạn đĩa mềm, thiết bị nhớ cực nhanh, .v.v., và ổ đĩa tương ứng của nó) làm thiết bị lưu trữ cố định. Tương tự thiết bị lưu trữ cố định 1535, bộ nhớ hệ thống 1520 là thiết bị nhớ đọc và ghi được. Tuy nhiên, không giống thiết bị lưu trữ 1535, bộ nhớ hệ thống 1520 là bộ nhớ đọc và ghi khả biến, chẳng hạn bộ nhớ truy cập ngẫu nhiên. Bộ nhớ hệ thống 1520 lưu một số câu lệnh và dữ liệu mà bộ xử lý cần để khi vận hành. Theo một số phương án, những quy trình xử lý của sáng chế được lưu trong bộ nhớ hệ thống 1520, thiết bị lưu trữ cố định 1535, và/hoặc bộ nhớ chỉ đọc 1530. Ví dụ, nhiều bộ nhớ khác nhau chứa các lệnh để xử lý các đoạn phim đa phương tiện theo một số phương án. Từ những bộ nhớ vừa nêu, bộ xử lý (các bộ xử lý) 1510 gọi ra các lệnh xử lý và dữ liệu nhằm xử lý các quá trình xử lý của một số phương án.

Bus 1505 còn kết nối với các thiết bị nhập và xuất 1540 và 1545. Các thiết bị nhập 1540 cho phép người sử dụng truyền thông tin và chọn các lệnh cho hệ thống điện tử. Các thiết bị nhập 1540 bao gồm các bàn phím chữ-số và các thiết bị trỏ (còn được gọi “các thiết bị điều khiển con trỏ”), các máy quay (ví dụ, các webcam), các ống nói hoặc các thiết bị tương tự để nhận các lệnh âm thanh, v.v. Các thiết bị xuất 1545 hiển thị các hình ảnh được tạo ra bởi hệ thống điện tử hoặc nếu không thì hiển thị dữ liệu đầu ra. Các thiết bị xuất 1545 bao gồm các máy in và các thiết bị hiển thị, chẳng hạn các màn hình ống tia catốt CRT (Cathode Ray Tube - CRT) hoặc màn hình tinh

thể lỏng (Liquid Crystal Display - LCD), cũng như các loại loa hoặc các thiết bị xuất âm thanh tương tự. Một số phương án bao gồm các thiết bị chẳng hạn màn hình cảm ứng có chức năng làm cả thiết bị nhập và thiết bị xuất.

Cuối cùng, như được thể hiện trên FIG. 15, bus 1505 còn kết nối hệ thống điện tử 1500 vào mạng 1525 thông qua các mạng (không được thể hiện). Theo cách thức vừa nêu, máy tính có thể làm một thành phần của mạng máy tính (chẳng hạn mạng máy tính cục bộ LAN (“Local Area Network - LAN”), mạng diện rộng (“Wide Area Network - WAN”), hoặc mạng nội bộ Intranet, hoặc mạng của các mạng, chẳng hạn mạng Internet. Bất kỳ hoặc tất cả các bộ phận của hệ thống điện tử 1500 có thể được sử dụng theo sáng chế.

Một số phương án bao gồm các thiết bị điện tử, chẳng hạn các bộ vi xử lý, bộ lưu trữ và bộ nhớ để lưu các lệnh chương trình máy tính trong phương tiện có thể đọc được bằng máy hoặc có thể đọc được bằng máy tính (theo cách khác còn được gọi là các phương tiện có thể đọc được bằng máy tính, các phương tiện có thể đọc được bằng máy, hoặc các phương tiện lưu trữ có thể đọc được bằng máy). Một số ví dụ về phương tiện có thể đọc được bằng máy tính bao gồm RAM, ROM, các đĩa nén chỉ đọc (CD-ROM), các đĩa nén có thể ghi (CD-R), các đĩa nén có thể ghi lại (CD-RW), các đĩa đa năng số DVD chỉ đọc (ví dụ, đĩa DVD-ROM, đĩa hai lớp DVD-ROM), nhiều loại đĩa DVD có thể ghi/ghi lại (ví dụ, đĩa DVD-RAM, đĩa DVD-RW, đĩa DVD+RW, v.v), bộ nhớ cực nhanh (ví dụ, các thẻ SD, các thẻ mini-SD, các thẻ micro-SD, v.v), các ổ cứng từ và/hoặc các ổ cứng trạng thái rắn, các đĩa Blu-Ray® chỉ đọc và có thể ghi được, các đĩa quang siêu mật độ, bất kỳ phương tiện từ hoặc quang khác, và các đĩa mềm. Phương tiện có thể đọc được bằng máy tính có thể lưu chương trình máy tính mà có thể xử lý được bởi ít nhất một bộ xử lý và gồm có bộ các lệnh dùng để thực hiện nhiều phép tính khác nhau. Các ví dụ về các chương trình máy tính hoặc mã máy tính bao gồm mã máy, chẳng hạn được tạo ra bởi trình biên dịch, và các tệp tin bao gồm mã bậc cao được xử lý bởi máy tính, thiết bị điện tử, hoặc bộ vi xử lý bằng cách sử dụng trình biên dịch.

Mặc dù phần trình bày ở trên chủ yếu đề cập đến bộ vi xử lý hoặc các bộ vi xử lý đa nhân để xử lý phần mềm, song rất nhiều các dấu hiệu và ứng dụng được mô tả ở trên được thực hiện bởi một hoặc nhiều vi mạch tích hợp, chẳng hạn các vi mạch tích hợp chuyên dụng ASIC (Application Specific Integrated Circuit - ASIC) hoặc vi mạch

tích hợp FPGA (Field Programmable Gate Array - FPGA). Theo một số phương án, các mạch tích hợp vừa nêu xử lý các lệnh được lưu trên chính mạch đó. Ngoài ra, một số phương án xử lý phần mềm được lưu trong các thiết bị logic lập trình được PLD (Programmable Logic Device - PLD), ROM, hoặc RAM.

