



(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)



1-0050474

(51)^{2020.01} H04N 19/105

(13) B

(21) 1-2022-00166

(22) 09/07/2020

(86) PCT/CN2020/100992 09/07/2020

(87) WO2021/004495 14/01/2021

(30) PCT/CN2019/095504 10/07/2019 CN

(45) 25/08/2025 449

(43) 25/05/2022 410A

(73) 1. BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD. (CN)

Room B-0035, 2/F, No.3 Building No.30, Shixing Road, Shijingshan District Beijing
100041, China

2. BYTEDANCE INC. (US)

12655 West Jefferson Boulevard Sixth Floor, Suite No. 137 Los Angeles, California
90066, USA

(72) XU, Jizheng (CN); ZHANG, Li (CN); ZHANG, Kai (CN); LIU, Hongbin (CN);
WANG, Yue (CN).

(74) Công ty Luật TNHH Phạm và Liên danh (PHAM & ASSOCIATES)

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ XỬ LÝ DỮ LIỆU VIDEO, PHƯƠNG PHÁP LƯU
TRỮ DÒNG BIT, VÀ VẬT GHI MÁY TÍNH ĐỌC ĐƯỢC

(21) 1-2022-00166

- (57) Sáng chế đề cập đến phương pháp xử lý media hình ảnh bao gồm xác định kích thước của bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ sao chép trong khối (intra block copy, IBC); và thực hiện biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

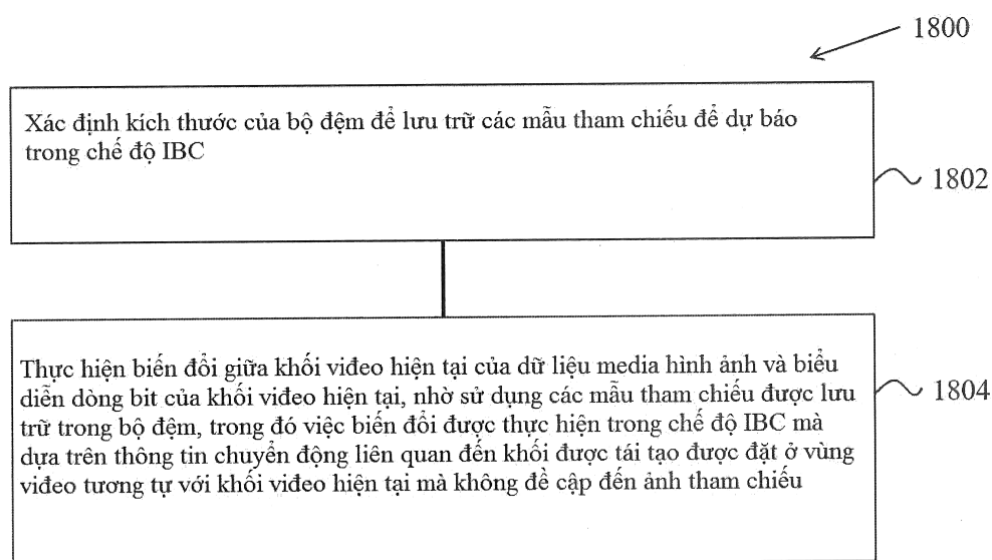


Fig.18

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến các kỹ thuật, các thiết bị và các hệ thống mã hóa và giải mã video.

Tình trạng kỹ thuật của sáng chế

Mặc dù có các cải tiến trong kỹ thuật nén video, video số vẫn chiếm lượng lớn sử dụng băng thông trên mạng Internet và các mạng truyền thông số khác. Do số lượng thiết bị người dùng được kết nối có thể nhận và hiển thị video tăng lên, mong đợi rằng nhu cầu băng thông đối với việc sử dụng video số sẽ tiếp tục tăng lên.

Bản chất kỹ thuật của sáng chế

Sáng chế mô tả các phương án thực hiện và các kỹ thuật khác nhau để quản lý bộ đệm và mã hóa vector khối đối với chế độ sao chép bên trong khối (IBC) để giải mã hoặc mã hóa video hoặc ảnh.

Một khía cạnh ví dụ đề cập đến phương pháp xử lý video hoặc ảnh (dữ liệu hình ảnh). Phương pháp bao gồm xác định kích thước của bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC; và thực hiện biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Khía cạnh ví dụ khác đề cập đến phương pháp khác xử lý dữ liệu hình ảnh. Phương pháp bao gồm các bước xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu được tái tạo để dự báo trong chế độ IBC, trong đó bộ

đệm được sử dụng để lưu trữ các mẫu được tái tạo trước bước lọc vòng; và thực hiện việc biến đổi mà sử dụng các mẫu được tái tạo được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu..

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý dữ liệu hình ảnh. Phương pháp bao gồm các bước xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu được tái tạo để dự báo trong chế độ IBC, trong đó bộ đệm được sử dụng để lưu trữ các mẫu được tái tạo after bước lọc vòng; và thực hiện việc biến đổi mà sử dụng các mẫu được tái tạo được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu được tái tạo để dự báo trong chế độ IBC, trong đó bộ đệm được sử dụng để lưu trữ các mẫu được tái tạo cả trước bước lọc vòng lẫn sau bước lọc vòng; và thực hiện việc biến đổi mà sử dụng các mẫu được tái tạo được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Khía cạnh ví dụ khác đề cập phương pháp khác xử lý video. Phương pháp bao gồm sử dụng bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó chiều sâu bit thứ nhất của bộ đệm khác nhau than chiều sâu bit thứ hai được sử dụng để biểu diễn dữ liệu media hình ảnh trong biểu diễn dòng bit; và thực hiện biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà

dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm bước khởi tạo bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó bộ đệm được khởi tạo với giá trị thứ nhất; và thực hiện biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm bước khởi tạo bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó, dựa trên tính khả dụng của một hoặc nhiều khối video trong dữ liệu media hình ảnh, bộ đệm được khởi tạo với các giá trị pixel của một hoặc nhiều khối video trong dữ liệu media hình ảnh; và thực hiện biến đổi giữa khối video hiện tại mà không thuộc một hoặc nhiều khối video của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm các bước xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC; thực hiện việc biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu; và đối với pixel được đặt trong không gian ở vị trí (x_0, y_0) và có vector khối (BV_x, BV_y) được bao gồm trong thông tin chuyển động, tính toán tham chiếu tương ứng trong bộ đệm dựa trên vị trí tham

chiều ($P \bmod M, Q \bmod N$) trong đó “mod” là phép toán modulo và M và N là các số nguyên biểu diễn các chiều x và y của bộ đệm, trong đó vị trí tham chiếu (P, Q) được xác định nhờ sử dụng vector khối (BV_x, BV_y) và vị trí (x_0, y_0).

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm các bước xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC; thực hiện biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu; và đối với pixel được đặt trong không gian ở vị trí (x_0, y_0) và có vector khối (BV_x, BV_y) được bao gồm trong thông tin chuyển động, tính toán tham chiếu tương ứng trong bộ đệm dựa trên vị trí tham chiếu (P, Q), trong đó vị trí tham chiếu (P, Q) được xác định nhờ sử dụng vector khối (BV_x, BV_y) và vị trí (x_0, y_0).

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm các bước xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó các vị trí pixel trong bộ đệm được giải quyết nhờ sử dụng các số x và y ; và thực hiện, dựa trên các số x và y , việc biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm các bước xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu; đối với pixel được đặt trong không gian ở vị

trí (x_0, y_0) của khối video hiện tại và có vector khối (BV_x, BV_y) , tính toán tham chiếu tương ứng trong bộ đệm ở vị trí tham chiếu (P, Q) , trong đó vị trí tham chiếu (P, Q) được xác định nhờ sử dụng vector khối (BV_x, BV_y) và vị trí (x_0, y_0) ; và upon xác định rằng vị trí tham chiếu (P, Q) nằm ngoài bộ đệm, tính toán lại vị trí tham chiếu nhờ sử dụng mẫu trong bộ đệm.

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm các bước xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu; đối với pixel được đặt trong không gian ở vị trí (x_0, y_0) của khối video hiện tại so với vị trí trên bên trái của CTU bao gồm khối video hiện tại và có vector khối (BV_x, BV_y) , tính toán tham chiếu tương ứng trong bộ đệm ở vị trí tham chiếu (P, Q) , trong đó vị trí tham chiếu (P, Q) được xác định nhờ sử dụng vector khối (BV_x, BV_y) và vị trí (x_0, y_0) ; và khi xác định rằng vị trí tham chiếu (P, Q) nằm ngoài bộ đệm, giới hạn ít nhất một phần vị trí tham chiếu để nằm trong khoảng định trước.

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu; đối với pixel được đặt trong không gian ở vị trí (x_0, y_0) của khối video hiện tại so với vị trí trên bên trái của CTU bao gồm khối video hiện tại và có vector khối (BV_x, BV_y) , tính toán tham chiếu tương ứng trong bộ đệm ở vị trí tham chiếu (P, Q) , trong đó vị trí tham chiếu (P, Q) được xác định nhờ sử dụng vector khối (BV_x, BV_y) và vị trí (x_0, y_0) ; và khi xác định rằng vector khối (BV_x, BV_y) nằm ngoài bộ đệm, đệm vector khối (BV_x, BV_y) theo vector khối của giá trị mẫu nằm trong bộ đệm.

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm bước thiết lập lại, trong khi biến đổi giữa video và biểu diễn dòng bit của video, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC ở biên video; và thực hiện việc biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi khối video của video được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video mà không đề cập đến ảnh tham chiếu.

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm thực hiện biến đổi giữa khối video hiện tại và biểu diễn dòng bit của khối video hiện tại; và cập nhật bộ đệm được sử dụng để lưu trữ các mẫu tham chiếu để dự báo ở chế độ IBC, trong đó bộ đệm được sử dụng đối với biến đổi giữa khối video tiếp theo và biểu diễn dòng bit của khối video tiếp theo, trong đó việc biến đổi giữa khối video tiếp theo và biểu diễn dòng bit của khối video tiếp theo được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video tiếp theo mà không đề cập đến ảnh tham chiếu.

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm xác định, đối với biến đổi giữa khối video hiện tại và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà được sử dụng để lưu trữ các mẫu được tái tạo để dự báo trong chế độ IBC, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu; và áp dụng hoạt động tiền xử lý cho các mẫu được tái tạo được lưu trữ trong bộ đệm, đáp lại xác định rằng các mẫu được tái tạo được lưu trữ trong bộ đệm cần được sử dụng để dự báo các giá trị mẫu trong khi hội thoại.

Khía cạnh ví dụ khác nữa đề cập đến phương pháp khác xử lý video. Phương pháp bao gồm các bước xác định, có chọn lọc đối với biến đổi giữa khối video hiện tại của đơn vị dữ liệu đường ống ảo hiện tại (VPDU) của vùng video và biểu diễn dòng bit của khối video hiện tại, liệu có sử dụng K1 VPDU được xử lý trước đó từ hàng được đánh số chẵn của vùng video và/hoặc K2 VPDU được xử lý

trước đó từ hàng được đánh số lẻ của vùng video; và thực hiện việc biến đổi, trong đó việc biến đổi loại trừ việc sử dụng phần còn lại của VPDU hiện tại, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video mà không đề cập đến ảnh tham chiếu.

Khía cạnh ví dụ khác nữa đề cập đến bộ mã hóa video hoặc bộ giải mã bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp trên được mô tả.

Khía cạnh ví dụ khác đề cập đến vật ghi máy tính đọc được. Phương tiện lưu trữ mã mà thực thi các lệnh bộ xử lý có thể thực thi để thực hiện một trong các phương pháp được bộc lộ.

Các khía cạnh này và các khía cạnh khác được mô tả chi tiết hơn theo sáng chế.

Mô tả vắn tắt các hình vẽ

Fig.1 là hình vẽ thể hiện ví dụ tham chiếu ảnh hiện tại hoặc kỹ thuật mã hóa ảnh video hoặc ảnh sao chép trong khối;

Fig.2 là hình vẽ thể hiện ví dụ diện tích tham chiếu động;

Fig.3 là hình vẽ thể hiện ví dụ mã hóa khối bắt đầu từ (x,y);

Fig.4 là hình vẽ thể hiện các ví dụ của các cách khả thi để chọn các khối 64×64 được mã hóa trước đó;

Fig.5 là hình vẽ thể hiện ví dụ các cách khả thi để thay đổi thứ tự mã hóa/giải mã của các khối 64×64;

Fig.6 là lưu đồ của phương pháp lấy làm ví dụ xử lý video hoặc ảnh;

Fig.7 là sơ đồ khối của nền tảng phần cứng để mã hóa hoặc giải mã video hoặc ảnh;

Fig.8 là hình vẽ thể hiện các cách khả thi khác để chọn các khối 64×64 được mã hóa trước đó, khi thứ tự giải mã cho các khối 64x64 là từ trên xuống dưới, từ trái sang phải;

Fig.9 là hình vẽ thể hiện các cách khả thi khác để chọn các khối 64x64 được mã hóa trước đó;

Fig.10 là hình vẽ thể hiện lưu đồ ví dụ cho quá trình giải mã có tạo dạng lại;

Fig.11 là hình vẽ thể hiện các cách khả thi khác để chọn các khối 64x64 được mã hóa trước đó, khi thứ tự giải mã cho các khối 64x64 là từ trái sang phải, trên xuống dưới;

Fig.12 là hình vẽ minh họa trạng thái bộ đệm tham chiếu IBC, trong đó khối ký hiệu CTU 64x64;

Fig.13 là hình vẽ thể hiện một bố trí của khu vực tham chiếu cho IBC;

Fig.14 là hình vẽ thể hiện bố trí khác của khu vực tham chiếu cho IBC;

Fig.15 là hình vẽ thể hiện bố trí khác của khu vực tham chiếu cho IBC khi VPDU hiện tại là sang bên phải của biên ảnh;

Fig.16 là hình vẽ thể hiện ví dụ trạng thái của bộ đệm ảo khi các VPDU trong hàng CTU được giải mã tuần tự;

Fig.17 là sơ đồ khối của ví dụ hệ thống xử lý video trong đó các kỹ thuật được bộc lộ có thể được triển khai;

Fig.18 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.19 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.20 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.21 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.22 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.23 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.24 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.25 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.26 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.27 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.28 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.29 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.30 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.31 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.32 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh;

Fig.33 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh; và

Fig.34 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh.

Mô tả chi tiết sáng chế

Các tiêu đề bộ phận được sử dụng trong sáng chế để dễ hiểu và không giới hạn phạm vi của các phương án thực hiện được bộc lộ chỉ trong phần đó. Sáng chế mô tả các phương án thực hiện và các kỹ thuật khác nhau để quản lý bộ đệm và mã hóa vector khối đối với chế độ sao chép bên trong khối để giải mã hoặc mã hóa video hoặc các ảnh.

1. Tóm tắt

Sáng chế liên quan đến các công nghệ mã hóa video. Cụ thể là, liên quan đến IBC trong mã hóa video. Có thể được áp dụng cho chuẩn đang phát triển, chẳng hạn, tạo mã đa dụng (Versatile Video Coding, VVC). Có thể cũng được áp dụng cho các chuẩn mã hóa video hoặc codec video tương lai.

2. Mô tả vắn tắt

Các chuẩn mã hóa video đã phát triển chủ yếu qua phát triển các chuẩn ITU-T và ISO/IEC đã biết. ITU-T đã đưa ra H.261 và H.263, ISO/IEC đã tạo ra MPEG-1 và MPEG-4 Visual, và hai tổ chức này đồng thời tạo các chuẩn H.262/MPEG-2 Video và H.264/MPEG-4 mã hóa video cải tiến (AVC) và H.265/HEVC. Do H.262, các chuẩn mã hóa video dựa trên cấu trúc mã hóa video lai trong đó mã hóa biến đổi cộng dự báo thời gian được tận dụng. Để khai triển các công nghệ mã hóa video tương lai ngoài HEVC, nhóm khai thác video chung (JVET) được thành lập bởi VCEG và MPEG đồng thời vào năm 2015. Kể từ đó, nhiều phương pháp mới đã được chấp thuận bởi JVET và được đưa vào phần mềm tham chiếu tên mô hình có tên là mô hình khai thác chung (JEM). Vào tháng 4, 2018, JVET giữa VCEG (Q6/16) và ISO/IEC JTC1 SC29/WG11 (MPEG) được tạo để làm việc trên chuẩn VVC nhằm vào việc giảm tốc độ bit 50% so với HEVC.

2.1 Dự báo ngoài trong HEVC/H.265

Mỗi PU được dự báo ngoài có các tham số chuyển động đối với một hoặc hai danh sách ảnh tham chiếu. Các tham số chuyển động bao gồm vector chuyển động và chỉ số ảnh tham chiếu. Sử dụng một trong hai danh sách ảnh tham chiếu có

thể also được báo hiệu nhờ sử dụng *inter_pred_idc*. Các vector chuyển động có thể được mã hóa riêng rẽ như là các delta so với các bộ dự báo.

Khi CU được tạo mã bằng chế độ nhảy, một PU được liên kết với CU, và không có hệ số dư quan trọng, không có delta vector chuyển động được mã hóa hoặc chỉ số ảnh tham chiếu. Chế độ hợp nhất được xác định nhờ đó các tham số chuyển động cho PU hiện tại thu được từ các PU lân cận, bao gồm các ứng viên không gian và thời gian. Chế độ hợp nhất có thể được áp dụng cho PU được dự báo ngoài bất kỳ, không chỉ đối với chế độ nhảy. Tùy chọn đối với chế độ hợp nhất là phiên truyền tường minh của các tham số chuyển động, trong đó vector chuyển động (chính xác hơn, các hiệu số vector chuyển động (MVD) so với bộ dự báo vector chuyển động), chỉ số ảnh tham chiếu tương ứng đối với mỗi danh sách ảnh tham chiếu và sử dụng danh sách ảnh tham chiếu được báo hiệu tường minh cho mỗi PU. Chế độ này được đặt tên là dự báo vector chuyển động cao cấp (AMVP) theo sáng chế.

Khi báo hiệu chỉ báo rằng một trong hai danh sách ảnh tham chiếu cần được sử dụng, PU được tạo từ một khối mẫu. Đây được gọi là ‘dự báo đơn’. Dự báo đơn khả dụng cho cả các lát P và các lát B.

Khi báo hiệu chỉ báo rằng cả hai danh sách ảnh tham chiếu cần được sử dụng, PU được tạo từ hai khối mẫu. Đây được gọi là ‘dự báo kép’. Dự báo kép khả dụng cho riêng các lát B.

Văn bản sau nêu các chi tiết về các chế độ dự báo ngoài được xác định trong HEVC. Phần mô tả sẽ bắt đầu với chế độ hợp nhất.

2.2 Tham chiếu ảnh hiện tại

Tham chiếu ảnh hiện tại (CPR), hoặc khi được đặt tên là IBC đã được chấp thuận trong các phần mở rộng mã hóa nội dung màn hình HEVC (HEVC-SCC) và mô hình kiểm tra VVC hiện tại. IBC mở rộng khái niệm bù chuyển động từ mã hóa ngoài khung hoặc mã hóa trong khung. Như được minh họa trên Fig.1, khối hiện tại được dự báo bằng khối tham chiếu trong cùng ảnh khi CPR được áp dụng. Các mẫu trong khối tham chiếu đã được tái tạo trước khi khối hiện tại được tạo mã hoặc được giải mã. Mặc dù CPR không quá hiệu quả cho phần lớn

chuỗi camera được chụp, thể hiện các độ lợi mã hóa đáng kể cho nội dung màn hình. Nguyên nhân chính là có nhiều mẫu hình lặp lại, chẳng hạn các biểu tượng và các ký tự văn bản trong ảnh nội dung màn hình. CPR có thể loại bỏ hiệu quả dư thừa giữa các mẫu hình lặp lại này. Trong HEVC-SCC, CU được mã hóa ngoài có thể áp dụng CPR nếu nó chọn ảnh hiện tại làm ảnh tham chiếu. MV được đặt lại tên là vector khối (BV) trong trường hợp này, và BV luôn có độ chính xác pixel nguyên. Để tương thích với HEVC tiêu chuẩn, ảnh hiện tại được đánh dấu là ảnh tham chiếu “dài hạn” trong bộ đệm ảnh được giải mã (DPB). Cần lưu ý rằng tương tự, trong các chuẩn mã hóa video nhiều khung hình/3D, ảnh tham chiếu khung ngoài cũng được đánh dấu là ảnh tham chiếu “dài hạn”.

Tiếp theo BV để tìm khối tham chiếu của nó, dự báo có thể được tạo bằng cách sao chép khối tham chiếu. Phần dư có thể thu được bằng cách lấy các tín hiệu ban đầu trừ đi các pixel tham chiếu. Sau đó, biến đổi và lượng tử hóa có thể được áp dụng như trong các chế độ mã hóa khác.

Fig.1 là ví dụ minh họa tham chiếu ảnh hiện tại.

Tuy nhiên, khi khối tham chiếu nằm ngoài ảnh, hoặc chồng lấp với khối hiện tại, hoặc ngoài khu vực được tái tạo, hoặc ngoài khu vực hợp lệ bị giới hạn bởi một số giới hạn, một phần hoặc tất cả các giá trị pixel không được định nghĩa. Về cơ bản, có hai giải pháp xử lý vấn đề này. Một là không cho phép tình trạng này, chẳng hạn, khi tương thích dòng bit. Giải pháp còn lại là có áp dụng đệm cho các giá trị pixel không được định nghĩa đó. Các phần phụ sau mô tả chi tiết các giải pháp.

2.3 CPR trong các phần mở rộng mã hóa nội dung màn hình HEVC

Trong các phần mở rộng mã hóa nội dung màn hình của HEVC, khi khối sử dụng ảnh hiện tại làm tham chiếu, cần đảm bảo rằng toàn bộ khối tham chiếu nằm trong khu vực được tái tạo khả dụng, như được chỉ báo trong đoạn mô tả sau:

The variables `offsetX` and `offsetY` are derived as follows:

$$\text{offsetX} = (\text{ChromaArrayType} == 0) ? 0 : (\text{mvCLX}[0] \& 0x7 ? 2 : 0) \quad (8-104)$$

$$\text{offsetY} = (\text{ChromaArrayType} == 0) ? 0 : (\text{mvCLX}[1] \& 0x7 ? 2 : 0) \quad (8-105)$$

It is a requirement of bitstream conformance that when the reference picture is the current picture, the luma motion vector `mvLX` shall obey the following constraints:

- When the derivation process for z-scan order block availability as specified in clause 6.4.1 is invoked with `(xCurr, yCurr)` set equal to `(xCb, yCb)` and the neighbouring luma location `(xNbY, yNbY)` set equal to `(xPb + (mvLX[0] >> 2) - offsetX, yPb + (mvLX[1] >> 2) - offsetY)` as inputs, the output shall be equal to TRUE.
- When the derivation process for z-scan order block availability as specified in clause 6.4.1 is invoked with `(xCurr, yCurr)` set equal to `(xCb, yCb)` and the neighbouring luma location `(xNbY, yNbY)` set equal to `(xPb + (mvLX[0] >> 2) + nPbW - 1 + offsetX, yPb + (mvLX[1] >> 2) + nPbH - 1 + offsetY)` as inputs, the output shall be equal to TRUE.
- One or both of the following conditions shall be true:
 - The value of `(mvLX[0] >> 2) + nPbW + xB1 + offsetX` is less than or equal to 0.
 - The value of `(mvLX[1] >> 2) + nPbH + yB1 + offsetY` is less than or equal to 0.
- The following condition shall be true:

$$(\text{xPb} + (\text{mvLX}[0] \gg 2) + \text{nPbSw} - 1 + \text{offsetX}) / \text{CtbSizeY} - \text{xCb} / \text{CtbSizeY} \leq \text{yCb} / \text{CtbSizeY} - (\text{yPb} + (\text{mvLX}[1] \gg 2) + \text{nPbSh} - 1 + \text{offsetY}) / \text{CtbSizeY} \quad (8-106)$$

Do vậy, trường hợp mà khối tham chiếu chồng lấp với khối hiện tại hoặc khối tham chiếu nằm ngoài ảnh sẽ không xuất hiện. Không cần đếm khối tham chiếu hoặc khối dự báo.

2.4 Các ví dụ về CPR/IBC

Trong mô hình kiểm tra VVC, toàn bộ khối tham chiếu nên có CTU hiện tại và không chồng lấp với khối hiện tại. Do vậy, không cần đếm khối tham chiếu hoặc dự báo.

Khi cây đôi được kích hoạt, cấu trúc phân vùng có thể khác với các CTU độ sáng đến sắc độ. Do vậy, đối với định dạng màu 4:2:0, một khối sắc độ (chẳng hạn, CU) có thể tương ứng với một vùng độ sáng cùng vị trí đã được tách thành nhiều CU độ sáng.

Khối sắc độ có thể chỉ được mã hóa với chế độ CPR khi các điều kiện sau là đúng:

- 1) Mỗi CU trong CU độ sáng trong khối độ sáng cùng vị trí sẽ được mã hóa với chế độ CPR.
- 2) Mỗi BV của BV của khối 4×4 độ sáng được biến đổi trước sang BV của khối sắc độ và BV của khối sắc độ là BV hợp lệ.

Nếu một trong hai điều kiện là sai, khối sắc độ sẽ không được mã hóa bằng chế độ CPR.

Lưu ý rằng định nghĩa về ‘BV hợp lệ’ có các giới hạn sau:

1) Tất cả các mẫu trong khối tham chiếu được nhận diện bởi BV sẽ nằm trong trong khoảng tìm kiếm bị giới hạn (chẳng hạn, sẽ nằm trong cùng CTU trong thiết kế VVC hiện tại).

2) Tất cả các mẫu trong khối tham chiếu được nhận diện bởi BV đã được tái tạo.

2.5 Các ví dụ về CPR/IBC

Theo một số ví dụ, khu vực tham chiếu cho CPR/IBC bị giới hạn ở CTU hiện tại, lên đến 128×128 . Khu vực tham chiếu bị thay đổi động để sử dụng lại bộ nhớ để lưu trữ các mẫu tham chiếu cho CPR/IBC sao cho khối CPR/IBC có thể có nhiều ứng viên tham chiếu trong khi bộ đệm tham chiếu cho CPR/IBC có thể được giữ hoặc giảm từ một CTU.