Như được sử dụng ở bản mô tả này và bất kỳ điểm yêu cầu bảo hộ của sáng chế, các thuật ngữ “máy tính”, “máy chủ”, “bộ xử lý”, và “bộ nhớ” toàn bộ chúng đề cập đến các thiết bị điện tử hoặc các thiết bị công nghệ khác. Những thuật ngữ vừa nêu không bao gồm người hoặc nhóm người. Nhằm những mục đích của bản mô tả này, các thuật ngữ hiển thị hoặc biểu thị mang nghĩa là hiển thị trên thiết bị điện tử. Như được sử dụng ở bản mô tả này và bất kỳ điểm yêu cầu bảo hộ của sáng chế, các thuật ngữ “phương tiện đọc được bằng máy tính,” “các phương tiện đọc được bằng máy tính,” và “phương tiện đọc được bằng máy” tất cả đều được giới hạn là các đối tượng vật lý có tính hữu hình mà lưu thông tin dưới dạng có thể đọc được bằng máy tính. Những thuật ngữ này không bao gồm bất kỳ tín hiệu truyền không dây, tín hiệu truyền theo dây, và bất kỳ tín hiệu nào khác.

Mặc dù sáng chế vừa được mô tả theo nhiều phương án chi tiết cụ thể, song một người trình độ trung bình trong lĩnh vực kỹ thuật này sẽ nhận ra rằng sáng chế có thể được thực hiện theo các cách thức cụ thể khác nhưng không trệch khỏi nội dung của sáng chế. Ngoài ra, số lượng cách hình vẽ (bao gồm FIG. 11 và FIG. 14) minh họa về mặt khái niệm các quy trình. Các phần tử cụ thể của những quy trình này có thể không được thực hiện được thực hiện theo thứ tự chính xác như được thể hiện và được mô tả. Những phần tử cụ thể đó có thể không được thực hiện theo chuỗi liên tiếp của các phần tử, và những phần tử cụ thể khác có thể được thực hiện in các phương án khác. Ngoài ra, quy trình vừa nêu có thể được thực hiện bằng cách sử dụng một số quy trình con, hoặc một phần của quy trình macrô lớn hơn. Do đó, một người có kiến thức trung bình trong lĩnh vực kỹ thuật này sẽ hiểu rằng sáng chế không bị giới hạn bởi phần mô tả chi tiết minh họa được trình bày ở trên, mà sáng chế được định rõ theo những điểm yêu cầu bảo hộ đính kèm.

Những lưu ý phụ

Đối tượng được mô tả ở bản mô tả này đôi khi minh họa các thành phần khác chứa trong, hoặc được kết nối với, các thành phần khác nữa. Sẽ được hiểu rằng những cấu trúc được mô tả như vậy chỉ là các ví dụ, và rằng trong thực tế rất nhiều cấu

trúc khác có thể được thực hiện để đạt được chức năng tương tự. Về mặt khái niệm, bất kỳ sự bố trí nào của các thành phần nhằm đạt được chức năng tương tự đều “được kết hợp” hiệu quả. Vì vậy, bất kỳ hai thành phần nào ở đây được phối hợp nhằm đạt được chức năng cụ thể đều có thể được xem là “được kết hợp” với nhau sao cho chức năng mong muốn đạt được, bất kể đến các cấu trúc hoặc các thành phần trung gian. Tương tự, bất kỳ hai thành phần nào được kết hợp như vậy có thể còn được coi là “được kết nối hoạt động được”, hoặc “được liên kết hoạt động được”, với nhau nhằm đạt được chức năng mong muốn, và bất kỳ hai thành phần nào có khả năng được kết hợp như vậy có thể còn được coi là “có thể liên kết hoạt động được”, với nhau nhằm đạt được chức năng mong muốn. Các ví dụ cụ thể về liên kết hoạt động được bao gồm nhưng không bị giới hạn chỉ ở các thành phần có thể liên kết vật lý với nhau và/hoặc có thể tương tác vật lý với nhau và/hoặc các thành phần có thể tương tác không dây với nhau và/hoặc có thể liên kết không dây với nhau và/hoặc các thành phần có thể tương tác logic với nhau và/hoặc có thể liên kết logic với nhau.

Ngoài ra, liên quan đến việc sử dụng bất kỳ các thuật ngữ mang nghĩa số nhiều và/hoặc số ít ở bản mô tả này, những người có chuyên môn trong cùng lĩnh vực kỹ thuật này có thể dịch từ số nhiều sang số ít và/hoặc từ số ít sang số nhiều miễn là phù hợp với ngữ cảnh và/hoặc ứng dụng. Nhiều phép hoán vị số ít/số nhiều khác nhau có thể được trình bày rõ ràng ở đây nhằm giúp cho hiểu rõ.

Ngoài ra, sẽ được hiểu bởi những chuyên gia trong cùng lĩnh vực kỹ thuật này rằng, nhìn chung, các thuật ngữ được sử dụng ở bản mô tả này, và đặc biệt là trong các điểm yêu cầu bảo hộ đính kèm, ví dụ, các phần chính của các điểm yêu cầu bảo hộ đính kèm, thường được hiểu là các thuật ngữ “mở”, ví dụ, thuật ngữ “bao gồm” nên được hiểu dưới dạng “bao gồm nhưng không bị giới hạn chỉ ở,” thuật ngữ “có” nên được hiểu dưới dạng “có ít nhất,” thuật ngữ “gồm có” nên được hiểu dưới dạng “gồm có nhưng không bị giới hạn chỉ có,” v.v. Sẽ được hiểu rõ hơn bởi những người làm trong cùng lĩnh vực kỹ thuật này rằng nếu có ý định miêu tả số lượng chính xác các điểm yêu cầu bảo hộ, thì ý định đó sẽ được nêu rõ trong phần yêu cầu bảo hộ, và nếu không có miêu vừa nêu thì không có ý định trên. Ví dụ, như một cách để hiểu, các điểm yêu cầu bảo hộ đính kèm dưới đây có thể chứa các cụm từ giới thiệu sau “ít nhất một” và “một hoặc nhiều” để miêu tả yêu cầu bảo hộ. Tuy nhiên, việc sử dụng các cụm từ như vậy sẽ không được hiểu là ám chỉ rằng việc miêu tả yêu cầu bảo hộ bằng

mạo từ không xác định “một” sẽ giới hạn bất kỳ yêu cầu bảo hộ cụ thể nào đều chứa phần mô tả yêu cầu bảo hộ được giới thiệu đối với các phương án thực hiện chỉ chứa một miêu tả như vậy, ngay cả khi yêu cầu bảo hộ giống nhau gồm có các cụm từ giới thiệu “một hoặc nhiều” hoặc “ít nhất một” và mạo từ không xác định chẳng hạn “một” nên được hiểu là có nghĩa “ít nhất một” hoặc “một hoặc nhiều;” điều tương tự cũng đúng khi sử dụng các mạo từ xác định để mô tả các yêu cầu bảo hộ. Ngoài ra, thậm chí nếu số lượng chính xác yêu cầu bảo hộ được liệt kê rõ ràng thì những chuyên gia trong cùng lĩnh vực kỹ thuật này sẽ nhận ra rằng số lượng yêu cầu bảo hộ như vậy sẽ được hiểu là mang nghĩa là giá trị tối thiểu số được liệt kê, ví dụ, liệt kê nguyên bản là “hai liệt kê,” mà không có sửa đổi khác, có nghĩa là ít nhất hai liệt kê, hoặc hai hoặc nhiều hơn hai liệt kê. Ngoài ra, ở những trường hợp trong đó thuật ngữ quy ước “ít nhất một trong số A, B, và C, v.v.” được sử dụng, nhìn chung cách sử dụng thuật ngữ như vậy về mặt ý nghĩa sẽ được một người có kiến thức trong lĩnh vực kỹ thuật này hiểu thuật ngữ quy ước đó, ví dụ, “hệ thống có ít nhất một trong số A, B, và C” sẽ bao gồm nhưng không bị giới hạn chỉ ở các hệ thống có chỉ riêng A, chỉ riêng B, chỉ riêng C, A và B cùng nhau, A và C cùng nhau, B và C cùng nhau, và/hoặc A, B, và C cùng nhau, v.v. Ở những trường hợp trong đó thuật ngữ quy ước “ít nhất một trong số A, B, hoặc C, v.v.” được sử dụng, nhìn chung cách sử dụng thuật ngữ như vậy về mặt ý nghĩa sẽ được một người có kiến thức trong lĩnh vực kỹ thuật này hiểu thuật ngữ quy ước đó, ví dụ, “hệ thống có ít nhất một trong số A, B, hoặc C” sẽ bao gồm nhưng không bị giới hạn chỉ ở các hệ thống có chỉ riêng A, chỉ riêng B, chỉ riêng C, A và B cùng nhau, A và C cùng nhau, B và C cùng nhau, và/hoặc A, B, và C cùng nhau, v.v. Sẽ được hiểu rõ hơn bởi những người làm trong cùng lĩnh vực kỹ thuật này rằng hầu như bất kỳ từ và/hoặc cụm từ phân biệt diễn đạt hai hoặc nhiều hơn hai thuật ngữ có thể chọn khả năng, liệu trong phần mô tả, các điểm yêu cầu bảo hộ, hoặc các hình vẽ, sẽ được hiểu để suy ra những khả năng bao gồm một trong số các thuật ngữ, hoặc một trong các thuật ngữ đó, hay bao gồm cả hai thuật ngữ đó. Ví dụ, cụm từ “A hoặc B” sẽ được hiểu là bao gồm những khả năng sau “A” hoặc “B” hoặc “A và B”.

Từ những gì được trình bày ở trên, sẽ được hiểu rằng nhiều phương án thực hiện sáng chế được mô tả ở đây nhằm các mục đích minh họa, và rằng nhiều sửa đổi có thể được thực hiện mà không trệch khỏi phạm vi và nội dung của sáng chế. Theo đó, các phương án thực hiện khác nhau được bộc lộ ở bản mô tả này không nhằm giới hạn

sáng chế, và phạm vi và nội dung sáng chế được định rõ bởi các điểm yêu cầu bảo hộ dưới đây.

Yêu cầu bảo hộ

1. Phương pháp giải mã video bao gồm:

nhận tham số điều khiển phép phân chia từ dòng bit trong đó định rõ kích cỡ khối lớn nhất để sử dụng phân chia cây tam phân TT được ràng buộc bằng 64 hoặc nhỏ hơn;

nhận dữ liệu từ dòng bit cho khối các điểm ảnh sẽ được giải mã thành khối hiện thời của ảnh hiện thời của video;

phân chia khối hiện thời thành một hoặc nhiều phân vùng theo cách đệ quy, trong đó phân chia TT được hủy bỏ đối với phân vùng khối hiện thời trừ khi phân vùng vừa nêu nhỏ hơn hoặc bằng kích cỡ khối lớn nhất; và

khôi phục một hoặc nhiều phân vùng khối hiện thời.

2. Phương pháp giải mã video theo điểm 1, trong đó tham số điều khiển phép phân chia xác định kích cỡ khối lớn nhất bằng cách tham chiếu kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân.

3. Phương pháp giải mã video theo điểm 2, trong đó tham số điều khiển phép phân chia được xác định bằng cách sử dụng sai lệch giữa (i) lôgarit cơ số 2 của kích cỡ khối lớn nhất theo đó khối có thể được phân chia tiếp bằng phân chia cây tam phân và (ii) lôgarit cơ số 2 của kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân.

4. Phương pháp giải mã video theo điểm 2, trong đó kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân được xác định bằng cách tham chiếu kích cỡ khối lập mã nhỏ nhất.