Fig.2 thể hiện phương pháp, trong đó khối có 64×64 và CTU chứa 4 khối 64×64 . Khi mã hóa khối 64×64 , 3 khối 64×64 trước đó có thể được sử dụng làm tham chiếu. Nhờ như vậy, bộ giải mã chỉ cần lưu trữ 4 khối 64×64 để hỗ trợ CPR/IBC.

Giả sử rằng vị trí của CU độ sáng hiện tại so với góc trên bên trái của ảnh là (x, y) và vectơ khối là (BV_x, BV_y) . Theo thiết kế hiện tại, nếu BV là hợp lệ có thể được báo bởi việc vị trí độ sáng $((x+BV_x) \gg 6 \ll 6 + (1 \ll 7), (y+BV_y) \gg 6 \ll 6)$ đã không được tái tạo và $((x+BV_x) \gg 6 \ll 6 + (1 \ll 7), (y+BV_y) \gg 6 \ll 6)$ không bằng $(x \gg 6 \ll 6, y \gg 6 \ll 6)$.

2.6 Tạo dạng lại trong vòng (ILR)

Ý tưởng cơ bản về ILR là để biến đổi tín hiệu ban đầu (trong miền thứ nhất) (tín hiệu dự báo/tái tạo) sang miền thứ hai (miền được tạo dạng lại).

ILR được triển khai dưới dạng cặp bảng tra cứu (LUT), nhưng chỉ một trong hai LUT cần được báo hiệu dưới dạng LUT còn lại có thể được tính toán từ LUT được báo hiệu. Mỗi LUT là bảng ánh xạ 1024 mục, 10 bit, một chiều (1D-LUT).

Một LUT là LUT thuận, *FwdLUT*, mà ánh xạ các giá trị mã độ sáng đầu vào Y_i đến các giá trị được chỉnh sửa Y_r : $Y_r = FwdLUT[Y_i]$. LUT còn lại là LUT đảo, *InvLUT*, mà ánh xạ các giá trị mã được chỉnh sửa Y_r đến $\hat{Y}_i$: $\hat{Y}_i = InvLUT[Y_r]$. ($\hat{Y}_i$ là các giá trị tái tạo của Y_i).

2.6.1 Mô hình PWL

Về mặt khái niệm, PWL được triển khai theo cách sau:

Lấy x_1, x_2 là hai điểm trục xoay đầu vào, và y_1, y_2 là các điểm trục xoay đầu ra tương ứng cho một đoạn. Giá trị đầu ra y cho giá trị đầu vào bất kỳ x giữa x_1 và x_2 có thể được nội suy bằng phương trình sau:

$$y = ((y_2 - y_1) / (x_2 - x_1)) * (x - x_1) + y_1$$

Theo triển khai dấu chấm tĩnh, phương trình có thể được viết lại dưới dạng:

$$y = ((m * x + 2^{FP_PREC-1}) >> FP_PREC) + c$$

trong đó m là vô hướng, c là độ lệch, và FP_PREC là giá trị hằng số để xác định độ chính xác.

Theo một số ví dụ, mô hình PWL được sử dụng để tính toán trước các bảng ánh xạ *FwdLUT* và *InvLUT* 1024 mục; nhưng mô hình PWL cũng cho phép các cách triển khai để tính toán các giá trị ánh xạ giống hệt trực tiếp mà không tính toán trước các LUT.

2.6.2.1 Tạo dạng lại độ sáng

Phương pháp tạo dạng lại trong vòng độ sáng đề cập đến đường ống có độ phức tạp thấp hơn cũng loại bỏ độ trễ giải mã để dự báo trong theo khối khi tái tạo ngoài lát. Dự báo trong được thực hiện trong miền được tạo dạng lại cho cả hai lát trong và ngoài.

Dự báo trong cũng được thực hiện trong miền được tạo dạng lại bất kể loại lát. Với cách bố trí này, dự báo trong có thể khởi động ngay sau khi thực hiện tái tạo TU trước đó. Cách bố trí này cũng có thể nêu quá trình thống nhất cho chế độ trong thay vì phụ thuộc lát. Fig.10 thể hiện sơ đồ khối của quá trình giải mã CE12-2 dựa trên chế độ.

Các mô hình PWL 16 đoạn được kiểm tra đối với độ sáng và chia tỷ lệ phần dư sắc độ thay vào các mô hình PWL 32 đoạn.

Tái tạo ngoài lát bằng bộ tạo dạng lại độ sáng trong vòng (các khối được tô màu lục nhạt chỉ báo tín hiệu trong miền được tạo dạng lại: phần dư độ sáng; được dự báo trong độ sáng; và được tái tạo trong độ sáng)

2.6.2.2 Phép chia tỷ lệ phần dư sắc độ phụ thuộc độ sáng

Phép chia tỷ lệ phần dư sắc độ phụ thuộc độ sáng là quá trình nhân bội được triển khai với phép toán số nguyên dấu chấm tĩnh. Phép chia tỷ lệ phần dư sắc độ bù đối với tương tác tín hiệu độ sáng với tín hiệu sắc độ. Phép chia tỷ lệ phần dư sắc độ được áp dụng ở mức TU. Cụ thể hơn, phần sau áp dụng:

Đối với bên trong, độ sáng được tái tạo được tính trung bình.

Đối với bên ngoài, độ sáng dự báo được tính trung bình.

Giá trị trung bình được sử dụng để nhận diện chỉ số trong mô hình PWL. Chỉ số nhận diện hệ số chia tỷ lệ cScaleInv. Phần dư sắc độ được nhân với số đó.

Lưu ý rằng hệ số chia tỷ lệ sắc độ được tính toán từ các giá trị độ sáng được dự báo được ánh xạ thuận thay vì các giá trị độ sáng được tái tạo

2.6.2.3 Báo hiệu thông tin phụ ILR

Các tham số này (hiện) được gửi trong tiêu đề nhóm phiên (giống như ALF). Báo cáo rằng chúng chiếm 40 đến 100 bit.

Theo một số ví dụ, cú pháp được bổ sung được nêu bật bằng chữ in nghiêng.

Trong 7.3,2,1 Cú pháp RBSP tập hợp chuỗi tham số

seq_parameter_set_rbsp() {	Mô tả
sps_seq_parameter_set_id	• ue(v)
...	•
sps_triangle_enabled_flag	• u(1)
sps_ladf_enabled_flag	• u(1)
if (sps_ladf_enabled_flag) {	•
sps_num_ladf_intervals_minus2	• u(2)
sps_ladf_lowest_interval_qp_offset	• se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	•
sps_ladf_qp_offset[i]	• se(v)
sps_ladf_delta_threshold_minus1[i]	• ue(v)
}	•
}	•

<i>sps_reshaper_enabled_flag</i>	• <i>u(1)</i>
<i>rbps_trailing_bits()</i>	•
}	•

Trong 7.3.3.1 Cú pháp tiêu đề nhóm phiên chung

<i>tile_group_header()</i> {	Mô tả
...	
if(<i>num_tiles_in_tile_group_minus1</i> > 0) {	
<i>offset_len_minus1</i>	<i>ue(v)</i>
for(<i>i</i> = 0; <i>i</i> < <i>num_tiles_in_tile_group_minus1</i> ; <i>i</i> ++)	
<i>entry_point_offset_minus1[i]</i>	<i>u(v)</i>
}	
if(<i>sps_reshaper_enabled_flag</i>) {	•
<i>tile_group_reshaper_model_present_flag</i>	• <i>u(1)</i>
if(<i>tile_group_reshaper_model_present_flag</i>)	•
<i>tile_group_reshaper_model()</i>	•
<i>tile_group_reshaper_enable_flag</i>	• <i>u(1)</i>
if(<i>tile_group_reshaper_enable_flag</i> && (!(<i>qtbtt_dual_tree_intra_flag</i> && <i>tile_group_type</i> == 1)))	•
<i>tile_group_reshaper_chroma_residual_scale_flag</i>	• <i>u(1)</i>
}	•
<i>byte_alignment()</i>	•
}	•

Bổ sung mô hình bộ tạo dạng lại nhóm phiên bằng cú pháp mới:

<i>tile_group_reshaper_model()</i> {	Mô tả
<i>reshaper_model_min_bin_idx</i>	<i>ue(v)</i>
<i>reshaper_model_delta_max_bin_idx</i>	<i>ue(v)</i>
<i>reshaper_model_bin_delta_abs_cw_prec_minus1</i>	<i>ue(v)</i>
for (<i>i</i> = <i>reshaper_model_min_bin_idx</i> ; <i>i</i> <= <i>reshaper_model_max_bin_idx</i> ; <i>i</i> ++) {	
<i>reshape_model_bin_delta_abs_CW[i]</i>	<i>u(v)</i>
if(<i>reshaper_model_bin_delta_abs_CW[i]</i> > 0)	
<i>reshaper_model_bin_delta_sign_CW_flag[i]</i>	<i>u(1)</i>
}	
}	

Trong ngữ nghĩa RBSP tập hợp tham số tổng quát, bổ sung các ngữ nghĩa sau:

sps_reshaper_enabled_flag bằng 1 xác định rằng bộ tạo dạng lại được sử dụng trong chuỗi video được mã hóa (CVS). *sps_reshaper_enabled_flag* bằng 0 xác định rằng bộ tạo dạng lại không được sử dụng trong CVS.

Trong cú pháp tiêu đề nhóm phiên, bổ sung các ngữ nghĩa sau

tile_group_reshaper_model_present_flag bằng 1 xác định **tile_group_reshaper_model()** hiện có trong tiêu đề nhóm phiên. **tile_group_reshaper_model_present_flag** bằng 0 xác định **tile_group_reshaper_model()** không có trong tiêu đề nhóm phiên. Khi **tile_group_reshaper_model_present_flag** không có, được suy ra bằng 0.

tile_group_reshaper_enabled_flag bằng 1 xác định rằng bộ tạo dạng lại kích hoạt cho nhóm phiên hiện tại. **tile_group_reshaper_enabled_flag** bằng 0 xác định rằng bộ tạo dạng lại không được kích hoạt cho nhóm phiên hiện tại. Khi **tile_group_reshaper_enabled_flag** không có, được suy ra bằng 0.

tile_group_reshaper_chroma_residual_scale_flag bằng 1 xác định rằng phép chia tỷ lệ phần dư sắc độ được kích hoạt cho nhóm phiên hiện tại. **tile_group_reshaper_chroma_residual_scale_flag** bằng 0 xác định rằng phép chia tỷ lệ phần dư sắc độ không được kích hoạt cho nhóm phiên hiện tại. Khi **tile_group_reshaper_chroma_residual_scale_flag** không có, được suy ra bằng 0.

*Bổ sung cú pháp **tile_group_reshaper_model()***

reshape_model_min_bin_idx xác định chỉ số bin (hoặc đoạn) nhỏ nhất cần được sử dụng trong quá trình tạo bộ tạo dạng lại. Giá trị của **reshape_model_min_bin_idx** cần nằm trọn trong khoảng từ 0 đến **MaxBinIdx**. Giá trị của **MaxBinIdx** cần bằng 15.

reshape_model_delta_max_bin_idx xác định chỉ số bin (hoặc đoạn) được phép lớn nhất **MaxBinIdx** trừ đi chỉ số bin lớn nhất cần được sử dụng trong quá trình tạo bộ tạo dạng lại. Giá trị của **reshape_model_delta_max_bin_idx** được gán bằng **MaxBinIdx – reshape_model_delta_max_bin_idx**.

reshaper_model_bin_delta_abs_cw_prec_minus1 cộng 1 xác định số lượng bit được sử dụng cho biểu diễn cú pháp **reshape_model_bin_delta_abs_CW[i]**.

reshape_model_bin_delta_abs_CW[i] xác định giá trị từ mã delta tuyệt đối cho bin thứ i.

reshaper_model_bin_delta_sign_CW_flag[i] xác định dấu của **reshape_model_bin_delta_abs_CW[i]** như sau:

– Nếu $\text{reshape_model_bin_delta_sign_CW_flag}[i]$ bằng 0, biến tương ứng $\text{RspDeltaCW}[i]$ là giá trị dương.

– Ngược lại ($\text{reshape_model_bin_delta_sign_CW_flag}[i]$ không bằng 0), biến tương ứng $\text{RspDeltaCW}[i]$ là giá trị âm.

Khi $\text{reshape_model_bin_delta_sign_CW_flag}[i]$ không có, được suy ra bằng 0.

Biến $\text{RspDeltaCW}[i] = (1 \ll \text{reshape_model_bin_delta_sign_CW}[i]) * \text{reshape_model_bin_delta_abs_CW}[i]$;

Biến $\text{RspCW}[i]$ được suy ra như các bước sau:

Biến OrgCW được gán bằng $(1 \ll \text{BitDepth}_Y) / (\text{MaxBinIdx} + 1)$.

– Nếu $\text{reshaper_model_min_bin_idx} < i \leq \text{reshaper_model_max_bin_idx}$

$\text{RspCW}[i] = \text{OrgCW} + \text{RspDeltaCW}[i]$.

– Ngược lại, $\text{RspCW}[i] = 0$.

Giá trị của $\text{RspCW}[i]$ cần nằm trọn trong khoảng từ 32 tới $2 * \text{OrgCW} - 1$ nếu giá trị của BitDepth_Y bằng 10.

Các biến $\text{InputPivot}[i]$ với i nằm trọn trong khoảng từ 0 đến $\text{MaxBinIdx} + 1$, inclusive được suy ra như sau

$\text{InputPivot}[i] = i * \text{OrgCW}$

Biến $\text{ReshapePivot}[i]$ với i nằm trọn trong khoảng từ 0 đến $\text{MaxBinIdx} + 1$, biến $\text{ScaleCoef}[i]$ và $\text{InvScaleCoeff}[i]$ với i nằm trọn trong khoảng từ 0 đến MaxBinIdx , được suy ra như sau:

$\text{shiftY} = 14$

$\text{ReshapePivot}[0] = 0$;

for($i = 0$; $i \leq \text{MaxBinIdx}$; $i++$) {

$\text{ReshapePivot}[i + 1] = \text{ReshapePivot}[i] + \text{RspCW}[i]$

$\text{ScaleCoef}[i] = (\text{RspCW}[i] * (1 \ll \text{shiftY}) + (1 \ll (\text{Log2}(\text{OrgCW}) - 1))) \gg (\text{Log2}(\text{OrgCW}))$

if ($\text{RspCW}[i] == 0$)

$\text{InvScaleCoeff}[i] = 0$

```

else
    InvScaleCoeff[i] = OrgCW * (1 << shiftY) / RspCW[i]
}

```

Biến ChromaScaleCoef[i] với i nằm trong khoảng từ 0 đến MaxBinIdx, được suy ra như sau:

```

ChromaResidualScaleLut[64] = {16384, 16384, 16384, 16384, 16384,
16384, 16384, 8192, 8192, 8192, 8192, 5461, 5461, 5461, 5461, 4096, 4096,
4096, 4096, 3277, 3277, 3277, 3277, 2731, 2731, 2731, 2731, 2341, 2341, 2341,
2048, 2048, 2048, 1820, 1820, 1820, 1638, 1638, 1638, 1638, 1489, 1489, 1489,
1489, 1365, 1365, 1365, 1365, 1260, 1260, 1260, 1260, 1170, 1170, 1170, 1170,
1092, 1092, 1092, 1092, 1024, 1024, 1024, 1024};

```

```

shiftC = 11

```

```

– if (RspCW[i] == 0)

```

```

    ChromaScaleCoef[i] = (1 << shiftC)

```

```

– Ngược lại (RspCW[i] != 0), ChromaScaleCoef[i] =

```

```

    ChromaResidualScaleLut[RspCW[i] >> 1]

```

2.6.2.4 Sử dụng ILR

Ở phía bộ mã hóa, mỗi ảnh (hoặc nhóm phiên) được biến đổi trước sang miền được tạo dạng lại. Và tất cả quá trình mã hóa được thực hiện trong miền được tạo dạng lại. Đối với dự báo trong, khối lân cận nằm trong miền được tạo dạng lại; đối với dự báo ngoài, các khối tham chiếu (được tạo từ miền ban đầu từ bộ đệm ảnh được giải mã) trước hết được biến đổi sang miền được tạo dạng lại. Sau đó, phần dư được tạo và được mã hóa thành dòng bit.

Sau khi toàn bộ ảnh (hoặc nhóm phiên) hoàn thành mã hóa/giải mã, các mẫu trong miền được tạo dạng lại được biến đổi sang miền ban đầu, thì lọc tách khối và các bộ lọc khác được áp dụng.

Tạo dạng lại thuận cho tín hiệu dự báo được vô hiệu hóa cho các trường hợp sau:

Khối hiện tại được tạo mã trong

Khối hiện tại được tạo mã dưới dạng CPR (tham chiếu ảnh hiện tại, hay còn gọi là sao chép trong khối, IBC)

Khối hiện tại được tạo mã dưới dạng chế độ trong – ngoài kết hợp (CIIP) và tạo dạng lại thuận được vô hiệu hóa cho khối dự báo trong

3. Các ví dụ về các vấn đề được giải quyết theo các phương án thực hiện khác nhau

Theo thiết kế hiện tại của CPR/IBC, có một số vấn đề.

Khu vực tham chiếu thay đổi động, khiến xử lý bộ mã hóa/bộ giải mã phức tạp.

Các vectơ khối không hợp lệ dễ được tạo và khó kiểm tra, mà gây phức tạp cả bộ mã hóa lẫn bộ giải mã.

Khu vực tham chiếu không đều dẫn đến mã hóa không hiệu quả vectơ khối.

Cách thức xử lý kích thước CTU nhỏ hơn 128x128 là không rõ ràng.

Trong quá trình xác định liệu BV là hợp lệ hoặc không hợp lệ, cho các khối sắc độ, quyết định dựa trên tính khả dụng của mẫu độ sáng mà có thể dẫn đến các quyết định sai do cấu trúc phân vùng cây đôi.

4. Các phương án thực hiện ví dụ

Theo một số phương án thực hiện, bộ đệm thường có thể được sử dụng cho khối CPR/IBC để có tham chiếu.

Hàm $\text{isRec}(x,y)$ được định nghĩa để chỉ báo liệu pixel (x,y) đã được tái tạo và được tham chiếu bởi chế độ IBC. Khi (x,y) nằm ngoài ảnh, của lát/phiến/gạch khác nhau, $\text{isRec}(x,y)$ trả về false; khi (x,y) đã không được tái tạo, $\text{isRec}(x,y)$ trả về false. Trong ví dụ khác, khi mẫu (x, y) đã được tái tạo nhưng một số điều kiện khác được thỏa mãn, cũng có thể được đánh dấu là không khả dụng, chẳng hạn nằm ngoài khu vực tham chiếu/trong VPDU khác, và $\text{isRec}(x,y)$ trả về false.

Hàm $\text{isRec}(c, x,y)$ được định nghĩa để chỉ báo nếu mẫu (x, y) đối với thành phần c khả dụng. Chẳng hạn, nếu mẫu (x, y) vẫn chưa được tái tạo, được đánh dấu là không khả dụng. Trong ví dụ khác, khi mẫu (x, y) đã được tái tạo nhưng một số điều kiện khác được thỏa mãn, cũng có thể được đánh dấu là không khả

dụng, chẳng hạn nằm ngoài ảnh/trong lát/phần/viên khác trong VPDU khác, nằm ngoài khu vực tham chiếu được phép. $\text{isRec}(c, x, y)$ trả về false khi mẫu (x, y) không khả dụng, ngược lại, trả về true.

Trong phần mô tả sau, các mẫu tham chiếu có thể là các mẫu được tái tạo. Lưu ý rằng ‘bộ đệm pixel’ có thể đáp lại ‘bộ đệm của một thành phần màu’ hoặc ‘bộ đệm của nhiều thành phần màu sắc’.

Bộ đệm tham chiếu cho CPR/IBC

1. Đề xuất sử dụng bộ đệm pixel $M \times N$ để lưu trữ các mẫu tham chiếu độ sáng cho CPR/IBC.

- a. Trong một ví dụ, kích thước bộ đệm bằng 64×64 .
- b. Trong một ví dụ, kích thước bộ đệm bằng 128×128 .
- c. Trong một ví dụ, kích thước bộ đệm bằng 64×128 .
- d. Trong một ví dụ, kích thước bộ đệm bằng 128×64 .
- e. Trong một ví dụ, N bằng chiều cao của CTU.
- f. Trong một ví dụ, $N = nH$, trong đó H là chiều cao của CTU, n là số nguyên dương.
- g. Trong một ví dụ, M bằng chiều rộng của CTU.
- h. Trong một ví dụ, $M = mW$, trong đó W là chiều rộng của CTU, m là số nguyên dương.
- i. Trong một ví dụ, kích thước bộ đệm không bằng kích thước CTU, chẳng hạn, 96×128 hoặc 128×96 .
- j. Trong một ví dụ, kích thước bộ đệm bằng kích thước CTU
- k. Trong một ví dụ, $M = mW$ và $N = H$, trong đó W và H là chiều rộng và chiều cao của CTU, m là số nguyên dương.
- l. Trong một ví dụ, $M = W$ và $N = nH$, trong đó W và H là chiều rộng và chiều cao của CTU, n là số nguyên dương.
- m. Trong một ví dụ, $M = mW$ và $N = nH$, trong đó W và H là chiều rộng và chiều cao của CTU, m và n là các số nguyên dương.
- n. Trong ví dụ trên, m và n có thể phụ thuộc kích thước CTU.
- i. Trong một ví dụ, khi kích thước CTU bằng 128×128 , $m = 1$ và $n = 1$.

- ii. Trong một ví dụ, khi kích thước CTU bằng 64×64 , $m=4$ và $n=1$.
- iii. Trong một ví dụ, khi kích thước CTU bằng 32×32 , $m=16$ và $n=1$.
- iv. Trong một ví dụ, khi kích thước CTU bằng 16×16 , $m=64$ và $n=1$.
 - o. Theo các cách, kích thước bộ đệm tương ứng với kích thước CTU.
 - p. Theo các cách, kích thước bộ đệm tương ứng với kích thước VPDU.
 - q. M và/hoặc N có thể được báo hiệu từ bộ mã hóa đến bộ giải mã, chẳng hạn trong VPS/SPS/PPS/tiêu đề ảnh/tiêu đề lát/tiêu đề nhóm phiên.

2. M và/hoặc N có thể khác nhau trong các tiêu sử/các mức/các tầng khác nhau được định nghĩa trong chuẩn. Đề xuất sử dụng bộ đệm pixel $M_c \times N_c$ khác để lưu trữ các mẫu tham chiếu sắc độ cho CPR/IBC.

- a. Trong một ví dụ, $M_c = M/2$ và $N_c = N/2$ cho video 4:2:0
- b. Trong một ví dụ, $M_c = M$ và $N_c = N$ cho video 4:4:4
- c. Trong một ví dụ, $M_c = M$ và $N_c = N/2$ cho video 4:2:2
- d. Theo các cách, M_c và N_c có thể độc lập với M và N .
- e. Trong một ví dụ, bộ đệm sắc độ bao gồm hai kênh, tương ứng với C_b và C_r .

- f. Trong một ví dụ, $M_c=M$ và $N_c=N$.

3. Đề xuất sử dụng bộ đệm mẫu $M \times N$ để lưu trữ các mẫu tham chiếu RGB cho CPR/IBC

- a. Trong một ví dụ, kích thước bộ đệm bằng 64×64 .
 - b. Trong một ví dụ, kích thước bộ đệm bằng 128×128 .
 - c. Trong một ví dụ, kích thước bộ đệm bằng 64×128 .
 - d. Trong một ví dụ, kích thước bộ đệm bằng 128×64 .
 - e. Theo các cách, kích thước bộ đệm tương ứng với kích thước CTU.
 - f. Theo các cách, kích thước bộ đệm tương ứng với kích thước VPDU.
4. Đề xuất rằng bộ đệm có thể lưu trữ các pixel được tái tạo trước khi lọc vòng. Lọc vòng có thể đề cập đến lọc tách khối, lọc vòng thích ứng (ALF), độ lệch mẫu thích ứng (SAO), ALF giữa các thành phần, hoặc các bộ lọc khác bất kỳ.

- a. Trong một ví dụ, bộ đệm có thể lưu trữ các mẫu trong CTU hiện tại.
- b. Trong một ví dụ, bộ đệm có thể lưu trữ các mẫu nằm ngoài CTU hiện tại.

c. Trong một ví dụ, bộ đệm có thể lưu trữ các mẫu từ phần bất kỳ của ảnh hiện tại.

d. Trong một ví dụ, bộ đệm có thể lưu trữ các mẫu từ các ảnh khác.

5. Đề xuất rằng bộ đệm có thể lưu trữ các pixel được tái tạo after lọc vòng. Lọc vòng có thể đề cập đến lọc tách khối, ALF, SAO, ALF giữa các thành phần, hoặc các bộ lọc khác bất kỳ.

a. Trong một ví dụ, bộ đệm có thể lưu trữ các mẫu trong CTU hiện tại.

b. Trong một ví dụ, bộ đệm có thể lưu trữ các mẫu ngoài CTU hiện tại.

c. Trong một ví dụ, bộ đệm có thể lưu trữ các mẫu từ phần bất kỳ của ảnh hiện tại.

d. Trong một ví dụ, bộ đệm có thể lưu trữ các mẫu từ các ảnh khác.

6. Đề xuất rằng bộ đệm có thể lưu trữ cả các mẫu được tái tạo trước khi lọc vòng và sau khi lọc vòng. Lọc vòng có thể đề cập đến lọc tách khối, ALF, SAO, ALF giữa các thành phần, hoặc các bộ lọc khác bất kỳ.

a. Trong một ví dụ, bộ đệm có thể lưu trữ cả các mẫu từ ảnh hiện tại lẫn các mẫu từ các ảnh khác, tùy thuộc tính khả dụng của các mẫu đó.

b. Trong một ví dụ, các mẫu tham chiếu từ các ảnh khác là từ các mẫu được tái tạo sau khi lọc vòng.

c. Trong một ví dụ, các mẫu tham chiếu từ các ảnh khác là từ các mẫu được tái tạo trước khi lọc vòng.