5. Phương pháp giải mã video theo điểm 4, trong đó kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân được xác định bằng cách sử dụng sai lệch giữa (i) lôgarit cơ số 2 của kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân và (ii) lôgarit cơ số 2 của kích cỡ khối lập mã nhỏ nhất.

6. Phương pháp giải mã video theo điểm 2, trong đó kích cỡ nhỏ nhất của khối lá phân chia cây tứ phân được ràng buộc là trị số nhỏ hơn giữa 64 và kích cỡ khối cây lập mã.

7. Phương pháp giải mã video theo điểm 1, trong đó kích cỡ khối lớn nhất để sử dụng phân chia TT được ràng buộc tiếp sao cho nhỏ hơn kích cỡ khối cây lập mã.

8. Phương pháp giải mã video theo điểm 1, kích cỡ khối lớn nhất được xác định dựa vào kích cỡ của đơn vị dữ liệu đường truyền ảo (VPDU).

9. Phương pháp giải mã video theo điểm 7, kích cỡ khối lớn nhất được định nghĩa dựa vào việc đảm bảo sao cho mỗi VPDU có thể được xử lý hoàn toàn bằng trạng thái

đường truyền giống nhau.

10. Phương pháp giải mã video theo điểm 1, trong đó tham số điều khiển phép phân chia được báo hiệu ở tiêu đề slice của slice mà chứa khối hiện thời.

11. Phương pháp giải mã video theo điểm 1, trong đó tham số điều khiển phép phân chia được báo hiệu ở tiêu đề ảnh của ảnh hiện thời.

12. Phương pháp giải mã video theo điểm 1, trong đó tham số điều khiển phép phân chia được báo hiệu theo bộ tham số chuỗi (SPS) của chuỗi các ảnh video mà chứa ảnh hiện thời.

13. Phương pháp mã hóa video bao gồm:

báo hiệu tham số điều khiển phép phân chia trong dòng bit trong đó định rõ kích cỡ khối lớn nhất để sử dụng phân chia tam phân TT được ràng buộc bằng 64 hoặc nhỏ hơn;

nhận dữ liệu điểm ảnh thô sẽ được mã hóa dưới dạng khối hiện thời của ảnh hiện thời của video thành dòng bit;

phân chia khối hiện thời thành một hoặc nhiều phân vùng, trong đó phân chia TT được hủy bỏ đối với phân vùng khối hiện thời trừ khi phân vùng vừa nêu nhỏ hơn hoặc bằng kích cỡ khối lớn nhất; và

mã hóa một hoặc nhiều phân vùng khối hiện thời thành dòng bit.

14. Thiết bị điện tử bao gồm:

mạch bộ giải mã video được cấu hình để thực hiện các chức năng sau:

nhận tham số điều khiển phép phân chia từ dòng bit trong đó định rõ kích cỡ khối lớn nhất để sử dụng phân chia tam phân TT được ràng buộc bằng 64 hoặc nhỏ hơn;

nhận dữ liệu từ dòng bit cho khối các điểm ảnh sẽ được giải mã thành khối hiện thời của ảnh hiện thời của video;

phân chia khối hiện thời thành một hoặc nhiều phân vùng, trong đó phân chia BT được hủy bỏ đối với phân vùng khối hiện thời trừ khi phân vùng vừa nêu nhỏ hơn hoặc bằng kích cỡ khối lớn nhất; và

khôi phục một hoặc nhiều phân vùng khối hiện thời.

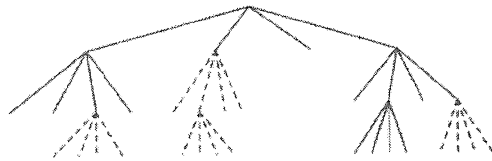
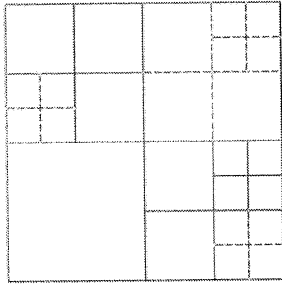
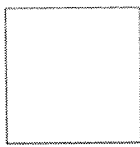
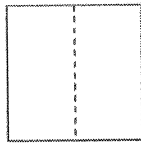
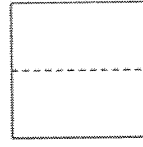
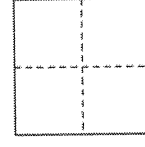
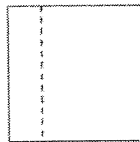
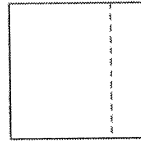
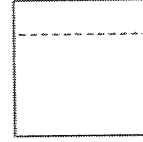
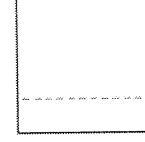
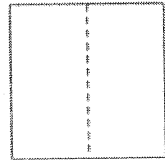
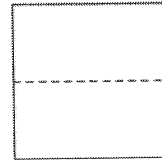
FIG. 1**FIG. 2** $M \times M$  $M/2 \times M$  $M \times M/2$  $M/2 \times M/2$  $M/4 \times M (L)$  $M/4 \times M (R)$  $M \times M/4 (U)$  $M \times M/4 (D)$

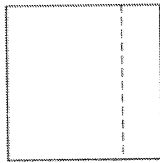
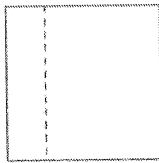
FIG. 3



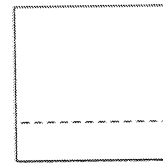
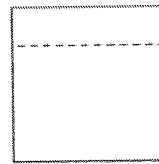
Phân chia đứng
đối xứng



Phân chia ngang
đối xứng



Phân chia đứng
không đối xứng



Phân chia ngang
không đối xứng

FIG. 4

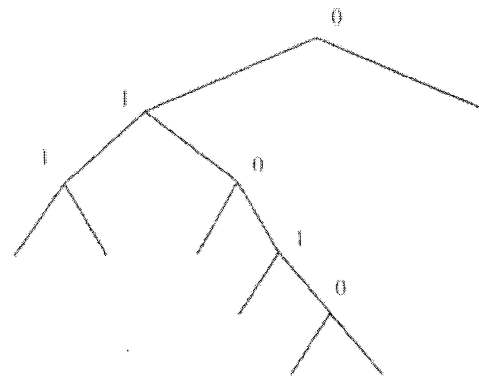
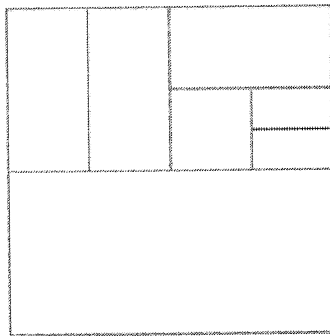
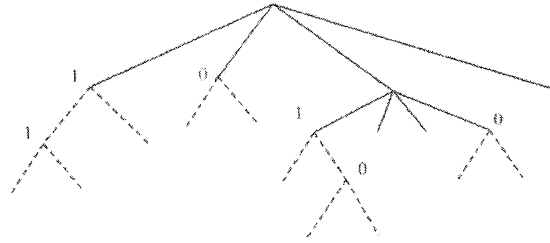
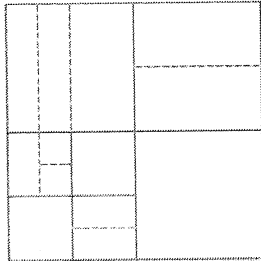
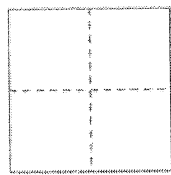
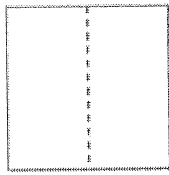
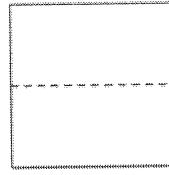


FIG. 5**FIG. 6**

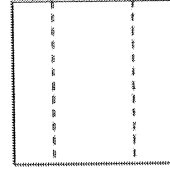
(a)



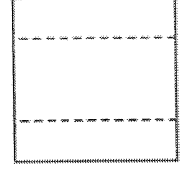
(b)



(c)



(d)



(e)