7. Đề xuất rằng bộ đệm lưu trữ các mẫu có chiều sâu bit cụ thể mà có thể khác với chiều sâu bit đối với dữ liệu video được mã hóa.

a. Trong một ví dụ, chiều sâu bit đối với bộ đệm tái tạo/dữ liệu video được mã hóa lớn hơn chiều sâu cho các mẫu tham chiếu IBC được lưu trữ trong bộ đệm.

b. Trong một ví dụ, thậm chí khi chiều sâu bit bên trong khác với chiều sâu bit đầu vào cho chuỗi video, chẳng hạn (10 bit với 8 bit), các mẫu tham chiếu IBC được lưu trữ để được căn chỉnh với chiều sâu bit đầu vào.

c. Trong một ví dụ, chiều sâu bit giống hệt chiều sâu của bộ đệm tái tạo.

d. Trong một ví dụ, chiều sâu bit giống hệt chiều sâu của ảnh/video đầu vào.

e. Trong một ví dụ, chiều sâu bit giống hệt số định trước.

- f. Trong một ví dụ, chiều sâu bit phụ thuộc vào tiểu sử của chuẩn.
- g. Trong một ví dụ, chiều sâu bit hoặc hiệu số chiều sâu bit so với chiều sâu bit đầu ra/chiều sâu bit đầu vào/chiều sâu bit bên trong có thể được báo hiệu trong SPS/PPS/tiêu đề chuỗi/tiêu đề ảnh/tiêu đề lát/Tiêu đề nhóm phiên/Tiêu đề phiên hoặc các loại khác của các đơn vị dữ liệu video.
- h. Các phương pháp được đề xuất có thể được áp dụng với các định nghĩa bộ đệm được đề xuất được nêu trong các mục khác, theo các cách, chúng cũng có thể được áp dụng cho thiết kế hiện tại của IBC.
- i. Chiều sâu bit của mỗi thành phần màu của bộ đệm có thể khác nhau.

Khởi tạo bộ đệm

- 8. Đề xuất khởi tạo bộ đệm với giá trị cụ thể
 - a. Trong một ví dụ, bộ đệm được khởi tạo với giá trị cụ thể.
 - i. Trong một ví dụ, giá trị cụ thể có thể phụ thuộc chiều sâu bit đầu vào và/hoặc chiều sâu bit bên trong.
 - ii. Trong một ví dụ, bộ đệm được khởi tạo với giá trị xấp xỉ giữa, chẳng hạn, 128 đối với tín hiệu 8 bit hoặc 512 cho tín hiệu 10 bit.
 - iii. Trong một ví dụ, bộ đệm được khởi tạo với $\text{forwardLUT}(m)$ khi ILR được sử dụng, chẳng hạn, $m = 1 \ll (\text{Bitdepth} - 1)$.
 - b. Theo các cách, bộ đệm được khởi tạo với giá trị được báo hiệu trong SPS/VPS/APS/PPS/tiêu đề chuỗi/Tiêu đề nhóm phiên/Tiêu đề ảnh /phiên/CTU/CU/VPDU/vùng.
 - c. Trong một ví dụ, giá trị cụ thể có thể được suy ra từ các mẫu của các ảnh được giải mã trước đó hoặc các lát hoặc các hàng CTU hoặc các CTU hoặc các CU.
 - d. Giá trị cụ thể có thể khác nhau cho thành phần màu khác nhau.
- 9. Theo các cách, đề xuất khởi tạo bộ đệm với các pixel được giải mã từ các khối được mã hóa trước đó.
 - a. Trong một ví dụ, các pixel được giải mã là các pixel trước khi lọc trong vòng.
 - b. Trong một ví dụ, khi kích thước bộ đệm là CTU, bộ đệm được khởi tạo với các pixel được giải mã của CTU được giải mã trước đó, nếu khả dụng.

c. Trong một ví dụ, khi kích thước bộ đệm là 64×64 , kích thước bộ đệm được khởi tạo với các pixel được giải mã của khối 64×64 được giải mã trước đó, nếu khả dụng.

d. Theo các cách, ngoài ra, nếu không khối được mã hóa nào trước đó khả dụng, các phương pháp trong mục 8 có thể được áp dụng.

Tham chiếu đến bộ đệm

10. Đối với khối sử dụng các pixel trong bộ đệm làm tham chiếu, có thể sử dụng vị trí (x, y) , $x=0, 1, 2, \dots, M-1; y=0, 1, 2, \dots, N-1$, trong bộ đệm để chỉ báo nơi có tham chiếu.

11. Theo các cách, vị trí tham chiếu có thể được ký hiệu là $l = y \cdot M + x$, $l=0, 1, \dots, M \cdot N - 1$.

12. Ký hiệu rằng vị trí trên bên trái của khối liên quan đến CTU hiện tại là (x_0, y_0) , vector khối $(BV_x, BV_y) = (x - x_0, y - y_0)$ có thể được gửi đến bộ giải mã để chỉ báo nơi có tham chiếu trong bộ đệm.

13. Theo các cách, vector khối (BV_x, BV_y) có thể được định nghĩa là $(x - x_0 + T_x, y - y_0 + T_y)$, trong đó, T_x và T_y là các độ lệch được định trước.

14. Đối với pixel (x_0, y_0) và (BV_x, BV_y) bất kỳ, tham chiếu của nó trong bộ đệm có thể được tìm thấy ở $(x_0 + BV_x, y_0 + BV_y)$

a. Trong một ví dụ, khi $(x_0 + BV_x, y_0 + BV_y)$ nằm ngoài bộ đệm, sẽ được xén đến biên.

b. Theo các cách, khi $(x_0 + BV_x, y_0 + BV_y)$ nằm ngoài bộ đệm, giá trị tham chiếu của nó được định trước là giá trị cụ thể, chẳng hạn, xám trung bình.

c. Theo các cách, vị trí tham chiếu được định nghĩa là $((x_0 + BV_x) \bmod M, (y_0 + BV_y) \bmod N)$, do vậy, nó luôn nằm trong bộ đệm.

15. Đối với pixel (x_0, y_0) và (BV_x, BV_y) bất kỳ, khi $(x_0 + BV_x, y_0 + BV_y)$ nằm ngoài bộ đệm, giá trị tham chiếu của nó có thể được suy ra từ các giá trị trong bộ đệm.

a. Trong một ví dụ, giá trị được suy ra từ mẫu $((x_0 + BV_x) \bmod M, (y_0 + BV_y) \bmod N)$ trong bộ đệm.

b. Trong một ví dụ, giá trị được suy ra từ mẫu $((x_0 + BV_x) \bmod M, \text{clip}(y_0 + BV_y, 0, N-1))$ trong bộ đệm.

c. Trong một ví dụ, giá trị được suy ra từ mẫu $(\text{clip}(x_0 + BV_x, 0, M-1), (y_0 + BV_y) \bmod N)$ trong bộ đệm.

d. Trong một ví dụ, giá trị được suy ra từ mẫu $(\text{clip}(x_0 + BV_x, 0, M-1), \text{clip}(y_0 + BV_y, 0, N-1))$ trong bộ đệm.

16. Có thể không cho phép tọa độ cụ thể nằm ngoài khoảng của bộ đệm

a. Trong một ví dụ, đối với pixel (x_0, y_0) bất kỳ so với góc trên bên trái của CTU và vector khối (BV_x, BV_y) , chính là giới hạn dòng bit $y_0 + BV_y$ nên nằm trong khoảng $[0, \dots, N-1]$.

b. Trong một ví dụ, đối với pixel (x_0, y_0) bất kỳ so với góc trên bên trái của CTU và vector khối (BV_x, BV_y) , chính là giới hạn dòng bit $x_0 + BV_x$ nên nằm trong khoảng của $[0, \dots, M-1]$.

c. Trong một ví dụ, đối với pixel (x_0, y_0) bất kỳ so với góc trên bên trái của CTU và vector khối (BV_x, BV_y) , chính là giới hạn dòng bit mà cả $y_0 + BV_y$ nên nằm trong khoảng của $[0, \dots, N-1]$ và $x_0 + BV_x$ nên nằm trong khoảng của $[0, \dots, M-1]$.

17. Khi vector khối được báo hiệu hoặc được suy ra của một khối trở đến đâu đó ngoài bộ đệm, đệm có thể được áp dụng theo bộ đệm.

a. Trong một ví dụ, giá trị của mẫu bất kỳ ngoài bộ đệm được định nghĩa có giá trị định trước.

i. Trong một ví dụ, giá trị có thể bằng $1 \ll (\text{Bitdepth} - 1)$, chẳng hạn, 128 đối với các tín hiệu 8 bit và 512 cho các tín hiệu 10 bit.

ii. Trong một ví dụ, giá trị có thể là $\text{forwardLUT}(m)$ khi ILR được sử dụng. Chẳng hạn, $m = 1 \ll (\text{Bitdepth} - 1)$.

iii. Theo các cách, chỉ báo giá trị định trước có thể được báo hiệu hoặc được chỉ báo ở mức SPS/PPS/tiêu đề chuỗi/tiêu đề ảnh/tiêu đề lát/Nhóm phiên/phiên/CTU/CU.

b. Trong một ví dụ, mẫu bất kỳ nằm ngoài bộ đệm được định nghĩa là giá trị của mẫu gần nhất trong bộ đệm.

18. Các phương pháp xử lý ngoài tham chiếu bộ đệm có thể khác nhau theo chiều ngang và chiều dọc hoặc có thể khác nhau theo vị trí của khối hiện tại (chẳng hạn, gần hơn với biên ảnh hay không).

a. Trong một ví dụ, khi $y_0 + BV_y$ nằm ngoài $[0, N-1]$, mẫu giá trị của $(x_0 + BV_x, y_0 + BV_y)$ được gán dưới dạng giá trị định trước.

b. Trong một ví dụ, khi $x_0 + BV_x$ nằm ngoài $[0, M-1]$, mẫu giá trị của $(x_0 + BV_x, y_0 + BV_y)$ được gán dưới dạng giá trị định trước.

c. Theo các cách, mẫu giá trị của $(x_0 + BV_x, y_0 + BV_y)$ được gán dưới dạng mẫu giá trị của $((x_0 + BV_x) \bmod M, y_0 + BV_y)$, có thể gọi phương pháp khác để suy ra tiếp giá trị nếu $((x_0 + BV_x) \bmod M, y_0 + BV_y)$ vẫn nằm ngoài bộ đệm.

d. Theo các cách, mẫu giá trị của $(x_0 + BV_x, y_0 + BV_y)$ được gán dưới dạng mẫu giá trị của $(x_0 + BV_x, (y_0 + BV_y) \bmod N)$, có thể gọi phương pháp khác để suy ra tiếp giá trị nếu $(x_0 + BV_x, (y_0 + BV_y) \bmod N)$ vẫn nằm ngoài bộ đệm.

Biểu diễn vector khối

19. Mỗi thành phần của vector khối (BV_x, BV_y) hoặc một trong các thành phần có thể được chuẩn hóa ở khoảng cụ thể.

a. Trong một ví dụ, BV_x có thể được thay thế bằng $(BV_x \bmod M)$.

b. Theo các cách, BV_x có thể được thay thế bằng $((BV_x + X) \bmod M) - X$, trong đó X là giá trị định trước.

i. Trong một ví dụ, X bằng 64.

ii. Trong một ví dụ, X bằng $M/2$;

iii. Trong một ví dụ, X bằng tọa độ ngang của khối so với CTU hiện tại.

c. Trong một ví dụ, BV_y có thể được thay thế bằng $(BV_y \bmod N)$.

d. Theo các cách, BV_y có thể được thay thế bằng $((BV_y + Y) \bmod N) - Y$, trong đó Y là giá trị định trước.

i. Trong một ví dụ, Y bằng 64.

ii. Trong một ví dụ, Y bằng $N/2$;

iii. Trong một ví dụ, Y là tọa độ dọc của khối so với CTU hiện tại.

20. BV_x và BV_y có thể có các khoảng được chuẩn hóa khác nhau.

21. Hiệu số vector khối (BVD_x, BVD_y) có thể được chuẩn hóa vào khoảng cụ thể.

a. Trong một ví dụ, BVD_x có thể được thay thế bằng $(BVD_x \bmod M)$ trong đó hàm mod trả về hàm nhắc.

b. Theo các cách, BVD_x có thể được thay thế bằng $((BVD_x + X) \bmod M) - X$, trong đó X là giá trị định trước.

i. Trong một ví dụ, X bằng 64.

ii. Trong một ví dụ, X bằng $M/2$;

c. Trong một ví dụ, BV_y có thể được thay thế bằng $(BVD_y \bmod N)$.

d. Theo các cách, BV_y có thể được thay thế bằng $((BVD_y + Y) \bmod N) - Y$, trong đó Y là giá trị định trước.

i. Trong một ví dụ, Y bằng 64.

ii. Trong một ví dụ, Y là $N/2$;

22. BVD_x và BVD_y có thể có các khoảng được chuẩn hóa khác nhau.

Kiểm tra tính hợp lệ cho vector khối

Ký hiệu chiều rộng và chiều cao của bộ đệm IBC là W_{buf} và H_{buf} . Đối với khối $W \times H$ (có thể là khối độ sáng, khối sắc độ, CU, TU, 4×4 , 2×2 , hoặc các khối phụ) bắt đầu từ (X, Y) so với góc trên bên trái của ảnh, phần sau có thể áp dụng để báo liệu vector khối (BV_x, BV_y) là hợp lệ hay không. Giả sử W_{pic} và H_{pic} là chiều rộng và chiều cao của ảnh và; W_{ctu} và H_{ctu} là chiều rộng và chiều cao của CTU. Hàm $\text{floor}(x)$ trả về số nguyên lớn nhất không lớn hơn x . Hàm $\text{isRec}(x, y)$ trả về nếu mẫu (x, y) đã được tái tạo.

23. Vector khối (BV_x, BV_y) có thể được gán là hợp lệ thậm chí nếu vị trí tham chiếu bất kỳ nằm ngoài biên ảnh.

a. Trong một ví dụ, vector khối có thể được gán là hợp lệ thậm chí nếu $X + BV_x < 0$.

b. Trong một ví dụ, vector khối có thể được gán là hợp lệ thậm chí nếu $X + W + BV_x > W_{pic}$.

c. Trong một ví dụ, vector khối có thể được gán là hợp lệ thậm chí nếu $Y + BV_y < 0$.

d. Trong một ví dụ, vector khối có thể được gán là hợp lệ thậm chí nếu $Y + H + BV_y > H_{pic}$.

24. Vector khối (BVx, BVy) có thể được gán là hợp lệ thậm chí nếu vị trí tham chiếu bất kỳ nằm ngoài hàng CTU hiện tại.

a. Trong một ví dụ, vector khối có thể được gán là hợp lệ thậm chí nếu $Y + BVy < \text{floor}(Y / H_{ctu}) * H_{ctu}$.

b. Trong một ví dụ, vector khối có thể được gán là hợp lệ thậm chí nếu $Y + H + BVy \geq \text{floor}(Y / H_{ctu}) * H_{ctu} + H_{ctu}$.

25. Vector khối (BVx, BVy) có thể được gán là hợp lệ thậm chí nếu vị trí tham chiếu bất kỳ nằm ngoài (n-1) CTU hiện tại và bên trái, trong đó n là số lượng CTU (bao gồm hoặc không bao gồm CTU hiện tại) có thể được sử dụng làm khu vực tham chiếu cho IBC.

a. Trong một ví dụ, vector khối có thể được gán là hợp lệ thậm chí nếu $X + BVx < \text{floor}(X / W_{ctu}) * W_{ctu} - (n-1) * W_{ctu}$.

b. Trong một ví dụ, vector khối có thể được gán là hợp lệ thậm chí nếu $X + W + BVx > \text{floor}(X / W_{ctu}) * W_{ctu} + W_{ctu}$.

26. Vector khối (BVx, BVy) có thể được gán là hợp lệ thậm chí nếu mẫu cụ thể đã không được tái tạo.

a. Trong một ví dụ, vector khối có thể được gán là hợp lệ thậm chí nếu $\text{isRec}(X + BVx, Y + BVy)$ là false.

b. Trong một ví dụ, vector khối có thể được gán là hợp lệ thậm chí nếu $\text{isRec}(X + BVx + W - 1, Y + BVy)$ là false.

c. Trong một ví dụ, vector khối có thể được gán là hợp lệ thậm chí nếu $\text{isRec}(X + BVx, Y + BVy + H - 1)$ là false.

d. Trong một ví dụ, vector khối có thể được gán là hợp lệ thậm chí nếu $\text{isRec}(X + BVx + W - 1, Y + BVy + H - 1)$ là false.

27. Vector khối (BVx, BVy) có thể luôn được thiết lập là hợp lệ khi khối không phải là CTU thứ nhất trong hàng CTU.

a. Theo các cách, vector khối có thể luôn được thiết lập là hợp lệ.

28. Vector khối (BVx, BVy) có thể luôn được thiết lập là hợp lệ khi 3 điều kiện sau đều được thỏa mãn

- $X + BVx \geq 0$
- $Y + BVy \geq \text{floor}(Y / H_{ctu})$

- $\text{isRec}(X + BV_x + W - 1, Y + BV_y + H - 1) == \text{true}$

a. Theo các cách, khi ba điều kiện đều được thỏa mãn cho khối của CTU thứ nhất trong hàng CTU, vector khối có thể luôn được thiết lập là hợp lệ.

29. Khi vector khối (BV_x, BV_y) là hợp lệ, bước sao chép mẫu đối với khối có thể dựa trên vector khối.

a. Trong một ví dụ, dự báo của mẫu (x, y) có thể là từ $((X + BV_x) \% W_{\text{buf}}, (Y + BV_y) \% H_{\text{buf}})$

Cập nhật bộ đệm

30. Khi mã hóa ảnh hoặc phiên mới, bộ đệm có thể được thiết lập lại.

a. Cụm từ “thiết lập lại” có thể đề cập việc bộ đệm được khởi tạo.

b. Cụm từ “thiết lập lại” có thể đề cập việc tất cả các mẫu/pixel trong bộ đệm được gán bằng giá trị cụ thể (chẳng hạn, 0 hoặc -1).

31. Khi hoàn thành mã hóa VPDU, bộ đệm có thể được cập nhật với các giá trị được tái tạo của VPDU.

32. Khi hoàn thành mã hóa của CTU, bộ đệm có thể được cập nhật với các giá trị được tái tạo của CTU.

a. Trong một ví dụ, khi bộ đệm không đầy, bộ đệm có thể được cập nhật tuần tự theo từng CTU.

b. Trong một ví dụ, khi bộ đệm đầy, khu vực bộ đệm tương ứng với CTU cũ nhất sẽ được cập nhật.

c. Trong một ví dụ, khi $M = mW$ và $N = H$ (W và H là kích thước CTU; M và N là kích thước bộ đệm) và khu vực được cập nhật trước đó được bắt đầu từ $(kW, 0)$, vị trí bắt đầu tiếp theo để cập nhật sẽ là $((k+1)W \bmod M, 0)$.

33. Bộ đệm có thể được thiết lập lại ở bắt đầu của mỗi hàng CTU.

a. Theo các cách, bộ đệm có thể được thiết lập lại ở bắt đầu của giải mã mỗi CTU.

b. Theo các cách, bộ đệm có thể được thiết lập lại ở bắt đầu của giải mã một phiên.

c. Theo các cách, bộ đệm có thể được thiết lập lại ở bắt đầu của giải mã một nhóm phiên/ảnh.

34. Khi hoàn thành mã hóa khối bắt đầu từ (x,y) , khu vực tương ứng của bộ đệm, bắt đầu từ (x,y) sẽ được cập nhật với việc tái tạo từ khối.

a. Trong một ví dụ, (x,y) là vị trí so với góc trên bên trái của CTU.

35. Khi hoàn thành mã hóa khối so với ảnh, khu vực tương ứng của bộ đệm sẽ được cập nhật với việc tái tạo từ khối.

a. Trong một ví dụ, giá trị ở vị trí $(x \bmod M, y \bmod N)$ trong bộ đệm có thể được cập nhật với giá trị pixel được tái tạo của vị trí (x, y) so với góc trên bên trái của ảnh.

b. Trong một ví dụ, giá trị ở vị trí $(x \bmod M, y \bmod N)$ trong bộ đệm có thể được cập nhật với giá trị pixel được tái tạo của vị trí (x, y) so với góc trên bên trái của phiên hiện tại.

c. Trong một ví dụ, giá trị ở vị trí $(x \bmod M, y \bmod N)$ trong bộ đệm có thể được cập nhật với giá trị pixel được tái tạo của vị trí (x, y) so với góc trên bên trái của hàng CTU hiện tại.

d. Trong một ví dụ, giá trị trong bộ đệm có thể được cập nhật với các giá trị pixel được tái tạo sau khi căn chỉnh chiều sâu bit.

36. Khi hoàn thành mã hóa khối bắt đầu từ (x,y) , khu vực tương ứng của bộ đệm, bắt đầu từ (x_b, y_b) sẽ được cập nhật với việc tái tạo từ khối trong đó (x_b, y_b) và (x, y) là hai tọa độ khác nhau

a. Trong một ví dụ, (x,y) là vị trí liên quan đến góc trên bên trái của CTU, và (x_b, y_b) là $(x + \text{update_x}, y + \text{update_y})$, trong đó update_x và update_y trở đến vị trí có thể cập nhật trong bộ đệm.

37. Đối với các ví dụ nêu trên, các giá trị được tái tạo của khối có thể chỉ báo các giá trị được tái tạo trước khi các bộ lọc (chẳng hạn, lọc tách khối) được áp dụng.

a. Theo các cách, các giá trị được tái tạo của khối có thể chỉ báo các giá trị được tái tạo sau khi các bộ lọc (chẳng hạn, lọc tách khối) được áp dụng.

38. Khi bộ đệm được cập nhật từ các mẫu được tái tạo, các mẫu được tái tạo có thể được chỉnh sửa trước trước khi được lưu trữ, chẳng hạn chiều sâu bit mẫu có thể được thay đổi.

- a. Trong một ví dụ, bộ đệm được cập nhật có giá trị mẫu được tái tạo sau khi căn chỉnh chiều sâu bit với chiều sâu bit của bộ đệm.
- b. Trong một ví dụ, giá trị bộ đệm được cập nhật theo giá trị $\{p+[1\ll(b-1)]\}\gg b$, trong đó p là giá trị mẫu được tái tạo, b là giá trị dịch bit định trước.
- c. Trong một ví dụ, giá trị bộ đệm được cập nhật theo giá trị $\text{clip}(\{p+[1\ll(b-1)]\}\gg b, 0, (1\ll\text{bitdepth})-1)$, trong đó p là giá trị mẫu được tái tạo, b là giá trị dịch bit định trước, bitdepth là chiều sâu bit bộ đệm.
- d. Trong một ví dụ, giá trị bộ đệm được cập nhật theo giá trị $\{p+[1\ll(b-1)]\}\gg b$, trong đó p là giá trị mẫu được tái tạo, b là giá trị dịch bit định trước.
- e. Trong một ví dụ, giá trị bộ đệm được cập nhật theo giá trị $\text{clip}(\{p+[1\ll(b-1)]\}\gg b, 0, (1\ll\text{bitdepth})-1)$, trong đó p là giá trị mẫu được tái tạo, b là giá trị dịch bit định trước, bitdepth là chiều sâu bit bộ đệm.
- f. Trong một ví dụ, giá trị bộ đệm được cập nhật theo giá trị $p\gg b$.
- g. Trong một ví dụ, giá trị bộ đệm được cập nhật theo giá trị $\text{clip}(p\gg b, 0, (1\ll\text{bitdepth})-1)$, trong đó bitdepth là chiều sâu bit bộ đệm.
- h. Ở các ví dụ nêu trên, b có thể là chiều sâu bit được tái tạo trừ đi chiều sâu bit mẫu đầu vào.

39. Khi sử dụng các mẫu bộ đệm để tạo dự báo, bước tiền xử lý có thể được áp dụng.

- a. Trong một ví dụ, giá trị dự báo là $p\ll b$, trong đó p là giá trị mẫu trong bộ đệm, và b là giá trị định trước.
- b. Trong một ví dụ, giá trị dự báo là $\text{clip}(p\ll b, 0, 1\ll\text{bitdepth})$, trong đó bitdepth là chiều sâu bit cho các mẫu tái tạo.
- c. Trong một ví dụ, giá trị dự báo là $(p\ll b)+(1\ll(\text{bitdepth}-1))$, trong đó p là giá trị mẫu trong bộ đệm, và b là giá trị định trước, bitdepth là chiều sâu bit cho các mẫu tái tạo.
- d. Ở các ví dụ nêu trên, b có thể là chiều sâu bit được tái tạo trừ chiều sâu bit mẫu đầu vào.

40. Bộ đệm có thể được cập nhật theo thứ tự cụ thể.

- a. Trong một ví dụ, bộ đệm có thể được cập nhật tuần tự.

b. Trong một ví dụ, bộ đệm có thể được cập nhật theo thứ tự của các khối được tái tạo.

41. Khi bộ đệm đầy, các mẫu trong bộ đệm có thể được thay thế bằng các mẫu được tái tạo mới nhất.

a. Trong một ví dụ, các mẫu có thể được cập nhật theo cách vào trước ra trước (FIFO).

b. Trong một ví dụ, các mẫu cũ nhất sẽ được thay thế.

c. Trong một ví dụ, các mẫu có thể được gán độ ưu tiên và được thay thế theo độ ưu tiên.

d. Trong một ví dụ, các mẫu có thể được đánh dấu là “dài hạn” sao cho các mẫu khác sẽ được thay thế trước.

e. Trong một ví dụ, cờ có thể được gửi cùng với khối để chỉ báo độ ưu tiên cao.

f. Trong một ví dụ, số có thể được gửi cùng với khối để chỉ báo độ ưu tiên.

g. Trong một ví dụ, các mẫu từ khối được tái tạo có đặc tính cụ thể sẽ được gán độ ưu tiên cao hơn sao cho các mẫu khác sẽ được thay thế trước.

i. Trong một ví dụ, khi phần trăm mẫu được mã hóa trong chế độ IBC lớn hơn ngưỡng, tất cả các mẫu của khối có thể được gán độ ưu tiên cao.

ii. Trong một ví dụ, khi phần trăm mẫu được mã hóa trong chế độ bảng màu lớn hơn ngưỡng, tất cả các mẫu của khối có thể được gán độ ưu tiên cao.

iii. Trong một ví dụ, khi phần trăm mẫu được mã hóa ở chế độ IBC hoặc bảng màu lớn hơn ngưỡng, tất cả các mẫu của khối có thể được gán độ ưu tiên cao.

iv. Trong một ví dụ, khi phần trăm mẫu được mã hóa trong chế độ bỏ qua biến đổi lớn hơn ngưỡng, tất cả các mẫu của khối có thể được gán độ ưu tiên cao.

v. Ngưỡng có thể khác nhau theo kích thước khối, thành phần màu, kích thước CTU.

vi. Ngưỡng có thể được báo hiệu trong SPS/ PPS/tiêu đề chuỗi/tiêu đề lát/Nhóm phiên/mức phiên/vùng.

h. Trong một ví dụ, việc bộ đệm đầy có thể nghĩa là số lượng mẫu khả dụng trong bộ đệm bằng hoặc lớn hơn ngưỡng cụ thể.

i. Trong một ví dụ, khi số lượng mẫu khả dụng trong bộ đệm bằng hoặc lớn hơn $64 \times 64 \times 3$ các mẫu độ sáng, bộ đệm có thể được xác định là đầy.