FIG. 7

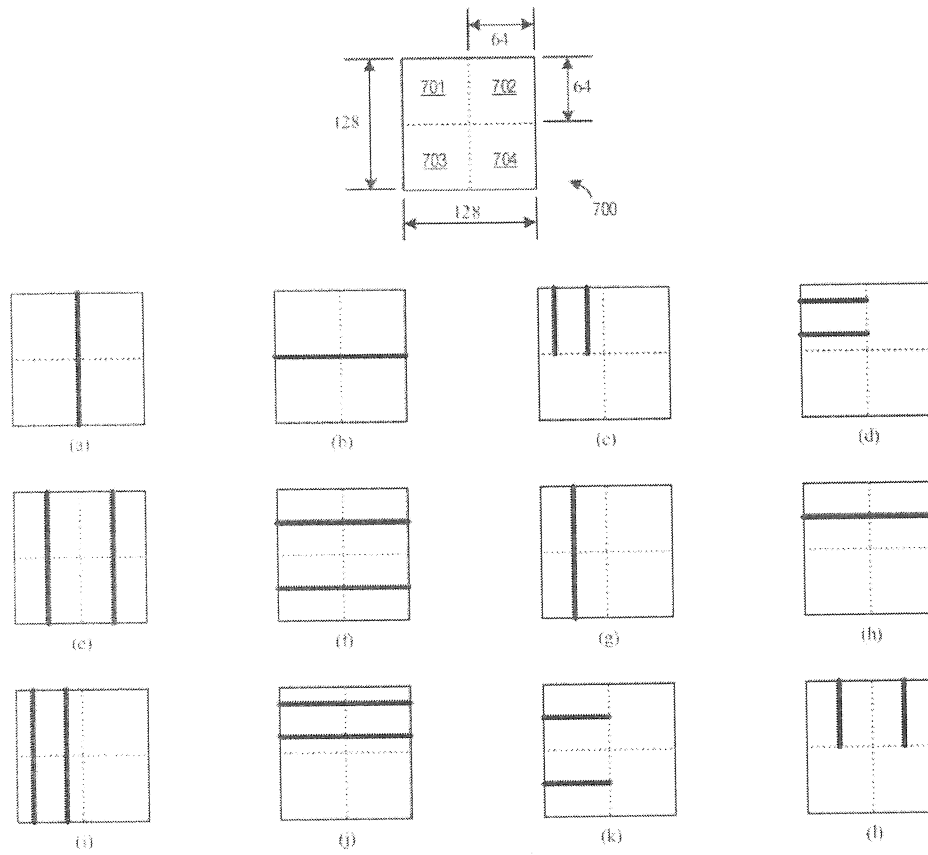


FIG. 8 Kích cỡ cây tam
phân lớn nhất = 64
Kích cỡ cây nhị
phân lớn nhất = 64

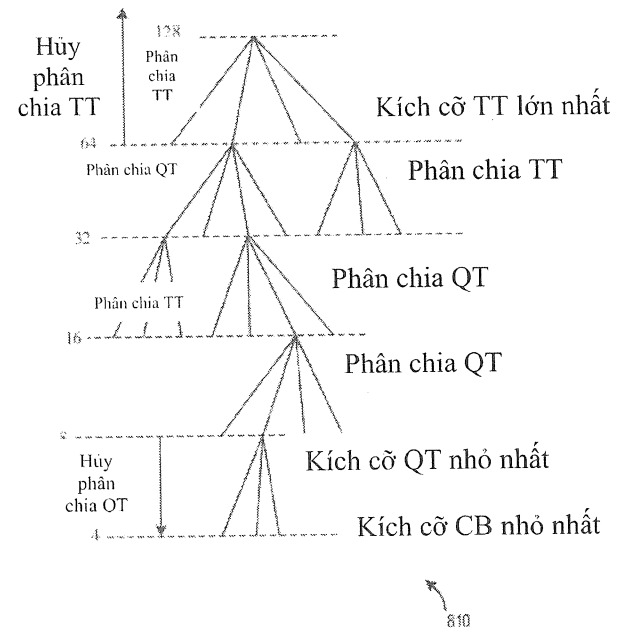
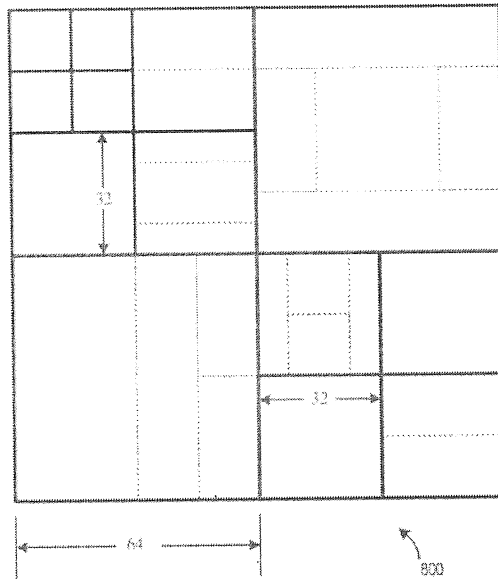