Tổ hợp bộ đệm luân phiên

42. Thay cho việc luôn sử dụng ba khối 64×64 được mã hóa trước như là vùng tham chiếu, đề xuất thay đổi thích ứng nó dựa trên vị trí của khối hiện tại (hoặc VPDU).

a. Trong một ví dụ, khi mã hóa/giải mã khối 64×64 , 3 khối 64×64 trước đó có thể được sử dụng làm tham chiếu. So với Fig.2, nhiều loại tổ hợp hơn của các khối 64×64 trước đó có thể được áp dụng. Fig.2 là hình vẽ thể hiện ví dụ tổ hợp khác của các khối 64×64 trước đó.

43. Thay cho việc sử dụng thứ tự quét z, thứ tự quét dọc có thể được sử dụng thay thế.

a. Trong một ví dụ, khi một khối được tách thành 4 VPDU có chỉ số 0..3 theo thứ tự quét z, thứ tự mã hóa/giải mã là 0, 2, 1, 3.

b. Trong một ví dụ, khi mã hóa/giải mã các khối 64×64 , 3 khối 64×64 trước đó có thể được sử dụng làm tham chiếu. So với Fig.2, nhiều loại thứ tự mã hóa/giải mã của các khối 64×64 có thể được áp dụng. Fig.4 là hình vẽ thể hiện ví dụ thứ tự mã hóa/giải mã khác của các khối 64×64 s.

c. Theo các cách, các phương pháp nêu trên có thể được áp dụng chỉ cho mã hóa nội dung màn hình

d. Theo các cách, các phương pháp nêu trên có thể được áp dụng chỉ khi CPR được kích hoạt cho một phiên/nhóm phiên/ảnh.

e. Theo các cách, các phương pháp nêu trên có thể được áp dụng chỉ khi CPR được kích hoạt cho một CTU hoặc một hàng CTU.

Bộ đệm IBC ảo

Dưới đây, chiều rộng và chiều cao của VPDU lần lượt được ký hiệu là W_{VPDU} (chẳng hạn, 64) và H_{VPDU} (chẳng hạn, 64), trong các mẫu độ sáng. Theo các cách, W_{VPDU} và/hoặc H_{VPDU} có thể ký hiệu chiều rộng và/hoặc chiều cao của đơn vị video khác (chẳng hạn, CTU).

44. Bộ đệm ảo có thể được duy trì để giám sát trạng thái vùng tham chiếu IBC.

- a. Trong một ví dụ, kích thước bộ đệm ảo là $mW_{VPDU} \times nH_{VPDU}$.
 - i. Trong một ví dụ, m bằng 3 và n bằng 2.
 - ii. Trong một ví dụ, m và/hoặc n có thể phụ thuộc độ phân giải ảnh, các kích thước CTU.
 - iii. Trong một ví dụ, m và/hoặc n có thể được báo hiệu hoặc được định trước.
- b. Trong một ví dụ, các phương pháp được mô tả ở các mục và các mục phụ nêu trên có thể được áp dụng cho bộ đệm ảo.
- c. Trong một ví dụ, mẫu (x, y) so với góc trên bên trái của ảnh/lát/phiên/viên có thể được ánh xạ đến $(x\%(mW_{VPDU}), y\%(nH_{VPDU}))$

45. Mảng có thể được sử dụng để theo dõi tính khả dụng của mỗi mẫu được liên kết với bộ đệm ảo.

- a. Trong một ví dụ, cờ có thể được liên kết với mẫu in bộ đệm ảo để xác định nếu mẫu trong bộ đệm có thể được sử dụng làm tham chiếu IBC hay không.
- b. Trong một ví dụ, mỗi khối 4×4 chứa các mẫu độ sáng và sắc độ có thể chia sẻ cờ để chỉ báo liệu các mẫu bất kỳ có được liên kết với khối có thể được sử dụng làm tham chiếu IBC hay không.

46. Sau khi hoàn thành giải mã VPDU hoặc đơn vị video, các mẫu cụ thể được liên kết với bộ đệm ảo có thể được đánh dấu là không khả dụng cho tham chiếu IBC.

- a. Trong một ví dụ, các mẫu nào có thể được đánh dấu là không khả dụng phụ thuộc vị trí của VPDU được giải mã gần đây nhất.
 - b. Khi một mẫu được đánh dấu là không khả dụng, dự báo từ mẫu không được phép.
 - i. Theo các cách, các cách (chẳng hạn, nhờ sử dụng các giá trị mặc định) có thể còn được áp dụng để suy ra bộ dự báo để thay thế mẫu không khả dụng.
47. Vị trí của VPDU được giải mã gần đây nhất có thể được ghi nhận để giúp dễ nhận diện các mẫu nào được liên kết với bộ đệm ảo có thể được đánh dấu là không khả dụng.

a. Trong một ví dụ, lúc bắt đầu giải mã VPDU, các mẫu cụ thể được liên kết với bộ đệm ảo có thể được đánh dấu là không khả dụng theo vị trí của VPDU được giải mã gần đây nhất.

i. Trong một ví dụ, ký hiệu ($x_{PrevVPDU}$, $y_{PrevVPDU}$) như là vị trí trên bên trái so với góc trên bên trái của ảnh/lát/phiên/viên/đơn vị xử lý video khác của VPDU được giải mã gần đây nhất, nếu $y_{PrevVPDU} \% (n_{HVPDU})$ bằng 0, các vị trí (x , y) cụ thể có thể được đánh dấu là không khả dụng.

1. Trong một ví dụ, x có thể nằm trong khoảng, chẳng hạn $[x_{PrevVPDU} - 2WVPDU + 2mWVPDU) \% mWVPDU, ((x_{PrevVPDU} - 2WVPDU + 2mWVPDU) \% mWVPDU) - 1 + WVPDU]$;

2. Trong một ví dụ, y có thể nằm trong khoảng, chẳng hạn $[y_{PrevVPDU} \% (n_{HVPDU}), (y_{PrevVPDU} \% (n_{HVPDU})) - 1 + HVPDU]$;

3. Trong một ví dụ, x có thể nằm trong khoảng, chẳng hạn $[x_{PrevVPDU} - 2WVPDU + 2mWVPDU) \% mWVPDU, ((x_{PrevVPDU} - 2WVPDU + 2mWVPDU) \% mWVPDU) - 1 + WVPDU]$ và y có thể nằm trong khoảng, chẳng hạn $[y_{PrevVPDU} \% (n_{HVPDU}), (y_{PrevVPDU} \% (n_{HVPDU})) - 1 + HVPDU]$.

ii. Trong một ví dụ, ký hiệu ($x_{PrevVPDU}$, $y_{PrevVPDU}$) như là vị trí trên bên trái so với góc trên bên trái của ảnh/lát/phiên/viên/khối xử lý video khác của VPDU được giải mã gần đây nhất, nếu $y_{PrevVPDU} \% (n_{HVPDU})$ không bằng 0, các vị trí (x , y) cụ thể có thể được đánh dấu là không khả dụng.

1. Trong một ví dụ, x có thể nằm trong khoảng, chẳng hạn $[x_{PrevVPDU} - WVPDU + 2mWVPDU) \% mWVPDU, ((x_{PrevVPDU} - WVPDU + 2mWVPDU) \% mWVPDU) - 1 + WVPDU]$;

2. Trong một ví dụ, y có thể nằm trong khoảng, chẳng hạn $[y_{PrevVPDU} \% (n_{HVPDU}), (y_{PrevVPDU} \% (n_{HVPDU})) - 1 + HVPDU]$

3. Trong một ví dụ, x có thể nằm trong khoảng, chẳng hạn $[x_{PrevVPDU} - WVPDU + 2mWVPDU) \% mWVPDU, ((x_{PrevVPDU} - WVPDU + 2mWVPDU) \% mWVPDU) - 1 + WVPDU]$ và y có thể nằm trong khoảng, chẳng hạn $[y_{PrevVPDU} \% (n_{HVPDU}), (y_{PrevVPDU} \% (n_{HVPDU})) - 1 + HVPDU]$.

48. Khi CU chứa nhiều VPDU, thay cho việc áp dụng quá trình đánh dấu tính khả dụng tham chiếu IBC theo VPDU, quá trình đánh dấu tính khả dụng tham chiếu IBC có thể theo CU

a. Trong một ví dụ, ở bắt đầu của giải mã CU chứa nhiều VPDU, quá trình đánh dấu tính khả dụng tham chiếu IBC có thể được áp dụng cho mỗi VPDU trước VPDU trong CU được giải mã.

b. Trong trường hợp này, các khối IBC 128x64 và 64x128 IBC có thể không được phép.

i. Trong một ví dụ, `pred_mode_ibc_flag` cho các CU 128x64 và 64x128 có thể không được gửi và có thể được suy ra bằng 0.

49. Cho khối tham chiếu hoặc khối phụ, trạng thái khả dụng tham chiếu của góc trên bên phải có thể không cần được kiểm tra để báo liệu vector khối được liên kết với khối tham chiếu là hợp lệ hay không.

a. Trong một ví dụ, chỉ góc trên bên trái, dưới bên trái và dưới bên phải của khối/khối phụ sẽ được kiểm tra để báo liệu vector khối là hợp lệ hay không.

50. Kích thước bộ đệm IBC có thể phụ thuộc kích thước VPDU (trong đó chiều rộng/chiều cao được ký hiệu bởi `vSize`) và/hoặc CTB/kích thước CTU (trong đó chiều rộng/chiều cao được ký hiệu bởi `ctbSize`)

a. Trong một ví dụ, chiều cao của bộ đệm có thể bằng `ctbSize`.

b. Trong một ví dụ, chiều rộng của bộ đệm có thể phụ thuộc $\min(\text{ctbSize}, 64)$

i. Trong một ví dụ, chiều rộng của bộ đệm có thể là $(128 \times 128 / \text{vSize}, \min(\text{ctbSize}, 64))$

51. Bộ đệm IBC có thể chứa các giá trị nằm ngoài khoảng pixel, mà chỉ báo rằng vị trí có thể không khả dụng cho tham chiếu IBC, chẳng hạn, không được sử dụng để dự báo các mẫu khác.

a. Giá trị mẫu có thể được gán bằng giá trị mà chỉ báo mẫu không khả dụng.

b. Trong một ví dụ, giá trị có thể bằng -1.

c. Trong một ví dụ, giá trị có thể là giá trị bất kỳ nằm ngoài $[0, 1 \ll (\text{internal_bit_depth}) - 1]$ trong đó `internal_bit_depth` là giá trị số nguyên

ương. Chẳng hạn, `internal_bit_depth` chiều sâu bit trong được sử dụng cho mã hóa/giải mã mẫu cho thành phần màu sắc.

d. Trong một ví dụ, giá trị có thể là giá trị bất kỳ nằm ngoài $[0, 1 << (\text{input_bit_depth}) - 1]$ trong đó `input_bit_depth` là giá trị số nguyên dương. Chẳng hạn, `input_bit_depth` là chiều sâu bit vào được sử dụng cho mã hóa/giải mã mẫu cho thành phần màu sắc.

52. Bước đánh dấu tính khả dụng cho các mẫu trong bộ đệm IBC có thể phụ thuộc vị trí của khối hiện tại, kích thước của khối hiện tại, kích thước CTU/CTB và kích thước VPDU. Trong một ví dụ, giả sử (x_{Cb}, y_{Cb}) ký hiệu vị trí của khối so với phía trên bên trái của ảnh; `ctbSize` là kích thước (tức là, chiều rộng và/hoặc chiều cao) của CTU/CTB; $vSize = \min(ctbSize, 64)$; `wIbcBuf` và `hIbcBuf` là chiều rộng và chiều cao bộ đệm IBC.

a. Trong một ví dụ, nếu $(x_{Cb} \% vSize)$ bằng 0 và $(y_{Cb} \% vSize)$ bằng 0, tập hợp cụ thể của các vị trí trong bộ đệm IBC có thể được đánh dấu là không khả dụng.

b. Trong một ví dụ, khi kích thước khối hiện tại nhỏ hơn kích thước VPDU, tức là, $\min(ctbSize, 64)$, vùng được đánh dấu là không khả dụng có thể theo kích thước VPDU.

c. Trong một ví dụ, khi kích thước khối hiện tại lớn hơn kích thước VPDU, tức là, $\min(ctbSize, 64)$, vùng được đánh dấu là không khả dụng có thể theo kích thước CU.

53. Lúc bắt đầu giải mã đơn vị video (chẳng hạn, VPDU (x_V, y_V)) so với vị trí trên bên trái của ảnh, các vị trí tương ứng trong bộ đệm IBC có thể được gán bằng giá trị ngoài khoảng pixel.

a. Trong một ví dụ, các mẫu bộ đệm có vị trí $(x \% wIbcBuf, y \% hIbcBuf)$ trong bộ đệm, có $x = x_V, \dots, x_V + ctbSize - 1$ và $y = y_V, \dots, y_V + ctbSize - 1$, sẽ được gán bằng giá trị -1. Trong đó, `wIbcBuf` và `hIbcBuf` là chiều rộng và chiều cao bộ đệm IBC, `ctbSize` là chiều rộng của CTU/CTB.

i. Trong một ví dụ, `hIbcBuf` có thể bằng `ctbSize`.

54. Giới hạn tương thích dòng bit có thể theo giá trị của mẫu trong bộ đệm IBC

a. Trong một ví dụ, nếu khối tham chiếu liên kết với vector khối trong bộ đệm IBC chứa giá trị ngoài khoảng pixel, dòng bit có thể là bất hợp pháp.

55. Giới hạn tương thích dòng bit có thể được thiết lập theo chỉ báo tính khả dụng trong bộ đệm IBC.

a. Trong một ví dụ, nếu mẫu tham chiếu bất kỳ được ánh xạ trong bộ đệm IBC được đánh dấu là không khả dụng để mã hóa/giải mã khối, dòng bit có thể là bất hợp pháp.

b. Trong một ví dụ, khi cây đơn được sử dụng, nếu mẫu tham chiếu độ sáng bất kỳ được ánh xạ trong bộ đệm IBC để mã hóa/giải mã khối được đánh dấu là không khả dụng, dòng bit có thể là bất hợp pháp.

c. Dòng bit tương thích có thể thỏa mãn rằng đối với khối được mã hóa IBC, vector khối được liên kết có thể trỏ đến khối tham chiếu được ánh xạ trong bộ đệm IBC và mỗi mẫu tham chiếu độ sáng được đặt ở bộ đệm IBC để mã hóa/giải mã khối sẽ được đánh dấu là khả dụng (chẳng hạn, các giá trị của các mẫu nằm trong khoảng của $[K0, K1]$ trong đó chẳng hạn, $K0$ được gán bằng 0 và $K1$ được gán bằng $(1 \ll \text{BitDepth} - 1)$ trong đó BitDepth là chiều sâu bit bên trong hoặc chiều sâu bit đầu vào).

56. Giới hạn tương thích dòng bit có thể phụ thuộc phân vùng các loại cây và loại cây mã hóa của CU hiện tại

a. Trong một ví dụ, nếu cây đôi được phép trong mức cao (chẳng hạn, lát/ảnh/viên/phiến) và khối video hiện tại (chẳng hạn, CU/PU/CB/PB) được tạo mã bằng cây đơn, các dòng bit giới hạn có thể cần kiểm tra nếu tất cả các vị trí của các thành phần được ánh xạ trong bộ đệm IBC được đánh dấu là không khả dụng hay không.

b. Trong một ví dụ, nếu cây đôi được phép ở mức cao (chẳng hạn, lát/ảnh/viên/phiến) và khối video độ sáng hiện tại (chẳng hạn, CU/PU/CB/PB) được tạo mã bằng cây đôi, các dòng bit giới hạn có thể bỏ qua các vị trí của thành phần sắc độ được ánh xạ trong bộ đệm IBC được đánh dấu là không khả dụng hay không.

i. Theo các cách, trong trường hợp này, các dòng bit giới hạn có thể vẫn kiểm tra các vị trí của tất cả các thành phần được ánh xạ trong bộ đệm IBC được đánh dấu là không khả dụng hay không.

c. Trong một ví dụ, nếu cây đơn được sử dụng, các dòng bit giới hạn có thể bỏ qua các vị trí của thành phần sắc độ được ánh xạ trong bộ đệm IBC được đánh dấu là không khả dụng hay không.

Cải thiện với thiết kế VTM hiện tại

57. Dự báo cho IBC có thể có độ chính xác thấp hơn tái tạo.

a. Trong một ví dụ, giá trị dự báo là theo giá trị $\text{clip}\{p+[1 \ll (b-1)]\} \gg b, 0, (1 \ll (\text{bitdepth})-1) \ll b$, trong đó p là giá trị mẫu được tái tạo, b là giá trị dịch bit định trước, bitdepth là chiều sâu bit mẫu dự báo.

b. Trong một ví dụ, giá trị dự báo là theo giá trị $\text{clip}\{p+[1 \ll (b-1)]\} \gg b, 0, (1 \ll (\text{bitdepth})-1) \ll b$, trong đó p là giá trị mẫu được tái tạo, b là giá trị dịch bit định trước.

c. Trong một ví dụ, giá trị dự báo là theo giá trị $((p \gg b) + (1 \ll (\text{bitdepth}-1))) \ll b$, trong đó bitdepth là chiều sâu bit mẫu dự báo.

d. Trong một ví dụ, giá trị dự báo là theo giá trị $(\text{clip}((p \gg b), 0, (1 \ll (\text{bitdepth}-b)))) + (1 \ll (\text{bitdepth}-1))) \ll b$, trong đó bitdepth là chiều sâu bit mẫu dự báo.

e. Trong một ví dụ, giá trị dự báo được xén theo các cách tùy thuộc liệu ILR được áp dụng hay không.

f. Trong các ví dụ nêu trên, b có thể là chiều sâu bit được tái tạo trừ đi chiều sâu bit mẫu đầu vào.

g. Trong một ví dụ, chiều sâu bit hoặc hiệu số chiều sâu bit so với chiều sâu bit đầu ra/chiều sâu bit đầu vào/chiều sâu bit bên trong có thể được báo hiệu trong SPS/PPS/tiêu đề chuỗi/tiêu đề ảnh/tiêu đề lát/Tiêu đề nhóm phiên/Tiêu đề phiên hoặc các loại đơn vị dữ liệu video khác.

58. Phần dự báo của IBC có thể có độ chính xác thấp hơn và phần còn lại có cùng độ chính xác với tái tạo.

a. Trong một ví dụ, khu vực tham chiếu được phép có thể chứa các mẫu có các độ chính xác khác nhau (chẳng hạn, chiều sâu bit).

b. Trong một ví dụ, tham chiếu từ các khối 64x64 khác với khối 64x64 hiện tại được giải mã có độ chính xác thấp và tham chiếu từ khối 64x64 hiện tại có cùng độ chính xác với tái tạo.

c. Trong một ví dụ, tham chiếu từ các CTU khác ngoài CTU hiện tại được giải mã có độ chính xác thấp và tham chiếu từ CTU hiện tại có cùng độ chính xác với tái tạo.

d. Trong một ví dụ, tham chiếu từ tập hợp cụ thể của các thành phần màu có độ chính xác thấp và tham chiếu từ các thành phần màu khác có cùng độ chính xác với tái tạo.

59. Khi kích thước CTU bằng $M \times M$ và kích thước khu vực tham chiếu là $nM \times nM$, khu vực tham chiếu là CTU $n \times n$ khả dụng gần nhất trong hàng CTU.

a. Trong một ví dụ, khi kích thước khu vực tham chiếu là 128x128 và kích thước CTU bằng 64x64, 4 CTU khả dụng gần nhất trong hàng CTU có thể được sử dụng cho tham chiếu IBC.

b. Trong một ví dụ, khi kích thước khu vực tham chiếu bằng 128x128 và kích thước CTU bằng 32x32, 16 CTU khả dụng gần nhất trong hàng CTU có thể được sử dụng cho tham chiếu IBC.

60. Khi kích thước CTU bằng M và kích thước khu vực tham chiếu là nM , khu vực tham chiếu là $n-1$ CTU khả dụng gần nhất trong hàng CTU/phiên.

a. Trong một ví dụ, khi kích thước khu vực tham chiếu là 128x128 hoặc 256x64 và kích thước CTU bằng 64x64, 3 CTU khả dụng gần nhất trong hàng CTU có thể được sử dụng cho tham chiếu IBC.

b. Trong một ví dụ, khi kích thước khu vực tham chiếu là 128x128 hoặc 512x32 và kích thước CTU bằng 32x32, 15 CTU khả dụng gần nhất trong hàng CTU có thể được sử dụng cho tham chiếu IBC.

61. Khi kích thước CTU bằng M , kích thước VPDU là kM và kích thước khu vực tham chiếu là nM , và khu vực tham chiếu là $n-k$ CTU khả dụng gần nhất trong hàng CTU/tile.

a. Trong một ví dụ, kích thước CTU bằng 64×64 , kích thước VPDU cũng bằng 64×64 , kích thước khu vực tham chiếu là 128×128 , 3 CTU gần nhất trong hàng CTU có thể được sử dụng cho tham chiếu IBC.

b. Trong một ví dụ, kích thước CTU bằng 32×32 , kích thước VPDU cũng bằng 64×64 , kích thước khu vực tham chiếu là 128×128 , $(16-4)=12$ CTU gần nhất trong hàng CTU có thể được sử dụng cho tham chiếu IBC.

62. Đối với khối $w \times h$ có góc trên bên trái là (x, y) nhờ sử dụng IBC, có giới hạn giữ khối tham chiếu từ khu vực cụ thể để sử dụng lại bộ nhớ, trong đó W và H là chiều rộng và chiều cao của khối hiện tại.

a. Trong một ví dụ, khi kích thước CTU bằng 128×128 và $(x, y) = (m \times 64, n \times 64)$, khối tham chiếu không thể chồng lấp với vùng 64×64 bắt đầu từ $((m-2) \times 64, n \times 64)$.

b. Trong một ví dụ, khi kích thước CTU bằng 128×128 , khối tham chiếu không thể chồng lấp với khối $w \times h$ có góc trên bên trái bằng $(x-128, y)$.

c. Trong một ví dụ, khi kích thước CTU bằng 128×128 , $(x+B_{Vx}, y+B_{Vy})$ không thể nằm trong khối $w \times h$ có góc trên bên trái bằng $(x-128, y)$, trong đó B_{Vx} và B_{Vy} ký hiệu vector khối cho khối hiện tại.

d. Trong một ví dụ, khi kích thước CTU bằng $M \times M$ và kích thước bộ đệm IBC là $k \times M \times M$, khối tham chiếu không thể chồng lấp với khối $w \times h$ có góc trên bên trái bằng $(x-k \times M, y)$, trong đó B_{Vx} và B_{Vy} ký hiệu vector khối cho khối hiện tại.

e. Trong một ví dụ, khi kích thước CTU bằng $M \times M$ và kích thước bộ đệm IBC là $k \times M \times M$, $(x+B_{Vx}, y+B_{Vy})$ không thể nằm trong khối $w \times h$ có góc trên bên trái là $(x-k \times M, y)$, trong đó B_{Vx} và B_{Vy} ký hiệu vector khối cho khối hiện tại.

63. Khi kích thước CTU không bằng $M \times M$ và kích thước khu vực tham chiếu là $nM \times nM$, khu vực tham chiếu là $n \times n - 1$ CTU khả dụng gần nhất trong hàng CTU.

a. Trong một ví dụ, khi kích thước khu vực tham chiếu là 128×128 và kích thước CTU bằng 64×64 , 3 CTU gần nhất khả dụng trong hàng CTU có thể được sử dụng cho tham chiếu IBC.

b. Trong một ví dụ, khi kích thước khu vực tham chiếu là 128×128 và kích thước CTU bằng 32×32 , 15 CTU gần nhất khả dụng trong hàng CTU có thể được sử dụng cho tham chiếu IBC.

64. Đối với CU trong khối 64×64 bắt đầu từ $(2m \times 64, 2n \times 64)$, tức là, khối trên bên trái 64×64 trong CTU 128×128 , dự báo IBC của nó có thể từ được tái tạo các mẫu trong khối 64×64 bắt đầu từ $((2m-2) \times 64, 2n \times 64)$, khối 64×64 bắt đầu từ $((2m-1) \times 64, 2n \times 64)$, khối 64×64 bắt đầu từ $((2m-1) \times 64, (2n+1) \times 64)$ và khối 64×64 hiện tại.

65. Đối với CU trong khối 64×64 bắt đầu từ $((2m+1) \times 64, (2n+1) \times 64)$, tức là, khối dưới bên phải 64×64 trong CTU 128×128 , dự báo IBC của nó có thể từ CTU 128×128 hiện tại.

66. Đối với CU trong khối 64×64 bắt đầu từ $((2m+1) \times 64, 2n \times 64)$, tức là, khối trên bên phải 64×64 trong CTU 128×128 , dự báo IBC của nó có thể từ được tái tạo các mẫu trong khối 64×64 bắt đầu từ $((2m-1) \times 64, 2n \times 64)$, khối 64×64 bắt đầu từ $((2m-1) \times 64, (2n+1) \times 64)$, khối 64×64 bắt đầu từ $(2m \times 64, 2n \times 64)$ và khối 64×64 hiện tại.

a. Theo các cách, nếu khối 64×64 bắt đầu từ $(2m \times 64, (2n+1) \times 64)$ đã được tái tạo, dự báo IBC có thể từ các mẫu được tái tạo trong khối 64×64 bắt đầu từ $((2m-1) \times 64, 2n \times 64)$, khối 64×64 bắt đầu từ $(2m \times 64, 2n \times 64)$, khối 64×64 bắt đầu từ $(2m \times 64, (2n+1) \times 64)$ và khối 64×64 hiện tại.

67. Đối với CU trong khối 64×64 bắt đầu từ $(2m \times 64, (2n+1) \times 64)$, tức là, khối dưới bên trái 64×64 trong CTU 128×128 , dự báo IBC của nó có thể từ được tái tạo các mẫu trong khối 64×64 bắt đầu từ $((2m-1) \times 64, (2n+1) \times 64)$, khối 64×64 bắt đầu từ $(2m \times 64, 2n \times 64)$; khối 64×64 bắt đầu từ $((2m+1) \times 64, 2n \times 64)$ và khối 64×64 hiện tại.

a. Theo các cách, nếu khối 64×64 bắt đầu từ $((2m+1) \times 64, 2n \times 64)$ đã không được tái tạo, dự báo IBC có thể từ các mẫu được tái tạo trong khối 64×64 bắt đầu từ $((2m-1) \times 64, 2n \times 64)$, khối 64×64 bắt đầu từ $((2m-1) \times 64, (2n+1) \times 64)$, khối 64×64 bắt đầu từ $(2m \times 64, 2n \times 64)$ và khối 64×64 hiện tại.

68. Đề xuất điều chỉnh khu vực tham chiếu dựa trên các khối 64×64 nào mà có CU hiện tại.

a. Trong một ví dụ, cho CU bắt đầu từ (x,y) , khi $(y>>6)\&1 = 0$, hai hoặc gần đến hai khối 64×64 khác nhau, bắt đầu từ $((x>>6<<6)-128, y>>6<<6)$ và $((x>>6<<6)-64, y>>6<<6)$ có thể được tham chiếu bằng chế độ IBC.

b. Trong một ví dụ, cho CU bắt đầu từ (x,y) , khi $(y>>6)\&1 = 1$, một khối 64×64 trước đó, bắt đầu từ $((x>>6<<6)-64, y>>6<<6)$ có thể được tham chiếu bằng chế độ IBC.

69. Đối với khối bắt đầu từ (x,y) với vector khối (BVx, BVy) , nếu $\text{isRec}(((x+BVx)>>6<<6)+128-(((y+BVy)>>6)\&1)*64+(x\%64), ((y+BVy)>>6<<6)+(y\%64))$ là true, vector khối là không hợp lệ.

a. Trong một ví dụ, khối là khối độ sáng.

b. Trong một ví dụ, khối là khối sắc độ trong định dạng 4:4:4

c. Trong một ví dụ, khối chứa cả thành phần độ sáng và sắc độ

70. Đối với khối sắc độ ở định dạng 4:2:0 bắt đầu từ (x,y) và với vector khối (BVx, BVy) , nếu $\text{isRec}(((x+BVx)>>5<<5)+64-(((y+BVy)>>5)\&1)*32+(x\%32), ((y+BVy)>>5<<5)+(y\%32))$ là true, vector khối là không hợp lệ.

71. Xác định liệu BV là không hợp lệ hay không đối với khối của thành phần c có thể dựa trên tính khả dụng của các mẫu của thành phần X, thay cho kiểm tra duy nhất mẫu độ sáng.

a. Đối với khối của thành phần c bắt đầu từ (x,y) và với vector khối (BVx, BVy) , nếu $\text{isRec}(c, ((x+BVx)>>6<<6)+128-(((y+BVy)>>6)\&1)*64+(x\%64), ((y+BVy)>>6<<6)+(y\%64))$ là true, vector khối có thể được xử lý là không hợp lệ.

i. Trong một ví dụ, khối là khối độ sáng (chẳng hạn, c là thành phần độ sáng, hoặc thành phần G để mã hóa RGB).

ii. Trong một ví dụ, khối là khối sắc độ ở định dạng 4:4:4 (chẳng hạn, c là thành phần cb hoặc cr, hoặc thành phần B/R để mã hóa RGB).

iii. Trong một ví dụ, tính khả dụng của các mẫu cho cả hai thành phần độ sáng và sắc độ có thể được kiểm tra, chẳng hạn, khối chứa cả thành phần độ sáng và sắc độ

b. Đối với khối sắc độ trong định dạng 4:2:0 bắt đầu từ (x,y) của thành phần c và với vector khối (BVx, BVy) , nếu $\text{isRec}(c, ((x+BVx)>>5<<5)+64-$

$((y + BVy) > 5) \& 1) * 32 + (x \% 32)$, $((y + BVy) > 5 < 5) + (y \% 32)$ là true, vector khối có thể được xử lý là không hợp lệ.

c. Đối với khối sắc độ hoặc khối phụ bắt đầu từ (x, y) của thành phần c và với vector khối (BVx, BVy) , nếu $\text{isRec}(c, x + BVx + \text{Chroma_CTU_size}, y)$ cho thành phần sắc độ là true, vector khối có thể được xử lý là không hợp lệ, trong đó Chroma_CTU_size là kích thước CTU đối với thành phần sắc độ.

i. Trong một ví dụ, đối với định dạng 4:2:0, Chroma_CTU_size có thể bằng 64.

ii. Trong một ví dụ, khối sắc độ phụ có thể bằng khối 2x2 ở định dạng 4:2:0.

iii. Trong một ví dụ, khối sắc độ phụ có thể bằng khối 4x4 ở định dạng 4:4:4

iv. Trong một ví dụ, khối sắc độ phụ có thể tương ứng với kích thước CU nhỏ nhất trong thành phần độ sáng.

1. Theo các cách, khối sắc độ phụ có thể tương ứng với kích thước CU nhỏ nhất cho thành phần sắc độ.

72. Đối với tất cả các mục nêu trên, giả sử rằng bộ đệm tham chiếu chứa nhiều khối $M \times M$ ($M=64$). Tuy nhiên, có thể được mở rộng sang các trường hợp khác chẳng hạn bộ đệm tham chiếu chứa nhiều khối $N \times M$ (chẳng hạn, $N=128$, $M=64$).

73. Đối với tất cả các mục nêu trên, các giới hạn khác có thể được áp dụng việc bộ đệm tham chiếu nên nằm trong cùng viên/phiến/nhóm phiến/lát như khối hiện tại.

a. Trong một ví dụ, nếu một phần bộ đệm tham chiếu nằm ngoài viên/phiến/nhóm phiến/lát hiện tại, sử dụng IBC có thể được vô hiệu hóa. Báo hiệu về các phần tử cú pháp liên quan IBC có thể được bỏ qua.

b. Theo các cách, nếu một phần bộ đệm tham chiếu nằm ngoài viên/phiến/nhóm phiến/lát hiện tại, IBC có thể vẫn được kích hoạt cho một khối, tuy nhiên, vector khối được liên kết với một khối có thể chỉ trỏ đến bộ đệm tham chiếu còn lại.

74. Đề xuất có K1 VPDU được mã hóa gần đây nhất, nếu khả dụng, trong hàng VPDU thứ nhất của hàng CTU/CTB và K2 VPDU được mã hóa gần đây

nhất, nếu khả dụng, trong hàng VPDU thứ hai của hàng CTU/CTB dưới dạng khu vực tham chiếu cho IBC, không bao gồm VPDU hiện tại.

- a. Trong một ví dụ, K1 bằng 2 và K2 bằng 1.
- b. Trong một ví dụ, các phương pháp nêu trên có thể được áp dụng khi kích thước CTU/CTB là 128x128 và kích thước VPDU là 64x64.
- c. Trong một ví dụ, các phương pháp nêu trên có thể được áp dụng khi kích thước CTU/CTB là 64x64 và kích thước VPDU là 64x64 và/hoặc 32x32.
- d. Trong một ví dụ, các phương pháp nêu trên có thể được áp dụng khi kích thước CTU/CTB là 32x32 và kích thước VPDU là 32x32 hoặc nhỏ hơn.

75. Các phương pháp nêu trên có thể được áp dụng trong các giai đoạn khác nhau.

- a. Trong một ví dụ, phép toán môđun (chẳng hạn, mod b) của các vector khối (BV) có thể được gọi trong quá trình kiểm tra tính khả dụng của các BV để quyết định liệu BV là hợp lệ hay không.
- b. Trong một ví dụ, phép toán môđun (chẳng hạn, mod b) của các vector khối (BV) có thể được gọi để nhận diện vị trí của mẫu tham chiếu (chẳng hạn, theo các kết quả môđun của vị trí của mẫu hiện tại và BV) trong bộ đệm ảo IBC hoặc bộ đệm ảnh được tái tạo (chẳng hạn, trước quá trình lọc trong vòng).

5. Các phương án thực hiện

5.1 Phương án thực hiện #1

Triển khai của bộ đệm cho IBC được mô tả phía dưới:

Kích thước bộ đệm bằng 128x128. kích thước CTU cũng bằng 128x128. Để tạo mã của CTU thứ nhất trong hàng CTU, bộ đệm được khởi tạo với 128 (đối với tín hiệu video 8 bit). Để tạo mã CTU thứ k trong hàng CTU, bộ đệm được khởi tạo với việc tái tạo trước khi lọc vòng CTU thứ k-1.

Fig.3 là hình vẽ thể hiện ví dụ mã hóa khối bắt đầu từ (x,y).

Khi mã hóa khối bắt đầu từ (x,y) liên quan đến CTU hiện tại, vector khối (BV_x, BV_y) = (x-x₀, y-y₀) được gửi đến bộ giải mã để chỉ báo khối tham chiếu là từ (x₀,y₀) trong bộ đệm IBC. Giả sử chiều rộng và chiều cao của khối lần lượt là w

và h. Khi hoàn thành mã hóa của khối, khu vực wxh bắt đầu từ (x,y) trong bộ đệm IBC sẽ được cập nhật với việc tái tạo khối trước khi lọc vòng.

5.2 Phương án thực hiện #2

Fig.4 thể hiện các ví dụ về các cách khả thi để chọn các khối 64x64 được mã hóa trước đó.

5.3 Phương án thực hiện #3

Fig.5 là hình vẽ thể hiện ví dụ về các cách khả thi để thay đổi thứ tự mã hóa/giải mã của các khối 64x64.

5.4 Phương án thực hiện #4

Fig.8 thể hiện các cách khả thi khác để chọn các khối 64x64 được mã hóa trước đó, khi thứ tự giải mã cho khối 64x64s là từ trên xuống dưới, từ trái sang phải.

5.5 Phương án thực hiện #5

Fig.9 thể hiện các cách khả thi khác để chọn các khối 64x64 được mã hóa trước đó.

5.6 Phương án thực hiện #6

Fig.11 thể hiện các cách khả thi khác để chọn các khối 64x64 được mã hóa trước đó, khi thứ tự giải mã cho các khối 64x64 là từ trái sang phải, trên xuống dưới.

5.7 Phương án thực hiện #7

Giả sử rằng kích thước CTU bằng $W \times W$, triển khai bộ đệm IBC có kích thước $mW \times W$ và bitdepth là B , ở bộ giải mã như sau.

Lúc bắt đầu giải mã hàng CTU, khởi tạo bộ đệm có giá trị $(1 \ll (B-1))$ và thiết lập điểm bắt đầu để cập nhật (x_b, y_b) bằng $(0,0)$.

Khi CU bắt đầu từ (x, y) liên quan đến CTU góc trên bên trái và có kích thước wxh được giải mã, khu vực bắt đầu từ (x_b+x, y_b+y) và kích thước wxh sẽ được cập nhật với các giá trị pixel được tái tạo của CU, sau khi chiều sâu bit được căn chỉnh với B-bit.

Sau khi CTU được giải mã, điểm bắt đầu để cập nhật (x_b, y_b) sẽ được thiết lập bằng $((x_b + W) \bmod mW, 0)$.

Khi giải mã CU IBC với vector khối (BV_x, BV_y) , đối với pixel bất kỳ (x, y) liên quan đến CTU góc trên bên trái, dự báo được trích xuất từ bộ đệm ở vị trí $((x + BV_x) \bmod mW, (y + BV_y) \bmod W)$ sau khi căn chỉnh chiều sâu bit với chiều sâu bit của các tín hiệu dự báo.

Trong một ví dụ, B được gán bằng 7, hoặc 8 trong khi chiều sâu bit đầu ra/đầu vào của khối có thể bằng 10.

5.8 Phương án thực hiện #8

Đối với CU độ sáng hoặc CU độ sáng/sắc độ kết hợp bắt đầu từ (x, y) liên quan đến góc trên bên trái của ảnh và vector khối (BV_x, BV_y) , vector khối là không hợp lệ khi $\text{isRec}(((x + BV_x) \gg 6 \ll 6) + 128 - (((y + BV_y) \gg 6 \ll 6) \& 1) * 64 + (x \% 64), ((y + BV_y) \gg 6 \ll 6) + (y \% 64))$ là true.

Đối với CU sắc độ bắt đầu từ (x, y) liên quan đến góc trên bên trái của ảnh và vector khối (BV_x, BV_y) , vector khối là không hợp lệ khi $\text{isRec}(((x + BV_x) \gg 5 \ll 5) + 64 - (((y + BV_y) \gg 5 \ll 5) \& 1) * 32 + (x \% 32), ((y + BV_y) \gg 5 \ll 5) + (y \% 32))$ là true.

5.9 Phương án thực hiện #9

Đối với khối sắc độ hoặc khối phụ bắt đầu từ (x, y) trong định dạng 4:2:0 liên quan đến góc trên bên trái của ảnh và vector khối (BV_x, BV_y) , vector khối là không hợp lệ khi $\text{isRec}(c, (x + BV_x + 64, y + BV_y))$ là true, trong đó c là thành phần sắc độ.

Đối với khối sắc độ hoặc khối phụ bắt đầu từ (x, y) trong định dạng 4:4:4 liên quan đến góc trên bên trái của ảnh và vector khối (BV_x, BV_y) , vector khối là không hợp lệ khi $\text{isRec}(c, (x + BV_x + 128, y + BV_y))$ là true, trong đó c là thành phần sắc độ.

5.10 Phương án thực hiện #10

Đối với CU độ sáng hoặc CU độ sáng/sắc độ đồng thời bắt đầu từ (x, y) liên quan đến góc trên bên trái của ảnh và vector khối (BV_x, BV_y) , vector khối là

không hợp lệ khi $\text{isRec}(((x+Bv_x) \gg 6 \ll 6) + 128 - ((y+Bv_y) \gg 6 \ll 6) \& 1) * 64 + (x \% 64), ((y+Bv_y) \gg 6 \ll 6) + (y \% 64))$ là true.

Đối với khối sắc độ hoặc khối phụ bắt đầu từ (x,y) trong định dạng 4:2:0 liên quan đến góc trên bên trái của ảnh và vector khối (BVx, BVy), vector khối là không hợp lệ khi $\text{isRec}(c, ((x+Bv_x) \gg 5 \ll 5) + 64 - (((y+Bv_y) \gg 5 \ll 5) \& 1) * 32 + (x \% 32), ((y+Bv_y) \gg 5 \ll 5) + (y \% 32))$ là true, trong đó c là thành phần sắc độ.

5.11 Phương án thực hiện #11

Phương án thực hiện nêu bật cách triển khai duy trì hai VPDU được mã hóa gần nhất trong hàng VPDU thứ nhất và một VPDU được mã hóa gần nhất trong hàng VPDU thứ hai của hàng CTU/CTB, không bao gồm VPDU hiện tại.

Khi thứ tự mã hóa VPDU từ trên xuống dưới và trái sang phải, khu vực tham chiếu được minh họa như trên Fig.13.

Khi thứ tự mã hóa VPDU từ trái sang phải và trên xuống dưới và VPDU hiện tại không phải sang bên phải biên ảnh, khu vực tham chiếu được minh họa như trên Fig.14.

Khi thứ tự mã hóa VPDU từ trái sang phải và trên xuống dưới và VPDU hiện tại sang bên phải biên ảnh, khu vực tham chiếu có thể được minh họa dưới dạng Fig.15.

Giả sử khối độ sáng (x, y) có kích thước wxh, vector khối (BVx, BVy) là hợp lệ hay không có thể được thông báo bằng cách kiểm tra điều kiện sau:

$\text{isRec}(((x+Bv_x+128) \gg 6 \ll 6) - (\text{ref}_y \& 0x40) + (x \% 64), ((y+Bv_y) \gg 6 \ll 6) + (\text{ref}_y \gg 6 == y \gg 6) ? (y \% 64) : 0)$, trong đó $\text{ref}_y = (y \& 0x40) ? (y+Bv_y) : (y+Bv_y+w-1)$.

Nếu hàm trên trả về true, vector khối (BVx, BVy) là không hợp lệ, ngược lại vector khối có thể là hợp lệ.

5.12 Phương án thực hiện #12

Nếu kích thước CTU bằng 192x128, bộ đệm ảo có kích thước 192x128 được duy trì để theo dõi các mẫu tham chiếu cho IBC.

Mẫu (x, y) so với góc trên bên trái của ảnh được liên kết với vị trí $(x\%192, y\%128)$ so với góc trên bên trái của bộ đệm. Các bước sau thể hiện cách đánh dấu tính khả dụng của các mẫu liên kết với bộ đệm ảo cho tham chiếu IBC.

Vị trí $(x_{\text{PrevVPDU}}, y_{\text{PrevVPDU}})$ so với góc trên bên trái của ảnh được ghi nhận để tiến về mẫu trên bên trái của VPDU được giải mã gần đây nhất.

1) Lúc bắt đầu giải mã hàng VPDU, tất cả các vị trí của bộ đệm được đánh dấu là không khả dụng. $(x_{\text{PrevVPDU}}, y_{\text{PrevVPDU}})$ được thiết lập như là (0,0).

2) Lúc bắt đầu giải mã CU thứ nhất của VPDU, các vị trí (x, y) với $x = (x_{\text{PrevVPDU}} - 2W_{\text{VPDU}} + 2mW_{\text{VPDU}})\%(mW_{\text{VPDU}}), \dots, ((x_{\text{PrevVPDU}} - 2W_{\text{VPDU}} + 2mW_{\text{VPDU}})\%(mW_{\text{VPDU}})) - 1 + W_{\text{VPDU}};$ và $y = y_{\text{PrevVPDU}}\%(nH_{\text{VPDU}}), \dots, (y_{\text{PrevVPDU}}\%(nH_{\text{VPDU}})) - 1 + H_{\text{VPDU}}$ có thể được đánh dấu là không khả dụng. Sau đó, $(x_{\text{PrevVPDU}}, y_{\text{PrevVPDU}})$ được thiết lập dưới dạng $(x_{\text{CU}}, y_{\text{CU}})$, tức là, vị trí trên bên trái của CU so với ảnh.

3) Sau khi giải mã CU, các vị trí (x, y) với $x = x_{\text{CU}}\%(mW_{\text{VPDU}}), \dots, (x_{\text{CU}} + \text{CU_width} - 1)\%(mW_{\text{VPDU}})$ và $y = y_{\text{CU}}\%(nH_{\text{VPDU}}), \dots, (y_{\text{CU}} + \text{CU_height} - 1)\%(nH_{\text{VPDU}})$ được đánh dấu là khả dụng.

4) Đối với CU IBC có vector khối $(x_{\text{BV}}, y_{\text{BV}})$, nếu vị trí (x, y) bất kỳ có $x = (x_{\text{CU}} + x_{\text{BV}})\%(mW_{\text{VPDU}}), \dots, (x_{\text{CU}} + x_{\text{BV}} + \text{CU_width} - 1)\%(mW_{\text{VPDU}})$ và $y = (y_{\text{CU}} + y_{\text{BV}})\%(nH_{\text{VPDU}}), \dots, (y_{\text{CU}} + y_{\text{BV}} + \text{CU_height} - 1)\%(nH_{\text{VPDU}})$ được đánh dấu là không khả dụng, vector khối được xem xét là không hợp lệ.

Fig.16 thể hiện trạng thái bộ đệm cùng với trạng thái giải mã VPDU trong ảnh.

5.13 Phương án thực hiện #13

Nếu kích thước CTU bằng 128x128 hoặc kích thước CTU bằng lớn hơn kích thước VPDU (chẳng hạn, 64x64 theo thiết kế hiện tại) hoặc kích thước CTU bằng lớn hơn kích thước VPDU (chẳng hạn, 64x64 theo thiết kế hiện tại), bộ đệm ảo có kích thước 192x128 được duy trì để theo dõi các mẫu tham chiếu cho IBC. Dưới đây, khi $a < 0$, $(a \% b)$ được định nghĩa là $\text{floor}(a/b)*b$, trong đó $\text{floor}(c)$ trả về số nguyên lớn nhất không lớn hơn c.

Mẫu (x, y) so với góc trên bên trái của ảnh được liên kết với vị trí $(x\%192, y\%128)$ so với góc trên bên trái của bộ đệm. Các bước sau thể hiện cách thức đánh dấu tính khả dụng của các mẫu liên kết với bộ đệm ảo cho tham chiếu IBC.

Vị trí $(x_{\text{PrevVPDU}}, y_{\text{PrevVPDU}})$ so với góc trên bên trái của ảnh được ghi nhận đại diện mẫu trên bên trái của VPDU được giải mã gần đây nhất.

1) Lúc bắt đầu giải mã hàng VPDU, tất cả các vị trí của bộ đệm được đánh dấu là không khả dụng. $(x_{\text{PrevVPDU}}, y_{\text{PrevVPDU}})$ được thiết lập là $(0,0)$.

2) Lúc bắt đầu giải mã CU thứ nhất của VPDU,

a. Nếu $y_{\text{PrevVPDU}}\%64$ bằng 0, các vị trí (x, y) với $x = (x_{\text{PrevVPDU}} - 128)\%192, \dots, ((x_{\text{PrevVPDU}} - 128)\%192) + 63$; và $y = y_{\text{PrevVPDU}}\%128, \dots, (y_{\text{PrevVPDU}}\%128)+63$, được đánh dấu là không khả dụng. Sau đó, $(x_{\text{PrevVPDU}}, y_{\text{PrevVPDU}})$ được thiết lập là $(x_{\text{CU}}, y_{\text{CU}})$, tức là, vị trí trên bên trái của CU so với ảnh.

b. Ngược lại, các vị trí (x, y) với $x = (x_{\text{PrevVPDU}} - 64)\%192, \dots, ((x_{\text{PrevVPDU}} - 64)\%192) + 63$; và $y = y_{\text{PrevVPDU}}\%128, \dots, (y_{\text{PrevVPDU}}\%128)+63$, được đánh dấu là không khả dụng. Sau đó, $(x_{\text{PrevVPDU}}, y_{\text{PrevVPDU}})$ được thiết lập là $(x_{\text{CU}}, y_{\text{CU}})$, tức là, vị trí trên bên trái của CU so với ảnh.

3) Sau khi giải mã CU, các vị trí (x, y) với $x = x_{\text{CU}}\%192, \dots, (x_{\text{CU}}+x_{\text{CU_width}}-1)\%192$ và $y = y_{\text{CU}}\%128, \dots, (y_{\text{CU}}+y_{\text{CU_height}}-1)\%128$ được đánh dấu là khả dụng.

4) Đối với CU IBC có vector khối $(x_{\text{BV}}, y_{\text{BV}})$, nếu vị trí (x, y) bất kỳ có $x = (x_{\text{CU}}+x_{\text{BV}})\%192, \dots, (x_{\text{CU}}+x_{\text{BV}}+x_{\text{CU_width}}-1)\%192$ và $y = (y_{\text{CU}}+y_{\text{BV}})\%128, \dots, (y_{\text{CU}}+y_{\text{BV}}+y_{\text{CU_height}}-1)\%128$ được đánh dấu là không khả dụng, vector khối được xem xét là không hợp lệ.

Nếu kích thước CTU bằng $S \times S$, S không bằng 128, giả sử W_{buf} bằng $128 \times 128 / S$. Bộ đệm ảo có kích thước $W_{\text{buf}} \times S$ được duy trì để theo dõi các mẫu tham chiếu cho IBC. Kích thước VPDU bằng kích thước CTU trong trường hợp này.

Vị trí $(x_{\text{PrevVPDU}}, y_{\text{PrevVPDU}})$ so với góc trên bên trái của ảnh được ghi nhận đại diện mẫu trên bên trái của VPDU được giải mã gần đây nhất.

1) Lúc bắt đầu giải mã hàng VPDU, tất cả các vị trí của bộ đệm được đánh dấu là không khả dụng. $(x_{PrevVPDU}, y_{PrevVPDU})$ được thiết lập là $(0,0)$.

2) Lúc bắt đầu giải mã CU thứ nhất của VPDU, các vị trí (x, y) có $x = (x_{PrevVPDU} - W_{buf} * S) \% S, \dots, ((x_{PrevVPDU} - W_{buf} * S) \% S) + S - 1$; và $y = y_{PrevVPDU} \% S, \dots, (y_{PrevVPDU} \% S) + S - 1$; được đánh dấu là không khả dụng. Sau đó, $(x_{PrevVPDU}, y_{PrevVPDU})$ được thiết lập là (x_{CU}, y_{CU}) , tức là, vị trí trên bên trái của CU so với ảnh.

3) Sau khi giải mã CU, các vị trí (x, y) với $x = x_{CU} \% (W_{buf}), \dots, (x_{CU} + CU_width - 1) \% (W_{buf})$ và $y = y_{CU} \% S, \dots, (y_{CU} + CU_height - 1) \% S$ được đánh dấu là khả dụng.

4) Đối với CU IBC có vector khối (x_{BV}, y_{BV}) , nếu vị trí (x, y) bất kỳ có $x = (x_{CU} + x_{BV}) \% (W_{buf}), \dots, (x_{CU} + x_{BV} + CU_width - 1) \% (W_{buf})$ và $y = (y_{CU} + y_{BV}) \% S, \dots, (y_{CU} + y_{BV} + CU_height - 1) \% S$ được đánh dấu là không khả dụng, vector khối được xem xét là không hợp lệ.

5.14 Phương án thực hiện #14

Nếu kích thước CTU bằng 128×128 hoặc kích thước CTU bằng lớn hơn kích thước VPDU (chẳng hạn, 64×64 theo thiết kế hiện tại) hoặc kích thước CTU bằng lớn hơn kích thước VPDU (chẳng hạn, 64×64 theo thiết kế hiện tại), bộ đệm ảo có kích thước 256×128 được duy trì để theo dõi các mẫu tham chiếu cho IBC. Dưới đây, khi $a < 0$, $(a \% b)$ được định nghĩa là $\text{floor}(a/b) * b$, trong đó $\text{floor}(c)$ trả về số nguyên lớn nhất không lớn hơn c .

Mẫu (x, y) so với góc trên bên trái của ảnh được liên kết với vị trí $(x \% 256, y \% 128)$ so với góc trên bên trái của bộ đệm. Các bước sau thể hiện cách thức đánh dấu tính khả dụng của các mẫu liên kết với bộ đệm ảo để tham chiếu IBC.