FIG. 9

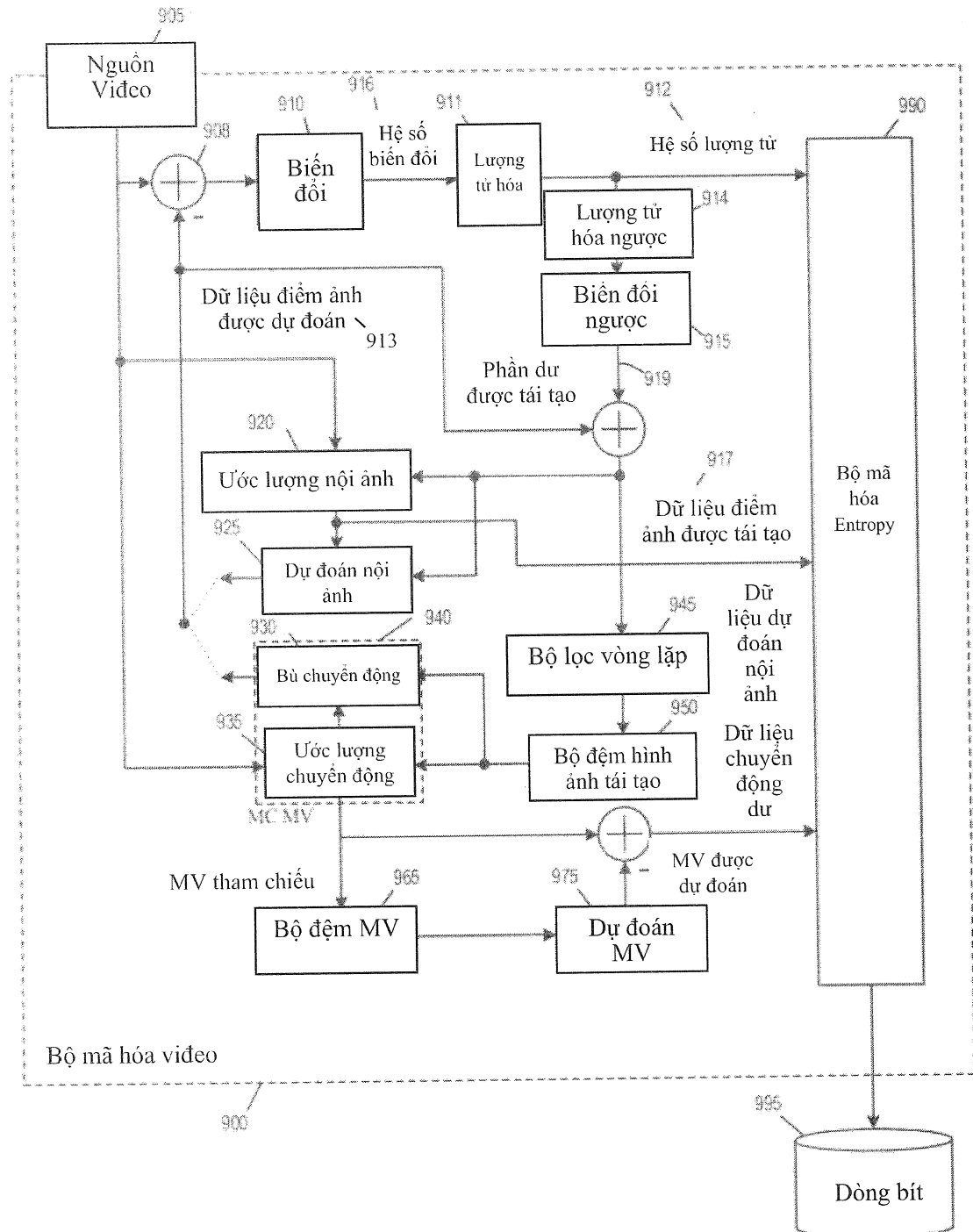


FIG. 10

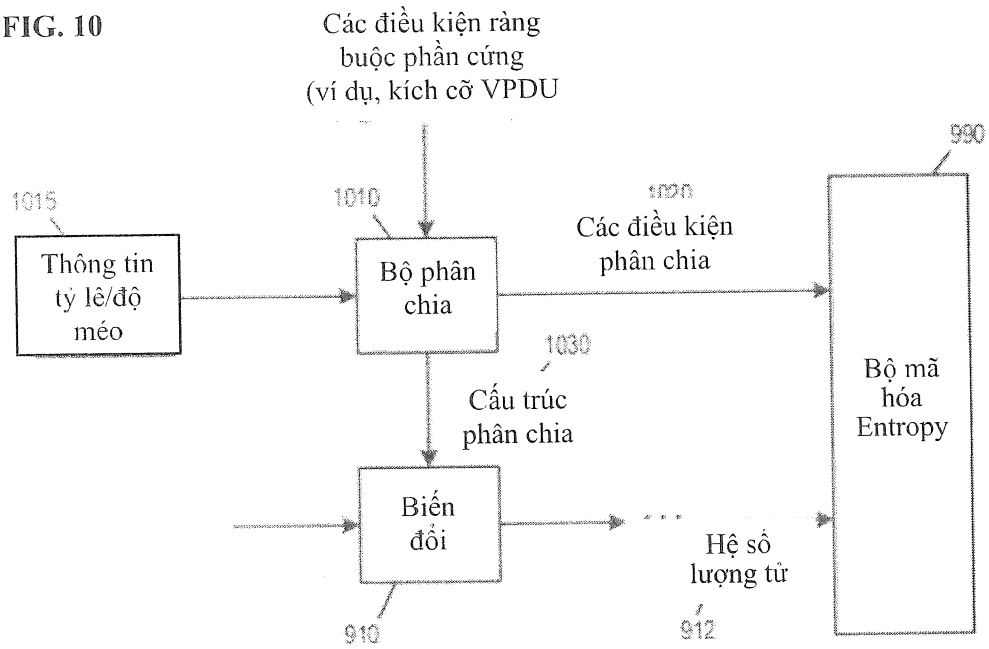


FIG. 11

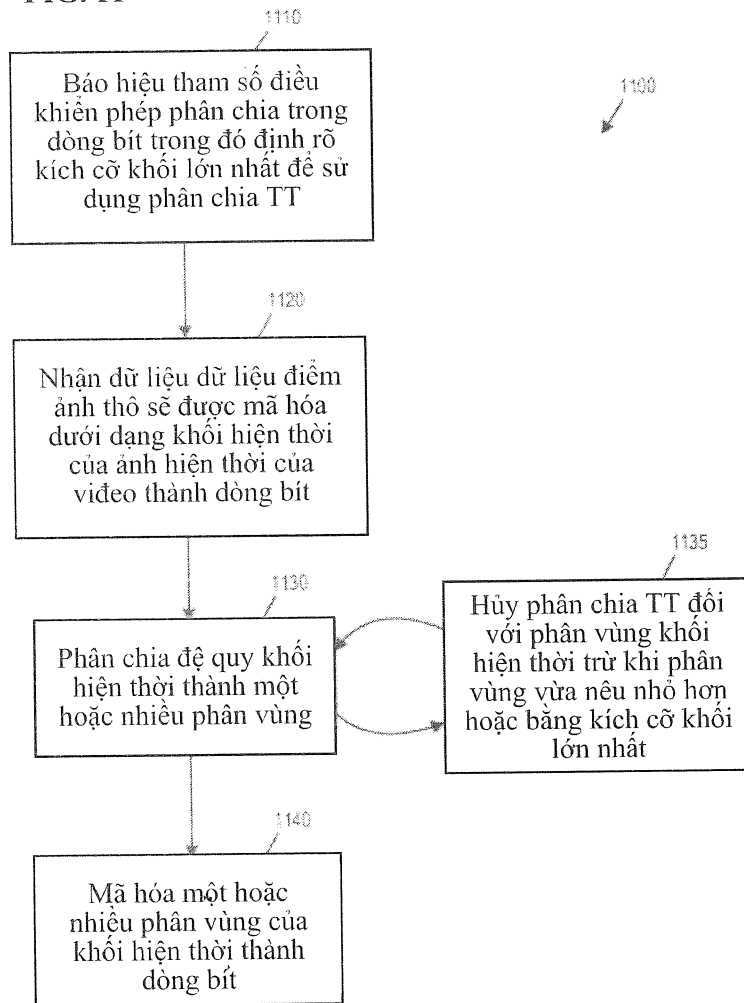


FIG. 12