Vị trí $(x_{PrevVPDU}, y_{PrevVPDU})$ so với góc trên bên trái của ảnh được ghi nhận để đại diện mẫu trên bên trái của VPDU được giải mã gần đây nhất.

1) Lúc bắt đầu giải mã hàng VPDU, tất cả các vị trí của bộ đệm được đánh dấu là không khả dụng. $(x_{PrevVPDU}, y_{PrevVPDU})$ được thiết lập là $(0,0)$.

2) Lúc bắt đầu giải mã CU thứ nhất của VPDU,

a. Nếu $y_{PrevVPDU} \% 64$ bằng 0, các vị trí (x, y) với $x = (x_{PrevVPDU} - 128) \% 256, \dots, ((x_{PrevVPDU} - 128) \% 256) + 63$; và $y = y_{PrevVPDU} \% 128, \dots, (y_{PrevVPDU} \% 128) + 63$, được đánh dấu là không khả dụng. Sau đó, $(x_{PrevVPDU}, y_{PrevVPDU})$ được thiết lập là (x_{CU}, y_{CU}) , tức là, vị trí trên bên trái của CU so với ảnh.

b. Ngược lại, các vị trí (x, y) với $x = (x_{PrevVPDU} - 64) \% 256, \dots, ((x_{PrevVPDU} - 64) \% 256) + 63$; và $y = y_{PrevVPDU} \% 128, \dots, (y_{PrevVPDU} \% 128) + 63$, được đánh dấu là không khả dụng. Sau đó, $(x_{PrevVPDU}, y_{PrevVPDU})$ được thiết lập là (x_{CU}, y_{CU}) , tức là, vị trí trên bên trái của CU so với ảnh.

3) Sau khi giải mã CU, các vị trí (x, y) với $x = x_{CU} \% 256, \dots, (x_{CU} + CU_width - 1) \% 256$ và $y = y_{CU} \% 128, \dots, (y_{CU} + CU_height - 1) \% 128$ được đánh dấu là khả dụng.

4) Đối với CU IBC với vector khối (x_{BV}, y_{BV}) , nếu vị trí (x, y) bất kỳ có $x = (x_{CU} + x_{BV}) \% 256, \dots, (x_{CU} + x_{BV} + CU_width - 1) \% 256$ và $y = (y_{CU} + y_{BV}) \% 128, \dots, (y_{CU} + y_{BV} + CU_height - 1) \% 128$ được đánh dấu là không khả dụng, vector khối được xem xét là không hợp lệ.

Khi kích thước CTU không bằng 128×128 hoặc nhỏ hơn 64×64 hoặc nhỏ hơn 64×64 , quá trình tương tự áp dụng như theo phương án thực hiện trước đó, tức là, phương án thực hiện #14.

5.15 Phương án thực hiện #15

Quá trình đánh dấu tính khả dụng tham chiếu IBC được mô tả như sau. Thay đổi được chỉ báo trong văn bản in đậm, gạch chân, in nghiêng theo sáng chế.

7.3.7.1 Cú pháp dữ liệu lát chung

slice data() {	Mô tả
for(i = 0; i < NumBricksInCurrSlice; i++) {	
CtbAddrInBs = FirstCtbAddrBs[SliceBrickIdx[i]]	
for(j = 0; j < NumCtusInBrick[SliceBrickIdx[i]]; j++, CtbAddrInBs++) {	
if((j % BrickWidth[SliceBrickIdx[i]]) == 0) {	
NumHmvpCand = 0	
NumHmvpIbcCand = 0	
$x_{PrevVPDU} = 0$	
$y_{PrevVPDU} = 0$	
if(CtbSizeY == 128)	

<u>reset ibc isDecoded(0, 0, 256, CtbSizeY, BufWidth, BufHeight)</u>	
<u>Else</u>	
<u>reset ibc isDecoded(0, 0, 128*128/CtbSizeY, CtbSizeY, BufWidth, BufHeight)</u>	
<u>}</u>	
<u>CtbAddrInRs = CtbAddrBsToRs[CtbAddrInBs]</u>	
<u>.....</u>	

<u>reset ibc isDecoded(x0, y0, w, h, BufWidth, BufHeight) {</u>	<u>Mô tả</u>
<u>if(x0 >= 0)</u>	
<u>for (x = x0 % BufWidth; x < x0 + w; x+=4)</u>	
<u>for (y = y0 % BufHeight; y < y0 + h; y+=4)</u>	
<u>isDecoded[x >> 2][y >> 2] = 0</u>	
<u>}</u>	

BufWidth bằng (CtbSizeY==128)?256:(128*128/CtbSizeY) và BufHeight bằng CtbSizeY.

7.3.7.5 Cú pháp CU

<u>coding_unit(x0, y0, cbWidth, cbHeight, treeType) {</u>	<u>Mô tả</u>
<u>if(treeType != DUAL_TREE_CHROMA && (CtbSizeY == 128) && (x0 % 64) == 0 && (y0 % 64) == 0) {</u>	
<u>for(x = x0; x < x0 + cbWidth; x += 64)</u>	
<u>for(y = y0; y < y0 + cbHeight; y += 64)</u>	
<u>if((yPrevVPDU % 64) == 0)</u>	
<u>reset ibc isDecoded(xPrevVPDU - 128, yPrevVPDU, 64, 64, BufWidth, BufHeight)</u>	
<u>Else</u>	
<u>thiết lập lại ibc isDecoded(xPrevVPDU - 64, yPrevVPDU, 64, 64, BufWidth, BufHeight)</u>	
<u>xPrevVPDU = x0</u>	
<u>yPrevVPDU = y0</u>	
<u>}</u>	
<u>if(treeType != DUAL_TREE_CHROMA && (CtbSizeY < 128) && (x0 % CtbSizeY) == 0 && (y0 % CtbSizeY) == 0) {</u>	
<u>reset ibc isDecoded(xPrevVPDU - (128*128/CtbSizeY - CtbSizeY), yPrevVPDU, 64, 64, BufWidth, BufHeight)</u>	
<u>xPrevVPDU = x0</u>	
<u>yPrevVPDU = y0</u>	
<u>}</u>	
<u>if(slice_type != I sps_ibc_enabled_flag) {</u>	
<u>if(treeType != DUAL_TREE_CHROMA && !(cbWidth == 4 && cbHeight == 4 && !sps_ibc_enabled_flag))</u>	
<u>cu_skip_flag[x0][y0]</u>	<u>ae(v)</u>
<u>if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4))</u>	
<u>pred_mode_flag</u>	<u>ae(v)</u>

if(((slice_type == I && cu_skip_flag[x0][y0] == 0) (slice_type != I && (CuPredMode[x0][y0] != MODE_INTRA (cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] == 0)))) && sps_ibc_enabled_flag && (cbWidth != 128 && cbHeight != 128))	
pred_mode_ibc_flag	ae(v)
}	
...	

8.6.2 Quá trình dẫn xuất đối với các thành phần vec tơ chuyển động cho các khối IBC

8.6.2.1 Nói chung

...

Yêu cầu tương thích dòng biết chính là khi quá trình kiểm tra tính hợp lệ vector khối theo mệnh đề 8.6.3,2 được gọi với vector khối mvL, isBVvalid sẽ là true.

...

8.6.3 Quá trình giải mã cho các khối ibc

8.6.3.1 Nói chung

Quá trình này được gọi khi giải mã CU được tạo mã trong chế độ dự báo ibc.

Các đầu vào cho quá trình này là:

- vị trí độ sáng (xCb, yCb) xác định mẫu trên bên trái của CB hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến cbWidth xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,
- biến cbHeight xác định chiều cao của CB hiện tại trong các mẫu độ sáng,
- các biến numSbX và numSbY xác định số lượng khối phụ mã hóa độ sáng theo hướng ngang và dọc,
- các vector chuyển động mv[xSbIdx][ySbIdx] có xSbIdx = 0 .. numSbX – 1, và ySbIdx = 0 .. numSbY – 1,
- biến cIdx xác định chỉ số thành phần màu của khối hiện tại.
- ***a(nIbcBufW)x(ctbSize) array ibcBuf***

...

Đối với mỗi khối phụ mã hóa ở chỉ số khối phụ (xSbIdx, ySbIdx) có xSbIdx = 0 .. numSbX – 1, và ySbIdx = 0 .. numSbY – 1, phần sau áp dụng:

– Vị trí độ sáng (x_{Sb} , y_{Sb}) xác định mẫu trên bên trái của khối phụ mã hóa hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại được suy ra như sau:

$$(x_{Sb}, y_{Sb}) = (x_{Cb} + x_{SbIdx} * sbWidth, y_{Cb} + y_{SbIdx} * sbHeight) \quad (8-913)$$

Nếu $cIdx$ bằng 0, $nIbcBufW$ được gán bằng $ibcBufferWidth$, ngược lại $nIbcBufW$ được gán bằng $(ibcBufferWidth / SubWidthC)$. Phần sau áp dụng:

$predSamples[x_{Sb} + x][y_{Sb} + y] = ibcBuf[(x_{Sb} + x + (mv[x_{Sb}][y_{Sb}][0] >> 4)) \% nIbcRefW][y_{Sb} + y + (mv[x_{Sb}][y_{Sb}][1] >> 4)]$

với $x = 0..sbWidth - 1$ and $y = 0..sbHeight - 1$.

...

8.6.3.2 Quá trình kiểm tra tính hợp lệ vector khối

Các đầu vào cho quá trình này là:

- vị trí độ sáng (x_{Cb} , y_{Cb}) xác định mẫu trên bên trái của CB hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
 - biến $cbWidth$ xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,
 - biến $cbHeight$ xác định chiều cao của CB hiện tại trong các mẫu độ sáng,
 - các biến $numSbX$ and $numSbY$ xác định số lượng khối phụ mã hóa độ sáng theo hướng ngang và dọc,
 - các vector khối $mv[x_{SbIdx}][y_{SbIdx}]$ với $x_{SbIdx} = 0 .. numSbX - 1$, và $y_{SbIdx} = 0 .. numSbY - 1$,
 - biến $cIdx$ xác định chỉ số thành phần màu của khối hiện tại.
 - $a(nIbcBufW)x(ctbSize)$ array $ibcBuf$
- Các đầu ra của quá trình này là cờ $isBVvalid$ để chỉ báo liệu vector khối là hợp lệ hay không.

Phần sau áp dụng

1. $isBVvalid$ được gán bằng true.
2. Nếu $((y_{Cb} \& (ctbSize - 1)) + mv[0][0][1] + cbHeight) > ctbSize$, $isBVvalid$ được gán bằng false.

3. Ngược lại, đối với mỗi chỉ số khối phụ $xSbIdx$, $ySbIdx$ với $xSbIdx = 0 .. numSbX - 1$, và $ySbIdx = 0 .. numSbY - 1$, vị trí của nó so với mẫu độ sáng trên bên trái của $ibcBuf$ được suy ra:

$$xTL = (xCb + xSbIdx * sbWidth + mv[xSbIdx][ySbIdx][0]) \& (nIbcBufW - 1)$$

$$yTL = (yCb \& (ctbSize - 1)) + ySbIdx * sbHeight + mv[xSbIdx][ySbIdx][1]$$

$$xBR = (xCb + xSbIdx * sbWidth + sbWidth - 1 + mv[xSbIdx][ySbIdx][0]) \& (nIbcBufW - 1)$$

$$yBR = (yCb \& (ctbSize - 1)) + ySbIdx * sbHeight + sbHeight - 1 + mv[xSbIdx][ySbIdx][1]$$

nếu $(isDecoded[xTL >> 2][yTL >> 2] == 0)$ hoặc $(isDecoded[xBR >> 2][yTL >> 2] == 0)$ hoặc $(isDecoded[xBR >> 2][yBR >> 2] == 0)$, $isBVvalid$ được gán bằng *false*.

8.7.5 Quá trình tái tạo ảnh

8.7.5.1 Tổng quát

Các đầu vào cho quá trình này là:

- Vị trí $(xCurr, yCurr)$ xác định mẫu trên bên trái của khối hiện tại so với mẫu trên bên trái của thành phần ảnh hiện tại,
- các biến $nCurrSw$ và $nCurrSh$ lần lượt xác định chiều rộng và chiều cao của khối hiện tại,
- biến $cIdx$ xác định thành phần màu của khối hiện tại,
- mảng $(nCurrSw) \times (nCurrSh)$ $predSamples$ xác định các mẫu được dự báo của khối hiện tại,
- mảng $(nCurrSw) \times (nCurrSh)$ $resSamples$ xác định các mẫu dư của khối hiện tại.

Đầu ra của quá trình này là

- mảng mẫu ảnh được tái tạo $recSamples$.
- an tham chiếu IBC array $ibcBuf$.

...

Ký hiệu $nIbcBufW$ là chiều rộng of $ibcBuf$, phần sau áp dụng:

$$\frac{ibcBuf[(xCurr + i) \& (nIbcBufW - 1)][(yCurr + j) \& (ctbSize - 1)]}{recSamples[xCurr + i][yCurr + j]}$$

$$\text{với } i = 0..nCurrSw - 1, j = 0..nCurrSh - 1.$$

5.16 Phương án thực hiện #16

Điều này giống hệt phương án thực hiện trước đó ngoại trừ các thay đổi sau

	Mô tả
slice_data() {	
for(i = 0; i < NumBricksInCurrSlice; i++) {	
CtbAddrInBs = FirstCtbAddrBs[SliceBrickIdx[i]]	
for(j = 0; j < NumCtusInBrick[SliceBrickIdx[i]]; j++, CtbAddrInBs++) {	
if((j % BrickWidth[SliceBrickIdx[i]]) == 0) {	
NumHmvpCand = 0	
NumHmvpIbcCand = 0	
<u>xPrevVPDU = 0</u>	
<u>yPrevVPDU = 0</u>	
<u>if(CtbSizeY == 128)</u>	
<u>thiết lập lại ibc isDecoded(0, 0, 192, CtbSizeY, BufWidth, BufHeight)</u>	
Else	
<u>thiết lập lại ibc isDecoded(0, 0, 128*128/CtbSizeY, CtbSizeY,</u>	
<u>BufWidth, BufHeight)</u>	
}	
CtbAddrInRs = CtbAddrBsToRs[CtbAddrInBs]	
.....	

<u>thiết lập lại ibc isDecoded(x0, y0, w, h, BufWidth, BufHeight) {</u>	Mô tả
<u>if(x0 >= 0)</u>	
<u>for (x = x0 % BufWidth; x < x0 + w; x+=4)</u>	
<u>for (y = y0 % BufHeight; y < y0 + h; y+=4)</u>	
<u>isDecoded[x >> 2][y >> 2] = 0</u>	
<u>}</u>	

BufWidth bằng (CtbSizeY==128)?192:(128*128/CtbSizeY) and BufHeight bằng CtbSizeY.

5.17 Phương án thực hiện #17

Thay đổi trong một số ví dụ được chỉ báo trong văn bản in đậm, gạch chân theo sáng chế.

7.3.7 Cú pháp dữ liệu lát

7.3.7.1 Cú pháp dữ liệu lát tổng quát

	Mô tả
slice_data() {	
for(i = 0; i < NumBricksInCurrSlice; i++) {	
CtbAddrInBs = FirstCtbAddrBs[SliceBrickIdx[i]]	
for(j = 0; j < NumCtusInBrick[SliceBrickIdx[i]]; j++, CtbAddrInBs++) {	
if((j % BrickWidth[SliceBrickIdx[i]]) == 0) {	
NumHmvpCand = 0	
NumHmvpIbcCand = 0	
<u>resetIbcBuf = 1</u>	
}	
CtbAddrInRs = CtbAddrBsToRs[CtbAddrInBs]	
coding_tree_unit()	
if(entropy_coding_sync_enabled_flag && ((j + 1) % BrickWidth[SliceBrickIdx[i]] == 0)) {	
end of subset one bit /* bằng 1 */	ae(v)
if(j < NumCtusInBrick[SliceBrickIdx[i]] - 1)	
byte_alignment()	
}	
}	
if(!entropy_coding_sync_enabled_flag) {	
end of brick one bit /* bằng 1 */	ae(v)
if(i < NumBricksInCurrSlice - 1)	
byte_alignment()	
}	
}	
}	

7.4.8.5 Các ngữ nghĩa CU

Khi tất cả các điều kiện sau là true, danh sách vec tơ chuyển động dựa trên lịch sử (HMVP) cho vùng danh sách ứng viên hợp nhất được chia sẻ được cập nhật bằng cách thiết lập NumHmvpSmrIbcCand bằng NumHmvpIbcCand, và thiết lập HmvpSmrIbcCandList[i] bằng HmvpIbcCandList[i] cho $i = 0..NumHmvpIbcCand - 1$:

- IsInSmr[x0][y0] bằng TRUE.
- SmrX[x0][y0] bằng x0.
- SmrY[x0][y0] bằng y0.

Các phép gán sau được thực hiện cho $x = x0..x0 + cbWidth - 1$ và $y = y0..y0 + cbHeight - 1$:

$$CbPosX[x][y] = x0 \quad (7-135)$$

$$CbPosY[x][y] = y0 \quad (7-136)$$

$$CbWidth[x][y] = cbWidth \quad (7-137)$$

$$CbHeight[x][y] = cbHeight \quad (7-138)$$

Thiết lập $vSize$ làm $\min(ctbSize, 64)$ và $wIbcBuf$ làm $(128*128/ctbSize)$.

Chiều rộng và chiều cao của $ibcBuf$ là $wIbcBuf$ và $ctbSize$ theo đó.

Nếu $refreshIbcBuf$ bằng 1, phần sau áp dụng

– $ibcBuf[x \% wIbcBuf][y \% ctbSize] = -1$, với $x = x0..x0 + wIbcBuf - 1$

và $y = y0..y0 + ctbSize - 1$

– $resetIbcBuf = 0$

Khi $(x0 \% vSize)$ bằng 0 và $(y0 \% vSize)$ bằng 0, với $x = x0..x0 + vSize - 1$

và $y = y0..y0 + vSize - 1$, phần sau áp dụng

$ibcBuf[x \% wIbcBuf][y \% ctbSize] = -1$

8.6.2 Quá trình dẫn xuất cho các thành phần vec tơ chuyển động cho các khối IBC

8.6.2.1 Tổng quát

Các đầu vào cho quá trình này là:

- vị trí độ sáng (xCb, yCb) của mẫu trên bên trái của khối mã độ sáng hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến $cbWidth$ xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,
- biến $cbHeight$ xác định chiều cao của CB hiện tại trong các mẫu độ sáng.

Các đầu ra của quá trình này là:

- vector chuyển động độ sáng có độ chính xác mẫu phân số 1/16 mvL.

Vector chuyển động độ sáng mvL được suy ra như sau:

- Quá trình dẫn xuất cho dự báo vector chuyển động độ sáng IBC như được xác định trong mệnh đề 8.6.2.2 được gọi với vị trí độ sáng (xCb, yCb), các đầu vào biến $cbWidth$ và $cbHeight$, và đầu ra là vector chuyển động độ sáng mvL.

- Khi $general_merge_flag[xCb][yCb]$ bằng 0, phần sau áp dụng:

1. Biến mvd được suy ra như sau:

$$mvd[0] = MvdL0[xCb][yCb][0] \quad (8-883)$$

$$mvd[1] = MvdL0[xCb][yCb][1] \quad (8-884)$$

2. Quá trình làm tròn cho các vector chuyển động như được xác định trong mệnh đề 8.5.2,14 được gọi với mvX được thiết lập bằng mvL , $rightShift$ được thiết lập bằng $MvShift + 2$, và $leftShift$ được thiết lập bằng $MvShift + 2$ làm đầu vào MV được làm trònL làm đầu ra.

3. Vector chuyển động độ sáng mvL được chỉnh sửa như sau:

$$u[0] = (mvL[0] + mvd[0] + 2^{18}) \% 2^{18} \quad (8-885)$$

$$mvL[0] = (u[0] \geq 2^{17}) ? (u[0] - 2^{18}) : u[0] \quad (8-886)$$

$$u[1] = (mvL[1] + mvd[1] + 2^{18}) \% 2^{18} \quad (8-887)$$

$$mvL[1] = (u[1] \geq 2^{17}) ? (u[1] - 2^{18}) : u[1] \quad (8-888)$$

Lưu ý 1– Các giá trị thu được của mvL[0] và mvL[1] như được xác định phía trên sẽ luôn nằm trọn trong khoảng từ -2^{17} đến $2^{17} - 1$.

Quá trình cập nhật cho danh sách HMVP như được xác định trong mệnh đề 8.6.2,6 được gọi với vector chuyển động độ sáng mvL.

Tương thích dòng bit yêu cầu rằng vector khối độ sáng mvL sẽ tuân thủ theo các ràng buộc sau:

- $((yCb + (mvL[1] \gg 4)) \% wIbcBuf) + cbHeight$ nhỏ hơn hoặc bằng $ctbSize$
- Với $x = xCb..xCb + cbWidth - 1$ and $y = yCb..yCb + cbHeight - 1$, $ibcBuf[(x + (mvL[0] \gg 4)) \% wIbcBuf][(y + (mvL[1] \gg 4)) \% ctbSize]$ sẽ không bằng -1 .

8.7.5 Quá trình tái tạo ảnh

8.7.5.1 Tổng quát

Các đầu vào cho quá trình này là:

- vị trí (xCurr, yCurr) xác định mẫu trên bên trái của khối hiện tại so với mẫu trên bên trái của thành phần ảnh hiện tại,
- các biến nCurrSw và nCurrSh lần lượt xác định chiều rộng và chiều cao của khối hiện tại,
- biến cIdx xác định thành phần màu của khối hiện tại,
- mảng (nCurrSw) x (nCurrSh) predSamples xác định các mẫu được dự báo của khối hiện tại,
- mảng (nCurrSw) x (nCurrSh) resSamples xác định các mẫu dư của khối hiện tại.

Đầu ra của quá trình này là mảng mẫu ảnh được tái tạo resSamples và mảng bộ đệm IBC ibcBuf.

Tùy thuộc giá trị của thành phần màu $cIdx$, các phép gán sau được thực hiện:

- Nếu $cIdx$ bằng 0, $recSamples$ tương ứng với mảng mẫu ảnh được tái tạo S_L và hàm $clipCidx1$ tương ứng với $Clip1_Y$.
- Ngược lại, nếu $cIdx$ bằng 1, $tuCbfChroma$ được gán bằng $tu_cbf_cb[xCurr][yCurr]$, $recSamples$ tương ứng với mảng mẫu sắc độ được tái tạo S_{Cb} và hàm $clipCidx1$ tương ứng với $Clip1_C$.
- Ngược lại ($cIdx$ bằng 2), $tuCbfChroma$ được gán bằng $tu_cbf_cr[xCurr][yCurr]$, $recSamples$ tương ứng với mảng mẫu sắc độ được tái tạo S_{Cr} và hàm $clipCidx1$ tương ứng với $Clip1_C$.

Tùy thuộc giá trị của $slice_lmcs_enabled_flag$, phần sau áp dụng:

- Nếu $slice_lmcs_enabled_flag$ bằng 0, khối $(nCurrSw) \times (nCurrSh)$ của các mẫu được tái tạo $recSamples$ ở vị trí $(xCurr, yCurr)$ được suy ra như sau với $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$:

$$recSamples[xCurr + i][yCurr + j] = clipCidx1(predSamples[i][j] + resSamples[i][j]) \quad (8-992)$$

- Ngược lại ($slice_lmcs_enabled_flag$ bằng 1), phần sau áp dụng:
- Nếu $cIdx$ bằng 0, phần sau áp dụng:
- Tái tạo ảnh bằng quá trình ánh xạ cho các mẫu độ sáng như được xác định trong mệnh đề 8.7.5.2 được gọi với vị trí độ sáng $(xCurr, yCurr)$, chiều rộng khối $nCurrSw$ và chiều cao $nCurrSh$, mảng mẫu độ sáng được dự báo $predSamples$, và mảng mẫu độ sáng dư $resSamples$ làm đầu vào, và đầu ra là mảng mẫu độ sáng được tái tạo $recSamples$.
- Ngược lại ($cIdx$ lớn hơn 0), tái tạo ảnh có quá trình chia tỷ lệ phần dư sắc độ phụ thuộc độ sáng cho các mẫu sắc độ như được xác định trong mệnh đề 8.7.5.3 được gọi với vị trí sắc độ $(xCurr, yCurr)$, chiều rộng khối biến đổi $nCurrSw$ và chiều cao $nCurrSh$, cờ khối được mã hóa của khối biến đổi sắc độ hiện tại $tuCbfChroma$, mảng mẫu sắc độ được dự báo $predSamples$, và mảng mẫu sắc độ dư $resSamples$ làm đầu vào, và đầu ra là mảng mẫu sắc độ được tái tạo $recSamples$.

Sau khi giải mã CU hiện tại, phần sau áp dụng:

$$\frac{ibcBuf[(xCurr + i) \% wIbcBuf][(yCurr + j) \% ctbSize]}{recSamples[xCurr + i][yCurr + j]} =$$

với $i = 0..nCurrSw - 1$, $j = 0..nCurrSh - 1$.

5.18 Phương án thực hiện #18

Các thay đổi trong một số ví dụ được chỉ báo trong văn bản in đậm, gạch chân, in nghiêng theo sáng chế.

7.3.7 Cú pháp dữ liệu lát

7.3.7.1 Cú pháp dữ liệu lát tổng quát

slice data() {	Mô tả
for(i = 0; i < NumBricksInCurrSlice; i++) {	
CtbAddrInBs = FirstCtbAddrBs[SliceBrickIdx[i]]	
for(j = 0; j < NumCtusInBrick[SliceBrickIdx[i]]; j++, CtbAddrInBs++) {	
if((j % BrickWidth[SliceBrickIdx[i]]) == 0) {	
NumHmvpCand = 0	
NumHmvpIbcCand = 0	
<u>resetIbcBuf = 1</u>	
}	
CtbAddrInRs = CtbAddrBsToRs[CtbAddrInBs]	
coding_tree_unit()	
if(entropy_coding_sync_enabled_flag && ((j + 1) % BrickWidth[SliceBrickIdx[i]] == 0)) {	
end_of_subset_one_bit /* bằng 1 */	ae(v)
if(j < NumCtusInBrick[SliceBrickIdx[i]] - 1)	
byte_alignment()	
}	
}	
if(!entropy_coding_sync_enabled_flag) {	
end_of_brick_one_bit /* bằng 1 */	ae(v)
if(i < NumBricksInCurrSlice - 1)	
byte_alignment()	
}	
}	

7.4.8.5 Ngữ nghĩa CU

Khi tất cả các điều kiện sau là true, danh sách HMVP đối với vùng danh sách ứng viên hợp nhất được chia sẻ được cập nhật bằng cách thiết lập NumHmvpSmrIbcCand bằng NumHmvpIbcCand, và thiết lập HmvpSmrIbcCandList[i] bằng HmvpIbcCandList[i] cho $i = 0..NumHmvpIbcCand - 1$:

- IsInSmr[x0][y0] bằng TRUE.
- SmrX[x0][y0] bằng x0.
- SmrY[x0][y0] bằng y0.