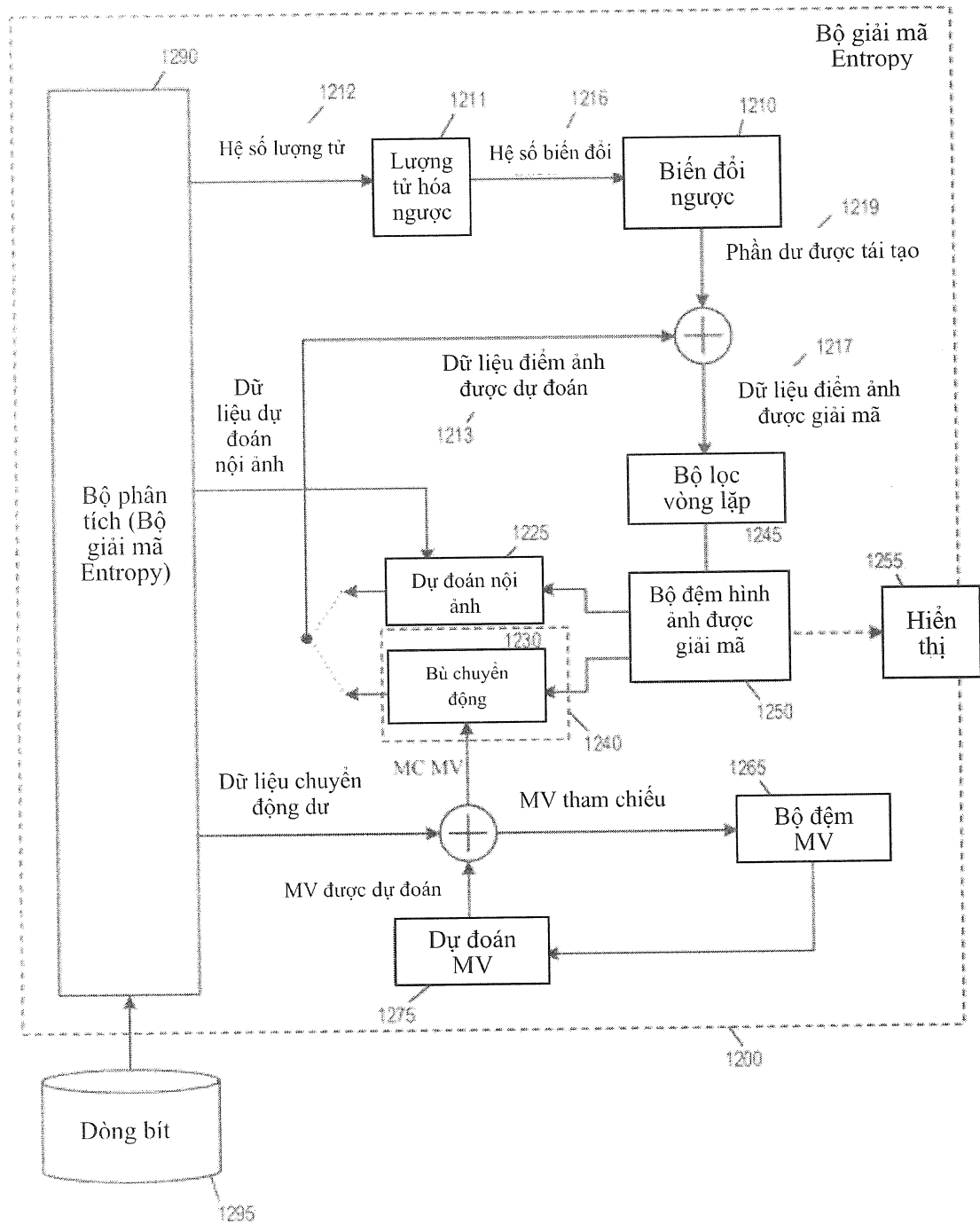


FIG. 13

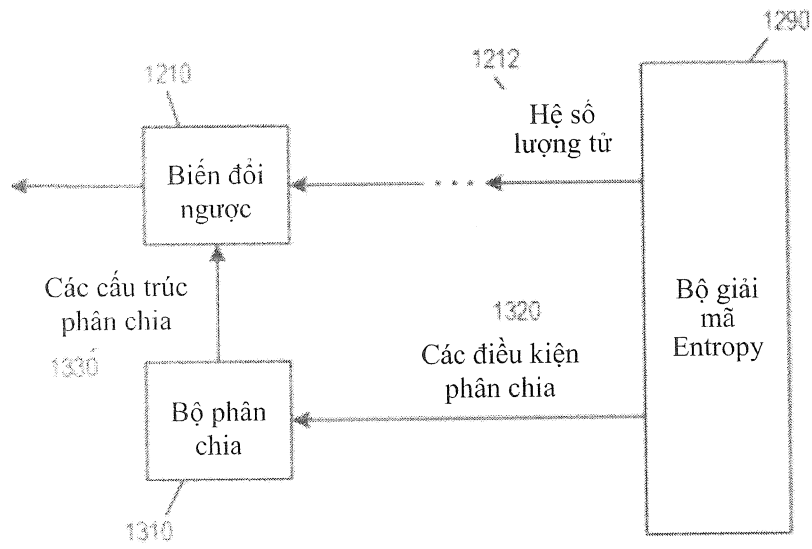


FIG. 14

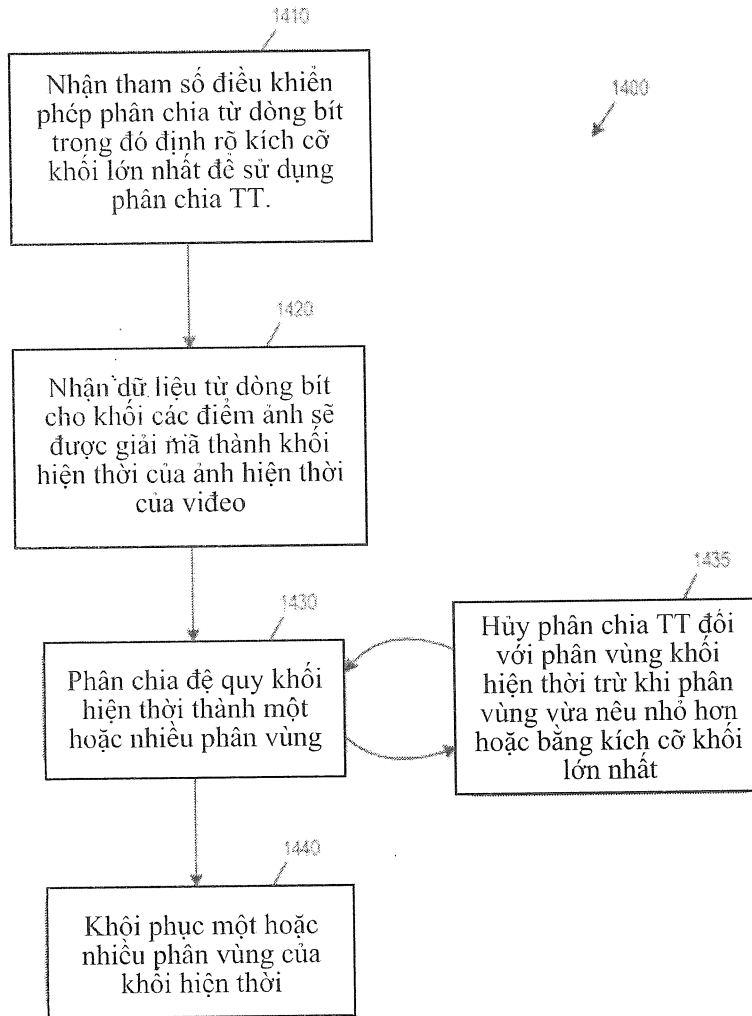


FIG. 15