Các phép gán sau được thực hiện cho $x = x0..x0 + cbWidth - 1$ và $y = y0..y0 + cbHeight - 1$:

$$CbPosX[x][y] = x0 \quad (7-135)$$

$$CbPosY[x][y] = y0 \quad (7-136)$$

$$CbWidth[x][y] = cbWidth \quad (7-137)$$

$$CbHeight[x][y] = cbHeight \quad (7-138)$$

Thiết lập vSize là min(ctbSize, 64) và wIbcBufY là (128*128/CtbSizeY).

ibcBufL là mảng có chiều rộng bằng wIbcBufY và chiều cao bằng CtbSizeY.

ibcBufCb và ibcBufCr là các mảng có chiều rộng là wIbcBufC = (wIbcBufY/SubWidthC) và chiều cao là (CtbSizeY/SubHeightC), tức là, CtbSizeC.

Nếu resetIbcBuf bằng 1, phần sau áp dụng

– $ibcBufL[x \% wIbcBufY][y \% CtbSizeY] = -1$, với $x = x0..x0 + wIbcBufY - 1$ and $y = y0..y0 + CtbSizeY - 1$

– $ibcBufCb[x \% wIbcBufC][y \% CtbSizeC] = -1$, với $x = x0..x0 + wIbcBufC - 1$ and $y = y0..y0 + CtbSizeC - 1$

– $ibcBufCr[x \% wIbcBufC][y \% CtbSizeC] = -1$, với $x = x0..x0 + wIbcBufC - 1$ and $y = y0..y0 + CtbSizeC - 1$

– resetIbcBuf = 0

Khi (x0 % vSizeY) bằng 0 and (y0 % vSizeY) bằng 0, phần sau áp dụng

– $ibcBufL[x \% wIbcBufY][y \% CtbSizeY] = -1$, for $x = x0..x0 + vSize - 1$ và $y = y0..y0 + vSize - 1$

– $ibcBufCb[x \% wIbcBufC][y \% CtbSizeC] = -1$, với $x = x0/SubWidthC..x0/SubWidthC + vSize/SubWidthC - 1$ và $y = y0/SubHeightC..y0/SubHeightC + vSize/SubHeightC - 1$

8.6.2 Quá trình dẫn xuất đối với các thành phần vector chuyển động cho các khối IBC

65

Lưu ý 1– Các giá trị thu được của $mvL[0]$ và $mvL[1]$ như được nêu trên sẽ luôn nằm trọn trong khoảng từ -2^{17} đến $2^{17} - 1$.

Quá trình cập nhật cho danh sách HMVP như được xác định trong mệnh đề 8.6.2.6 được gọi với vector chuyển động độ sáng mvL .

Mệnh đề 8.6.2.5 được gọi với mvL làm đầu vào và mvC làm đầu ra.

Tương thích dòng bit yêu cầu rằng vector khối độ sáng mvL sẽ tuân theo các giới hạn sau:

- $((yCb + (mvL[1] >> 4)) \% CtbSizeY) + cbHeight$ nhỏ hơn hoặc bằng $CtbSizeY$
- Với $x = xCb..xCb + cbWidth - 1$ and $y = yCb..yCb + cbHeight - 1$, $ibcBufL[(x + (mvL[0] >> 4)) \% wIbcBufY][(y + (mvL[1] >> 4)) \% CtbSizeY]$ sẽ không bằng -1 .
- Nếu $treeType$ bằng $SINGLE TREE$, for $x = xCb..xCb + cbWidth - 1$ và $y = yCb..yCb + cbHeight - 1$, $ibcBufCb[(x + (mvC[0] >> 5)) \% wIbcBufC][(y + (mvC[1] >> 5)) \% CtbSizeC]$ sẽ không bằng -1 .

8.6.3 Quá trình giải mã cho các khối ibc

8.6.3.1 Tổng quát

Quá trình này được gọi khi giải mã CU được mã hóa trong chế độ dự báo ibc.

Các đầu vào cho quá trình này là:

- vị trí độ sáng (xCb, yCb) xác định mẫu trên bên trái của CB hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến $cbWidth$ xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,
- biến $cbHeight$ xác định chiều cao của CB hiện tại trong các mẫu độ sáng,
- chỉ số thành phần màu của khối hiện tại.
- vector chuyển động mv ,
- mảng $(wIbcBufY)x(CtbSizeY)$ $ibcBufL$, mảng $(wIbcBufC)x(CtbSizeC)$ $ibcBufCb$, mảng $(wIbcBufC)x(CtbSizeC)$ $ibcBufC$.

Các đầu ra của quá trình này là:

- mảng $predSamples$ của các mẫu dự báo.

Với $x = xCb..xCb + Width - 1$ and $y = yCb..yCb + Height - 1$, phần sau áp dụng

Nếu cIdx bằng 0

$$\underline{\underline{predSamples[x][y] = ibcBuf_L[(x + mv[0] >> 4)) \% wIbcBufY][(y + (mv[1] >> 4)) \% CtbSizeY]}}$$

nếu cIdx bằng 1

$$\underline{\underline{predSamples[x][y] = ibcBuf_{Cb}[(x + mv[0] >> 5)) \% wIbcBufC][(y + (mv[1] >> 5)) \% CtbSizeC]}}$$

nếu cIdx bằng 2

$$\underline{\underline{predSamples[x][y] = ibcBuf_{Cr}[(x + mv[0] >> 5)) \% wIbcBufC][(y + (mv[1] >> 5)) \% CtbSizeC]}}$$

8.7.5 Quá trình tái tạo ảnh

8.7.5.1 Tổng quát

Các đầu vào cho quá trình này là:

- vị trí (xCurr, yCurr) xác định mẫu trên bên trái của khối hiện tại so với mẫu trên bên trái của thành phần ảnh hiện tại,
- các biến nCurrSw và nCurrSh lần lượt xác định chiều rộng và chiều cao của khối hiện tại,
- biến cIdx xác định thành phần màu của khối hiện tại,
- mảng (nCurrSw) x (nCurrSh) predSamples xác định các mẫu được dự báo của khối hiện tại,
- mảng (nCurrSw) x (nCurrSh) resSamples xác định các mẫu dư của khối hiện tại.

Đầu ra của quá trình này là mảng mẫu ảnh được tái tạo resSamples và các mảng bộ đệm IBC ibcBuf_L, ibcBuf_{Cb}, ibcBuf_{Cr}.

Tùy thuộc giá trị của thành phần màu cIdx, các phép gán sau được thực hiện:

- nếu cIdx bằng 0, recSamples tương ứng với mảng mẫu ảnh được tái tạo S_L và hàm clipCidx1 tương ứng với Clip1_Y.
- Ngược lại, nếu cIdx bằng 1, tuCbfChroma được gán bằng tu_cbf_cb[xCurr][yCurr], recSamples tương ứng với mảng mẫu sắc độ được tái tạo S_{Cb} và hàm clipCidx1 tương ứng với Clip1_C.

– Ngược lại ($cIdx$ bằng 2), $tuCbfChroma$ được gán bằng $tu_cbf_cr[xCurr][yCurr]$, $recSamples$ tương ứng với mảng mẫu sắc độ được tái tạo S_{Cr} và hàm $clipCidx1$ tương ứng với $Clip1c$.

Tùy thuộc giá trị của $slice_lmcs_enabled_flag$, phần sau áp dụng:

– Nếu $slice_lmcs_enabled_flag$ bằng 0, khối $(nCurrSw) \times (nCurrSh)$ của các mẫu được tái tạo $recSamples$ ở vị trí $(xCurr, yCurr)$ được suy ra như sau cho $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$:

$$recSamples[xCurr + i][yCurr + j] = clipCidx1(predSamples[i][j] + resSamples[i][j]) \quad (8-992)$$

– Ngược lại ($slice_lmcs_enabled_flag$ bằng 1), phần sau áp dụng:

– Nếu $cIdx$ bằng 0, phần sau áp dụng:

– tái tạo ảnh có quá trình ánh xạ cho các mẫu độ sáng như được xác định trong mệnh đề 8.7.5.2 được gọi với vị trí độ sáng $(xCurr, yCurr)$, chiều rộng khối $nCurrSw$ và chiều cao $nCurrSh$, mảng mẫu độ sáng được dự báo $predSamples$, và mảng mẫu độ sáng dư $resSamples$ làm đầu vào, và đầu ra là mảng mẫu độ sáng được tái tạo $recSamples$.

– Ngược lại ($cIdx$ lớn hơn 0), tái tạo ảnh có quá trình chia tỷ lệ phần dư sắc độ phụ thuộc độ sáng cho các mẫu sắc độ như được xác định trong mệnh đề 8.7.5.3 được gọi với vị trí sắc độ $(xCurr, yCurr)$, chiều rộng khối biến đổi $nCurrSw$ và chiều cao $nCurrSh$, cờ khối được mã hóa của khối biến đổi sắc độ hiện tại $tuCbfChroma$, mảng mẫu sắc độ được dự báo $predSamples$, và mảng mẫu sắc độ dư $resSamples$ làm đầu vào, và đầu ra là mảng mẫu sắc độ được tái tạo $recSamples$.

Sau khi giải mã CU hiện tại, phần sau có thể áp dụng:

Nếu $cIdx$ bằng 0, và nếu $treeType$ bằng SINGLE TREE hoặc DUAL TREE LUMA, phần sau áp dụng

$$recSamples[xCurr + i][yCurr + j] = \frac{ibcBufL[(xCurr + i) \% wIbcBufY][(yCurr + j) \% CtbSizeY] + 1}{2}$$

với $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$.

Nếu $cIdx$ bằng 1, và nếu $treeType$ bằng SINGLE TREE hoặc DUAL TREE CHROMA, phần sau áp dụng

$ibcBuf_{Cb}[(xCurr + i) \% wIbcBufC][(yCurr + j) \% CtbSizeC] =$
 $recSamples[xCurr + i][yCurr + j]$

với $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$.

Nếu $cIdx$ bằng 2, và nếu $treeType$ bằng *SINGLE TREE* or
DUAL TREE CHROMA, phần sau áp dụng

$ibcBuf_{Cr}[(xCurr + i) \% wIbcBufC][(yCurr + j) \% CtbSizeC] =$
 $recSamples[xCurr + i][yCurr + j]$

với $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$.

5.19 Phương án thực hiện #19

Các thay đổi trong một số ví dụ được chỉ báo trong văn bản in đậm, gạch chân theo sáng chế.

7.3.7 Cú pháp dữ liệu lát

7.3.7.1 Cú pháp dữ liệu lát tổng quát

	Mô tả
slice_data() {	
for(i = 0; i < NumBricksInCurrSlice; i++) {	
CtbAddrInBs = FirstCtbAddrBs[SliceBrickIdx[i]]	
for(j = 0; j < NumCtusInBrick[SliceBrickIdx[i]]; j++, CtbAddrInBs++) {	
if((j % BrickWidth[SliceBrickIdx[i]]) == 0) {	
NumHmvpCand = 0	
NumHmvpIbcCand = 0	
<u>resetIbcBuf = 1</u>	
}	
CtbAddrInRs = CtbAddrBsToRs[CtbAddrInBs]	
coding tree unit()	
if(entropy_coding_sync_enabled_flag &&	
((j + 1) % BrickWidth[SliceBrickIdx[i]] == 0)) {	
<u>end of subset one bit</u> /* bằng 1 */	ae(v)
if(j < NumCtusInBrick[SliceBrickIdx[i]] - 1)	
byte_alignment()	
}	
}	
}	
if(!entropy_coding_sync_enabled_flag) {	
<u>end of brick one bit</u> /* bằng 1 */	ae(v)
if(i < NumBricksInCurrSlice - 1)	
byte_alignment()	
}	
}	
}	

7.4.8.5 Ngữ nghĩa CU

Khi tất cả các điều kiện sau là đúng, danh sách HMVP đối với vùng danh sách ứng viên hợp nhất được chia sẻ được cập nhật bằng cách thiết lập NumHmvpSmrIbcCand bằng NumHmvpIbcCand, và thiết lập HmvpSmrIbcCandList[i] bằng HmvpIbcCandList[i] với $i = 0..NumHmvpIbcCand - 1$:

- IsInSmr[x0][y0] bằng TRUE.
- SmrX[x0][y0] bằng x0.
- SmrY[x0][y0] bằng y0.

Các phép gán sau được thực hiện với $x = x0..x0 + cbWidth - 1$ và $y = y0..y0 + cbHeight - 1$:

$$CbPosX[x][y] = x0 \quad (7-135)$$

$$CbPosY[x][y] = y0 \quad (7-136)$$

$$CbWidth[x][y] = cbWidth \quad (7-137)$$

$$CbHeight[x][y] = cbHeight \quad (7-138)$$

Thiết lập vSize làm min(ctbSize, 64) và wIbcBufY làm (128*128/CtbSizeY).

ibcBufL là mảng có chiều rộng là wIbcBufY và chiều cao là CtbSizeY.

ibcBufCb và ibcBufCr là các mảng có chiều rộng là wIbcBufC = (wIbcBufY/SubWidthC) và chiều cao là (CtbSizeY/SubHeightC), tức là, CtbSizeC.

Nếu resetIbcBuf bằng 1, phần sau áp dụng

– $ibcBufL[x \% wIbcBufY][y \% CtbSizeY] = -1$, với $x = x0..x0 + wIbcBufY - 1$ and $y = y0..y0 + CtbSizeY - 1$

– $ibcBufCb[x \% wIbcBufC][y \% CtbSizeC] = -1$, với $x = x0..x0 + wIbcBufC - 1$ and $y = y0..y0 + CtbSizeC - 1$

– $ibcBufCr[x \% wIbcBufC][y \% CtbSizeC] = -1$, for $x = x0..x0 + wIbcBufC - 1$ and $y = y0..y0 + CtbSizeC - 1$

– resetIbcBuf = 0

Khi (x0 % vSizeY) bằng 0 và (y0 % vSizeY) bằng 0, phần sau áp dụng

– $ibcBufL[x \% wIbcBufY][y \% CtbSizeY] = -1$, với $x = x0..x0 + \min(vSize, cbWidth) - 1$ và $y = y0..y0 + \min(vSize, cbHeight) - 1$

$$\begin{aligned}
& - \text{ibcBufCb}[x \% wIbcBufC][y \% CtbSizeC] = -1, \text{ với} \\
& x = x0/SubWidthC..x0/SubWidthC + \min(vSize/SubWidthC, cbWidth) - 1 \text{ và} \\
& y = y0/SubHeightC..y0/SubHeightC + \min(vSize/SubHeightC, cbHeight) - 1 \\
& - \text{ibcBufCr}[x \% wIbcBufC][y \% CtbSizeC] = -1, \text{ với} \\
& x = x0/SubWidthC..x0/SubWidthC + \min(vSize/SubWidthC, cbWidth) - 1 \text{ và} \\
& y = y0/SubHeightC..y0/SubHeightC + \min(vSize/SubHeightC, cbHeight) - 1
\end{aligned}$$

8.6.2 Quá trình dẫn xuất cho các thành phần vec tơ chuyển đổi cho các khối IBC

8.6.2.1 Tổng quát

Các đầu vào cho quá trình này là:

- vị trí độ sáng (xCb, yCb) của mẫu trên bên trái của khối mã hóa độ sáng hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,

- biến cbWidth xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,
- biến cbHeight xác định chiều cao của CB hiện tại trong các mẫu độ sáng.

Các đầu ra của quá trình này là:

- vector chuyển động độ sáng có độ chính xác mẫu phân số 1/16 mvL.

Vector chuyển động độ sáng mvL được suy ra như sau:

- Quá trình dẫn xuất để dự báo vector chuyển động độ sáng IBC như được xác định trong mệnh đề 8.6.2.2 được gọi với vị trí độ sáng (xCb, yCb), các biến cbWidth và cbHeight làm đầu vào, và đầu ra là vector chuyển động độ sáng mvL.

- Khi general_merge_flag[xCb][yCb] bằng 0, phần sau áp dụng:

7. Biến mvd được suy ra như sau:

$$mvd[0] = MvdL0[xCb][yCb][0] \quad (8-883)$$

$$mvd[1] = MvdL0[xCb][yCb][1] \quad (8-884)$$

8. Quá trình làm tròn cho các vector chuyển động như được xác định trong mệnh đề 8.5.2,14 được gọi với mvX được thiết lập bằng mvL, rightShift được thiết lập bằng MvShift + 2, và leftShift được thiết lập bằng MvShift + 2 làm đầu vào và mvL được làm tròn làm đầu ra.

9. Vector chuyển động độ sáng mvL được chỉnh sửa như sau:

$$u[0] = (mvL[0] + mvd[0] + 2^{18}) \% 2^{18} \quad (8-885)$$

$$mvL[0] = (u[0] \geq 2^{17}) ? (u[0] - 2^{18}) : u[0] \quad (8-886)$$

$$u[1] = (mvL[1] + mvd[1] + 2^{18}) \% 2^{18}(8-887)$$

$$mvL[1] = (u[1] \geq 2^{17}) ? (u[1] - 2^{18}) : u[1] \quad (8-888)$$

Lưu ý 1– Các giá trị thu được của $mvL[0]$ và $mvL[1]$ như được nêu trên đây sẽ luôn nằm trọn trong khoảng từ -2^{17} đến $2^{17} - 1$.

Quá trình cập nhật đối với danh sách HMVP như được xác định trong mệnh đề 8.6.2.6 được gọi với vector chuyển động độ sáng mvL .

Mệnh đề 8.6.2.5 được gọi với mvL làm đầu vào và mvC làm đầu ra.

Tương thích dòng bit yêu cầu rằng vector khối độ sáng mvL sẽ tuân theo các giới hạn sau:

- $((yCb + (mvL[1] \gg 4)) \% CtbSizeY) + cbHeight$ nhỏ hơn hoặc bằng $CtbSizeY$
- Với $x = xCb..xCb + cbWidth - 1$ and $y = yCb..yCb + cbHeight - 1$, $ibcBufL[(x + (mvL[0] \gg 4)) \% wIbcBufY][(y + (mvL[1] \gg 4)) \% CtbSizeY]$ sẽ không bằng -1 .
- Nếu $treeType$ bằng $SINGLE\ TREE$, for $x = xCb..xCb + cbWidth - 1$ và $y = yCb..yCb + cbHeight - 1$, $ibcBufCb[(x + (mvC[0] \gg 5)) \% wIbcBufC][(y + (mvC[1] \gg 5)) \% CtbSizeC]$ sẽ không bằng -1 .

8.6.3 Quá trình giải mã cho các khối IBC

8.6.3.1 Tổng quát

Quá trình này được gọi khi giải mã CU được mã hóa trong chế độ dự báo IBC.

Các đầu vào cho quá trình này là:

- vị trí độ sáng (xCb, yCb) xác định mẫu trên bên trái của CB hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến $cbWidth$ xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,
- biến $cbHeight$ xác định chiều cao của CB hiện tại trong các mẫu độ sáng,
- biến $cIdx$ xác định chỉ số thành phần màu của khối hiện tại.
- vector chuyển động mv ,
- mảng $(wIbcBufY)x(CtbSizeY)$ $ibcBufL$, mảng $(wIbcBufC)x(CtbSizeC)$ $ibcBufCb$, mảng $(wIbcBufC)x(CtbSizeC)$ $ibcBufCr$.

Các đầu ra của quá trình này là:

- mảng $predSamples$ của các mẫu dự báo.

Với $x = xCb..xCb + Width - 1$ and $y = yCb..yCb + Height - 1$, phần sau áp dụng

Nếu $cIdx$ bằng 0

$$\underline{predSamples[x][y] = ibcBuf_L[(x + mv[0] >> 4)) \% wIbcBufY][(y + (mv[1] >> 4)) \% CtbSizeY]}$$

nếu $cIdx$ bằng 1

$$\underline{predSamples[x][y] = ibcBuf_{Cb}[(x + mv[0] >> 5)) \% wIbcBufC][(y + (mv[1] >> 5)) \% CtbSizeC]}$$

nếu $cIdx$ bằng 2

$$\underline{predSamples[x][y] = ibcBuf_{Cr}[(x + mv[0] >> 5)) \% wIbcBufC][(y + (mv[1] >> 5)) \% CtbSizeC]}$$

8.7.5 Quá trình tái tạo ảnh

8.7.5.1 Tổng quát

Các đầu vào cho quá trình này là:

- vị trí $(xCurr, yCurr)$ xác định mẫu trên bên trái của khối hiện tại so với mẫu trên bên trái của thành phần ảnh hiện tại,
- các biến $nCurrSw$ và $nCurrSh$ lần lượt xác định chiều rộng và chiều cao của khối hiện tại,
- biến $cIdx$ xác định thành phần màu của khối hiện tại,
- mảng $(nCurrSw) \times (nCurrSh)$ $predSamples$ xác định các mẫu được dự báo của khối hiện tại,
- mảng $(nCurrSw) \times (nCurrSh)$ $resSamples$ xác định các mẫu dư của khối hiện tại.

Đầu ra của quá trình này là mảng mẫu ảnh được tái tạo $resSamples$ và các mảng bộ đệm IBC $ibcBuf_L$, $ibcBuf_{Cb}$, $ibcBuf_{Cr}$.

Tùy thuộc giá trị của thành phần màu $cIdx$, các phép gán sau được thực hiện:

- Nếu $cIdx$ bằng 0, $recSamples$ tương ứng với mảng mẫu ảnh được tái tạo S_L và hàm $clipCidx1$ tương ứng với $Clip1_Y$.

– Ngược lại, nếu $cIdx$ bằng 1, $tuCbfChroma$ được gán bằng $tu_cbf_cb[xCurr][yCurr]$, $recSamples$ tương ứng với mảng mẫu sắc độ được tái tạo S_{Cb} và hàm $clipCidx1$ tương ứng với $Clip1c$.

– Ngược lại ($cIdx$ bằng 2), $tuCbfChroma$ được gán bằng $tu_cbf_cr[xCurr][yCurr]$, $recSamples$ tương ứng với mảng mẫu sắc độ được tái tạo S_{Cr} và hàm $clipCidx1$ tương ứng với $Clip1c$.

Tùy thuộc giá trị của $slice_lmcs_enabled_flag$, phần sau áp dụng:

– Nếu $slice_lmcs_enabled_flag$ bằng 0, khối $(nCurrSw) \times (nCurrSh)$ của các mẫu được tái tạo $recSamples$ ở vị trí $(xCurr, yCurr)$ được suy ra như sau với $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$:

$recSamples[xCurr + i][yCurr + j] = clipCidx1(predSamples[i][j] + resSamples[i][j])$ (8-992)

– Ngược lại ($slice_lmcs_enabled_flag$ bằng 1), phần sau áp dụng:

– Nếu $cIdx$ bằng 0, phần sau áp dụng:

– tái tạo ảnh có quá trình ánh xạ cho các mẫu độ sáng như được xác định trong mệnh đề 8.7.5.2 được gọi với vị trí độ sáng $(xCurr, yCurr)$, chiều rộng khối $nCurrSw$ và chiều cao $nCurrSh$, mảng mẫu độ sáng được dự báo $predSamples$, và mảng mẫu độ sáng dư $resSamples$ làm đầu vào, và đầu ra là mảng mẫu độ sáng được tái tạo $recSamples$.

– Ngược lại ($cIdx$ lớn hơn 0), tái tạo ảnh có quá trình chia tỷ lệ phần dư sắc độ phụ thuộc độ sáng cho các mẫu sắc độ như được xác định trong mệnh đề 8.7.5.3 được gọi với vị trí sắc độ $(xCurr, yCurr)$, chiều rộng khối biến đổi $nCurrSw$ và chiều cao $nCurrSh$, cờ khối được mã hóa của khối biến đổi sắc độ hiện tại $tuCbfChroma$, mảng mẫu sắc độ được dự báo $predSamples$, và mảng mẫu sắc độ dư $resSamples$ làm đầu vào, và đầu ra là mảng mẫu sắc độ được tái tạo $recSamples$.

Sau khi giải mã CU hiện tại, phần sau có thể áp dụng:

Nếu $cIdx$ bằng 0, và nếu $treeType$ bằng SINGLE TREE hoặc DUAL TREE LUMA, phần sau áp dụng

$ibcBuf1[(xCurr + i) \% wIbcBufY][(yCurr + j) \% CtbSizeY] =$
 $recSamples[xCurr + i][yCurr + j]$

với $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$.

Nếu $cIdx$ bằng 1, và nếu $treeType$ bằng SINGLE TREE hoặc DUAL TREE CHROMA, phần sau áp dụng

$ibcBufCb[(xCurr + i) \% wIbcBufC][(yCurr + j) \% CtbSizeC] =$
 $recSamples[xCurr + i][yCurr + j]$

với $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$.

Nếu $cIdx$ bằng 2, và nếu $treeType$ bằng SINGLE TREE hoặc DUAL TREE CHROMA, phần sau áp dụng

$ibcBufCr[(xCurr + i) \% wIbcBufC][(yCurr + j) \% CtbSizeC] =$
 $recSamples[xCurr + i][yCurr + j]$

với $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$.

5.20 Phương án thực hiện #20

Các thay đổi trong một số ví dụ được chỉ báo trong văn bản được in đậm, được gạch chân, in nghiêng theo sáng chế.

7.3.7 Cú pháp dữ liệu lát

7.3.7.1 Cú pháp dữ liệu lát tổng quát

	Mô tả
slice_data() {	
for(i = 0; i < NumBricksInCurrSlice; i++) {	
CtbAddrInBs = FirstCtbAddrBs[SliceBrickIdx[i]]	
for(j = 0; j < NumCtusInBrick[SliceBrickIdx[i]]; j++, CtbAddrInBs++) {	
if((j % BrickWidth[SliceBrickIdx[i]]) == 0) {	
NumHmvpCand = 0	
NumHmvpIbcCand = 0	
<u>resetIbcBuf = 1</u>	
}	
CtbAddrInRs = CtbAddrBsToRs[CtbAddrInBs]	
coding_tree_unit()	
if(entropy_coding_sync_enabled_flag &&	
((j + 1) % BrickWidth[SliceBrickIdx[i]] == 0)) {	
end of subset one bit /* bằng 1 */	ae(v)
if(j < NumCtusInBrick[SliceBrickIdx[i]] - 1)	
byte_alignment()	
}	
}	
if(!entropy_coding_sync_enabled_flag) {	
end of brick one bit /* bằng 1 */	ae(v)
if(i < NumBricksInCurrSlice - 1)	
byte_alignment()	
}	

}	
}	

7.4.8.5 Ngữ nghĩa CU

Khi tất cả các điều kiện sau là đúng, danh sách HMVP đối với vùng danh sách ứng viên hợp nhất được chia sẻ được cập nhật bằng cách thiết lập NumHmvpSmrIbcCand bằng NumHmvpIbcCand, và thiết lập HmvpSmrIbcCandList[i] bằng HmvpIbcCandList[i] cho $i = 0..NumHmvpIbcCand - 1$:

- IsInSmr[x0][y0] bằng TRUE.
- SmrX[x0][y0] bằng x0.
- SmrY[x0][y0] bằng y0.

Các phép gán sau được thực hiện for $x = x0..x0 + cbWidth - 1$ và $y = y0..y0 + cbHeight - 1$:

$$CbPosX[x][y] = x0 \quad (7-135)$$

$$CbPosY[x][y] = y0 \quad (7-136)$$

$$CbWidth[x][y] = cbWidth \quad (7-137)$$

$$CbHeight[x][y] = cbHeight \quad (7-138)$$

Thiết lập vSize làm min(ctbSize, 64) và wIbcBufY làm (128*128/CtbSizeY).
ibcBufL là mảng có chiều rộng là wIbcBufY và chiều cao bằng CtbSizeY.
ibcBufCb và ibcBufCr là các mảng có chiều rộng là wIbcBufC
= (wIbcBufY/SubWidthC) và chiều cao là (CtbSizeY/SubHeightC), tức là,
CtbSizeC.

Nếu resetIbcBuf bằng 1, phần sau áp dụng

– $ibcBufL[x \% wIbcBufY][y \% CtbSizeY] = -1$, cho
 $x = x0..x0 + wIbcBufY - 1$ and $y = y0..y0 + CtbSizeY - 1$

– $ibcBufCb[x \% wIbcBufC][y \% CtbSizeC] = -1$, cho
 $x = x0..x0 + wIbcBufC - 1$ and $y = y0..y0 + CtbSizeC - 1$

– $ibcBufCr[x \% wIbcBufC][y \% CtbSizeC] = -1$, cho
 $x = x0..x0 + wIbcBufC - 1$ and $y = y0..y0 + CtbSizeC - 1$

– resetIbcBuf = 0

Khi $(x0 \% vSizeY)$ bằng 0 và $(y0 \% vSizeY)$ bằng 0, phần sau áp dụng

- $\frac{ibcBuf_L[x \% wIbcBufY][y \% CtbSizeY] - 1}{x = x0..x0 + \max(vSize, cbWidth) - 1 \text{ và } y = y0..y0 + \max(vSize, cbHeight) - 1} = -1$, cho
- $\frac{ibcBuf_{Cb}[x \% wIbcBufC][y \% CtbSizeC] - 1}{x = x0/SubWidthC..x0/SubWidthC + \max(vSize/SubWidthC, cbWidth) - 1 \text{ và } y = y0/SubHeightC..y0/SubHeightC + \max(vSize/SubHeightC, cbHeight) - 1} = -1$, cho
- $\frac{ibcBuf_{Cr}[x \% wIbcBufC][y \% CtbSizeC] - 1}{x = x0/SubWidthC..x0/SubWidthC + \max(vSize/SubWidthC, cbWidth) - 1 \text{ và } y = y0/SubHeightC..y0/SubHeightC + \max(vSize/SubHeightC, cbHeight) - 1} = -1$, cho

8.6.2 Quá trình dẫn xuất cho các thành phần vec tơ chuyển đổi cho các khối IBC

8.6.2.1 Tổng quát

Các đầu vào cho quá trình này là:

- vị trí độ sáng (xCb, yCb) của mẫu trên bên trái của khối mã hóa độ sáng hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến cbWidth xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,
- biến cbHeight xác định chiều cao của CB hiện tại trong các mẫu độ sáng.

Các đầu ra của quá trình này là:

- vector chuyển động độ sáng có độ chính xác mẫu phân số 1/16 mvL.

Vector chuyển động độ sáng mvL được suy ra như sau:

- Quá trình dẫn xuất để dự báo vector chuyển động độ sáng IBC như được xác định trong mệnh đề 8.6.2.2 được gọi với vị trí độ sáng (xCb, yCb), các biến cbWidth và cbHeight làm đầu vào, và đầu ra là vector chuyển động độ sáng mvL.

- Khi general_merge_flag[xCb][yCb] bằng 0, phân sau áp dụng:

10. Biến mvd được suy ra như sau:

$$mvd[0] = MvdL0[xCb][yCb][0] \quad (8-883)$$

$$mvd[1] = MvdL0[xCb][yCb][1] \quad (8-884)$$

11. Quá trình làm tròn cho các vector chuyển động như được xác định trong mệnh đề 8.5.2,14 được gọi với mvX được thiết lập bằng mvL, rightShift được thiết lập bằng MvShift + 2, và leftShift được thiết lập bằng MvShift + 2 làm các đầu vào và mvL được làm tròn làm đầu ra.

12. Vector chuyển động độ sáng mvL được chỉnh sửa như sau:

$$u[0] = (mvL[0] + mvd[0] + 2^{18}) \% 2^{18} \quad (8-885)$$

$$mvL[0] = (u[0] \geq 2^{17}) ? (u[0] - 2^{18}) : u[0] \quad (8-886)$$

$$u[1] = (mvL[1] + mvd[1] + 2^{18}) \% 2^{18} \quad (8-887)$$

$$mvL[1] = (u[1] \geq 2^{17}) ? (u[1] - 2^{18}) : u[1] \quad (8-888)$$

Lưu ý 1– Các giá trị thu được của mvL[0] và mvL[1] như được nêu trên đây sẽ luôn nằm trọn trong khoảng từ -2^{17} đến $2^{17} - 1$.

Quá trình cập nhật cho danh sách HMVP như được xác định trong mệnh đề 8.6.2,6 được gọi với vector chuyển động độ sáng mvL.

Mệnh đề 8.6.2,5 được gọi với mvL làm đầu vào và mvC làm đầu ra.

Tương thích dòng bit yêu cầu rằng vector khối độ sáng mvL sẽ tuân theo các giới hạn sau:

- $((yCb + (mvL[1] >> 4)) \% CtbSizeY) + cbHeight$ nhỏ hơn hoặc bằng $CtbSizeY$
- Với $x = xCb..xCb + cbWidth - 1$ và $y = yCb..yCb + cbHeight - 1$, $ibcBufL[(x + (mvL[0] >> 4)) \% wIbcBufY][(y + (mvL[1] >> 4)) \% CtbSizeY]$ sẽ không bằng -1 .

8.6.3 Quá trình giải mã cho các khối IBC

8.6.3.1 Tổng quát

Quá trình này được gọi khi giải mã CU được mã hóa trong chế độ dự báo IBC.

Các đầu vào cho quá trình này là:

- vị trí độ sáng (xCb, yCb) xác định mẫu trên bên trái của CB hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến cbWidth xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,
- biến cbHeight xác định chiều cao của CB hiện tại trong các mẫu độ sáng,
- biến cIdx xác định chỉ số thành phần màu của khối hiện tại.
- vector chuyển động mv,
- mảng $(wIbcBufY)x(CtbSizeY)$ ibcBufL, mảng $(wIbcBufC)x(CtbSizeC)$ ibcBufCb, mảng $(wIbcBufC)x(CtbSizeC)$ ibcBufCr.

Các đầu ra của quá trình này là:

- mảng predSamples của các mẫu dự báo.

Với $x = xCb..xCb + Width - 1$ và $y = yCb..yCb + Height - 1$, phần sau áp dụng

Nếu cIdx bằng 0

$$\underline{predSamples[x][y] = ibcBuf_L[(x + mv[0] >> 4) \% wIbcBufY][(y + (mv[1] >> 4) \% CtbSizeY)]}$$

nếu cIdx bằng 1

$$\underline{predSamples[x][y] = ibcBuf_{Cb}[(x + mv[0] >> 5) \% wIbcBufC][(y + (mv[1] >> 5) \% CtbSizeC)]}$$

nếu cIdx bằng 2

$$\underline{predSamples[x][y] = ibcBuf_{Cr}[(x + mv[0] >> 5) \% wIbcBufC][(y + (mv[1] >> 5) \% CtbSizeC)]}$$

8.7.5 Quá trình tái tạo ảnh

8.7.5.1 Tổng quát

Các đầu vào cho quá trình này là:

- vị trí (xCurr, yCurr) xác định mẫu trên bên trái của khối hiện tại so với mẫu trên bên trái của thành phần ảnh hiện tại,
- các biến nCurrSw and nCurrSh xác định chiều rộng và chiều cao, một cách lần lượt, của khối hiện tại,
- biến cIdx xác định thành phần màu của khối hiện tại,
- mảng (nCurrSw) x (nCurrSh) predSamples xác định các mẫu được dự báo của khối hiện tại,
- mảng (nCurrSw) x (nCurrSh) resSamples xác định các mẫu dư của khối hiện tại.

Đầu ra của quá trình này là mảng mẫu ảnh được tái tạo resSamples và các mảng bộ đệm IBC ibcBuf_L, ibcBuf_{Cb}, ibcBuf_{Cr}.

Tùy thuộc giá trị của thành phần màu cIdx, các phép gán sau được thực hiện:

- Nếu cIdx bằng 0, recSamples tương ứng với mảng mẫu ảnh được tái tạo S_L và hàm clipCidx1 tương ứng với Clip_{1Y}.

– Ngược lại, nếu $cIdx$ bằng 1, $tuCbfChroma$ được gán bằng $tu_cbf_cb[xCurr][yCurr]$, $recSamples$ tương ứng với mảng mẫu sắc độ được tái tạo S_{Cb} và hàm $clipCidx1$ tương ứng với $Clip1c$.

– Ngược lại ($cIdx$ bằng 2), $tuCbfChroma$ được gán bằng $tu_cbf_cr[xCurr][yCurr]$, $recSamples$ tương ứng với mảng mẫu sắc độ được tái tạo S_{Cr} và hàm $clipCidx1$ tương ứng với $Clip1c$.

Tùy thuộc giá trị của $slice_lmcs_enabled_flag$, phần sau áp dụng:

– Nếu $slice_lmcs_enabled_flag$ bằng 0, khối $(nCurrSw) \times (nCurrSh)$ của các mẫu được tái tạo $recSamples$ ở vị trí $(xCurr, yCurr)$ được suy ra như sau với $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$:

$recSamples[xCurr + i][yCurr + j] = clipCidx1(predSamples[i][j] + resSamples[i][j])$ (8-992)

– Ngược lại ($slice_lmcs_enabled_flag$ bằng 1), phần sau áp dụng:

– Nếu $cIdx$ bằng 0, phần sau áp dụng:

– tái tạo ảnh có quá trình ánh xạ cho các mẫu độ sáng như được xác định trong mệnh đề 8.7.5.2 được gọi với vị trí độ sáng $(xCurr, yCurr)$, chiều rộng khối $nCurrSw$ và chiều cao $nCurrSh$, mảng mẫu độ sáng được dự báo $predSamples$, và mảng mẫu độ sáng dư $resSamples$ làm đầu vào, và đầu ra là mảng mẫu độ sáng được tái tạo $recSamples$.

– Ngược lại ($cIdx$ lớn hơn 0), tái tạo ảnh có quá trình chia tỷ lệ phần dư sắc độ phụ thuộc độ sáng cho các mẫu sắc độ như được xác định trong mệnh đề 8.7.5.3 được gọi với vị trí sắc độ $(xCurr, yCurr)$, chiều rộng khối biến đổi $nCurrSw$ và chiều cao $nCurrSh$, cờ khối được mã hóa của khối biến đổi sắc độ hiện tại $tuCbfChroma$, mảng mẫu sắc độ được dự báo $predSamples$, và mảng mẫu sắc độ dư $resSamples$ làm đầu vào, và đầu ra là mảng mẫu sắc độ được tái tạo $recSamples$.

Sau khi giải mã CU hiện tại, phần sau có thể áp dụng:

Nếu $cIdx$ bằng 0, và nếu $treeType$ bằng SINGLE TREE hoặc DUAL TREE LUMA, phần sau áp dụng

$ibcBufL[(xCurr + i) \% wIbcBufY][(yCurr + j) \% CtbSizeY] = recSamples[xCurr + i][yCurr + j]$

với $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$.

Nếu $cIdx$ bằng 1, và nếu $treeType$ bằng SINGLE TREE hoặc DUAL TREE CHROMA, phần sau áp dụng

$$\underline{ibcBufCb[(xCurr + i) \% wIbcBufC][(yCurr + j) \% CtbSizeC] = recSamples[xCurr + i][yCurr + j]}$$

với $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$.

Nếu $cIdx$ bằng 2, và nếu $treeType$ bằng SINGLE TREE hoặc DUAL TREE CHROMA, phần sau áp dụng

$$\underline{ibcBufCr[(xCurr + i) \% wIbcBufC][(yCurr + j) \% CtbSizeC] = recSamples[xCurr + i][yCurr + j]}$$

với $i = 0..nCurrSw - 1, j = 0..nCurrSh - 1$.

Fig.6 là lưu đồ của phương pháp lấy làm ví dụ 600 xử lý media hình ảnh (video hoặc ảnh). Phương pháp 600 bao gồm các bước xác định (602), đối với biến đổi giữa khối video hiện tại và biểu diễn dòng bit của khối video hiện tại, kích thước của bộ đệm để lưu trữ các mẫu tham chiếu cho khối video hiện tại nhờ sử dụng chế độ mã hóa IBC, và thực hiện (604) việc biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm.

Các mệnh đề sau mô tả một số dấu hiệu ưu tiên ví dụ được triển khai theo các phương án thực hiện của phương pháp 600 và các phương pháp khác. Các ví dụ bổ sung được nêu trong phần 4 của sáng chế.

1. Phương pháp xử lý video, bao gồm các bước: xác định, đối với biến đổi giữa khối video hiện tại và biểu diễn dòng bit của khối video hiện tại, kích thước của bộ đệm để lưu trữ các mẫu tham chiếu cho khối video hiện tại nhờ sử dụng chế độ IBC; và thực hiện việc biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm.

2. Phương pháp theo mệnh đề 1, trong đó kích thước của bộ đệm là hằng số định trước.

3. Phương pháp theo mệnh đề 1 hoặc 2, trong đó kích thước bằng $M \times N$, trong đó M và N là các số nguyên.

4. Phương pháp theo mệnh đề 3, trong đó $M \times N$ bằng 64×64 hoặc 128×128 hoặc 64×128 .

5. Phương pháp theo mệnh đề 1, trong đó kích thước của bộ đệm bằng kích thước của CTU của khối video hiện tại.

6. Phương pháp theo mệnh đề 1, trong đó kích thước của bộ đệm bằng kích thước của VPDU được sử dụng trong khi biến đổi.

7. Phương pháp theo mệnh đề 1, trong đó kích thước của bộ đệm corresponds trường in biểu diễn dòng bit.

8. Phương pháp theo mệnh đề 7, trong đó trường này được bao gồm trong biểu diễn dòng bit ở mức VPS hoặc SPS hoặc PPS hoặc tiêu đề ảnh hoặc tiêu đề lát hoặc tiêu đề nhóm phiên.

9. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 8, trong đó kích thước của bộ đệm khác nhau với các mẫu tham chiếu đối với thành phần độ sáng và các mẫu tham chiếu cho thành phần sắc độ.

10. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 8, trong đó kích thước của bộ đệm phụ thuộc vào định dạng lấy mẫu phụ sắc độ của khối video hiện tại.

11. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 8, trong đó các mẫu tham chiếu được lưu trữ theo định dạng RGB.

12. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 11, trong đó bộ đệm được sử dụng để lưu trữ các mẫu được tái tạo trước khi lọc vòng và sau khi lọc vòng.

13. Phương pháp theo mệnh đề 12, trong đó lọc vòng bao gồm lọc tách khối hoặc ALF hoặc lọc SAO.

14. Phương pháp xử lý video, bao gồm: khởi tạo, đối với biến đổi giữa khối video hiện tại và biểu diễn dòng bit của khối video hiện tại, bộ đệm để lưu trữ các mẫu tham chiếu cho khối video hiện tại nhờ sử dụng chế độ IBC nhờ sử dụng các giá trị ban đầu cho các mẫu tham chiếu; và thực hiện việc biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm.

15. Phương pháp theo mệnh đề 14, trong đó các giá trị ban đầu tương ứng với hằng số.

16. Phương pháp theo mệnh đề 14 hoặc 15, trong đó các giá trị ban đầu là hàm của chiều sâu bit của khối video hiện tại.

17. Phương pháp theo mệnh đề 15, trong đó hằng số tương ứng với giá trị xám giữa.

18. Phương pháp theo mệnh đề 14, trong đó các giá trị ban đầu tương ứng với các giá trị pixel của khối video được giải mã trước đó.

19. Phương pháp theo mệnh đề 18, trong đó khối video được giải mã trước đó tương ứng với khối được giải mã trước khi lọc trong vòng.

20. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 14 đến 19, trong đó kích thước của bộ đệm được nêu theo một trong các mệnh đề từ 1 đến 13.

21. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 20, trong đó các vị trí pixel trong bộ đệm như được giải quyết nhờ sử dụng các số x và y .

22. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 20, trong đó các vị trí pixel trong bộ đệm như được giải quyết nhờ sử dụng một số mà mở rộng từ 0 đến $M*N-1$, trong đó M và N là chiều rộng pixel và chiều cao pixel của bộ đệm.

23. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 20, trong đó, biểu diễn dòng bit hiện tại bao gồm vector khối để biến đổi, trong đó vector khối, được ký hiệu là (BV_x, BV_y) bằng $(x-x_0, y-y_0)$, trong đó (x_0, y_0) tương ứng với vị trí trên bên trái của CTU của khối video hiện tại.

24. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 20, trong đó, biểu diễn dòng bit hiện tại bao gồm vector khối để biến đổi, trong đó vector khối, được ký hiệu là (BV_x, BV_y) bằng $(x-x_0+Tx, y-y_0+Ty)$, trong đó (x_0, y_0) tương ứng với vị trí trên bên trái của CTU của khối video hiện tại và trong đó T_x và T_y là các giá trị độ lệch.

25. Phương pháp theo mệnh đề 24, trong đó T_x và T_y là các giá trị độ lệch định trước.

26. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 20, trong đó trong khi biến đổi, đối với pixel ở vị trí (x_0, y_0) và có vector khối (BV_x, BV_y) , tham chiếu tương ứng trong bộ đệm được tìm thấy ở vị trí tham chiếu (x_0+BV_x, y_0+BV_y) .

27. Phương pháp theo mệnh đề 26, trong đó trong trường hợp mà vị trí tham chiếu nằm ngoài bộ đệm, tham chiếu trong bộ đệm được xác định bằng cách xén ở biên của bộ đệm.

28. Phương pháp theo mệnh đề 26, trong đó trong trường hợp mà vị trí tham chiếu nằm ngoài bộ đệm, tham chiếu trong bộ đệm được xác định có giá trị định trước.

29. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 20, trong đó trong khi biến đổi, đối với pixel ở vị trí (x_0, y_0) và có vectơ khối (BV_x, BV_y) , tham chiếu tương ứng trong bộ đệm được tìm thấy ở vị trí tham chiếu $((x_0 + BV_x) \bmod M, (y_0 + BV_y) \bmod N)$ trong đó “mod” là phép toán modulo và M và N là các số nguyên biểu diễn các chiều x và y của bộ đệm.

30. Phương pháp xử lý video, bao gồm: thiết lập lại, trong khi biến đổi giữa video và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu đối với mã hóa sao chép bên trong khối ở biên video; và thực hiện việc biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm.

31. Phương pháp theo mệnh đề 30, trong đó biên video tương ứng với ảnh mới hoặc phiên mới.

32. Phương pháp theo mệnh đề 30, trong đó việc biến đổi được thực hiện bằng cách cập nhật, sau khi thiết lập lại, bộ đệm có các giá trị được tái tạo của VPDU.

33. Phương pháp theo mệnh đề 30, trong đó việc biến đổi được thực hiện bằng cách cập nhật, sau khi thiết lập lại, bộ đệm có các giá trị được tái tạo của CTU.

34. Phương pháp theo mệnh đề 30, trong đó việc thiết lập lại được thực hiện ở đầu của mỗi hàng CTU.

35. Phương pháp theo mệnh đề 1, trong đó kích thước của bộ đệm tương ứng với L khối được giải mã trước đó 64×64 , trong đó L là số nguyên.

36. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 35, trong đó thứ tự quét đọc được sử dụng để đọc hoặc lưu trữ các mẫu trong bộ đệm trong khi biến đổi.

37. Phương pháp xử lý video, bao gồm: sử dụng, đối với biến đổi giữa khối video hiện tại và biểu diễn dòng bit của khối video hiện tại, bộ đệm để lưu trữ các mẫu tham chiếu cho khối video hiện tại nhờ sử dụng chế độ IBC, trong đó

chiều sâu bit thứ nhất của bộ đệm khác với chiều sâu bit thứ hai của dữ liệu được mã hóa; và thực hiện việc biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm.

38. Phương pháp theo mệnh đề 37, trong đó chiều sâu bit thứ nhất lớn hơn chiều sâu bit thứ hai.

39. Phương pháp theo mệnh đề 37 hoặc 38, trong đó chiều sâu bit thứ nhất giống hệt chiều sâu bit của bộ đệm tái tạo được sử dụng trong khi biến đổi.

40. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 37 đến 39, trong đó chiều sâu bit thứ nhất được báo hiệu ở biểu diễn dòng bit làm giá trị hoặc giá trị hiệu số.

41. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 37 đến 40, trong đó việc biến đổi sử dụng các chiều sâu bit khác nhau cho các thành phần sắc độ và độ sáng.

Các phương án thực hiện và các ví dụ khác theo các mệnh đề từ 37 đến 41 được mô tả trong mục 7 ở phần 4.

42. Phương pháp xử lý video, bao gồm: thực hiện biến đổi giữa khối video hiện tại và biểu diễn dòng bit của khối video hiện tại nhờ sử dụng IBC trong đó độ chính xác thứ nhất được sử dụng cho các phép tính dự báo trong khi biến đổi thấp hơn độ chính xác thứ hai được sử dụng cho các phép tính tái tạo trong khi biến đổi.

43. Phương pháp theo mệnh đề 42, trong đó các phép tính dự báo bao gồm xác định giá trị mẫu dự báo từ giá trị mẫu được tái tạo nhờ sử dụng $\text{clip}\{p + [1 \ll (b-1)]\} \gg b, 0, (1 \ll \text{bitdepth}) - 1\} \ll b$, trong đó p là giá trị mẫu được tái tạo, b là giá trị dịch bit định trước, và bitdepth là độ chính xác mẫu dự báo.

Các phương án thực hiện và các ví dụ bổ sung theo các mệnh đề 42 và 43 được mô tả trong mục 28 đến 31 và 34 trong phần 4.

44. Phương pháp xử lý video, bao gồm: thực hiện biến đổi giữa khối video hiện tại và biểu diễn dòng bit của khối video hiện tại nhờ sử dụng IBC trong đó khu vực tham chiếu có kích thước $nM \times nM$ được sử dụng cho kích thước CTU $M \times M$, trong đó n và M là các số nguyên và trong đó khối video hiện tại được đặt

ở CTU, và trong đó khu vực tham chiếu là CTU nxn khả dụng gần nhất trong hàng CTU tương ứng với khối video hiện tại.

Các phương án thực hiện và các ví dụ bổ sung theo mệnh đề 4 được mô tả trong mục 35 trong phần 4.

45. Phương pháp xử lý video, bao gồm: thực hiện biến đổi giữa khối video hiện tại và biểu diễn dòng bit của khối video hiện tại nhờ sử dụng IBC trong đó khu vực tham chiếu có kích thước $nM \times nM$ được sử dụng cho kích thước CTU khác ngoài $M \times M$, trong đó n và M là các số nguyên và trong đó khối video hiện tại được định vị trong CTU, và trong đó khu vực tham chiếu là CTU $nxn-1$ khả dụng gần nhất trong hàng CTU tương ứng với khối video hiện tại.

Các phương án thực hiện và các ví dụ bổ sung theo mệnh đề 4 được mô tả trong mục 36 trong phần 4. Fig.8 và 9 thể hiện các phương án thực hiện ví dụ bổ sung.

46. Phương pháp theo điểm 3, trong đó $M=mW$ và $N=H$, trong đó W và H là chiều rộng và chiều cao của CTU (CTU) của khối video hiện tại, và m là số nguyên dương.

47. Phương pháp theo điểm 3, trong đó $M=W$ và $N=nH$, trong đó W và H là chiều rộng và chiều cao của CTU (CTU), và n là số nguyên dương.

48. Phương pháp theo điểm 3, trong đó $M=mW$ và $N=nH$, trong đó W và H là chiều rộng và chiều cao của CTU (CTU), m và n là các số nguyên dương.

49. Phương pháp theo điểm bất kỳ trong số các điểm từ 46 đến 48, trong đó n và m phụ thuộc kích thước của CTU.

50. Phương pháp xử lý video, bao gồm: xác định, đối với biến đổi giữa khối video hiện tại của video và biểu diễn dòng bit của khối video hiện tại, tính hợp lệ của vector khối tương ứng với khối video hiện tại của thành phần c của video nhờ sử dụng thành phần X của video, trong đó thành phần X khác với thành phần độ sáng của video; và thực hiện việc biến đổi nhờ sử dụng vector khối khi xác định rằng vector khối là hợp lệ cho khối video hiện tại. Ở đây, vector khối, được ký hiệu là (BV_x, BV_y) bằng $(x-x_0, y-y_0)$, trong đó (x_0, y_0) tương ứng với vị trí trên bên trái của CTU của khối video hiện tại.

51. Phương pháp theo mệnh đề 50, trong đó thành phần c tương ứng với thành phần độ sáng của video.

52. Phương pháp theo mệnh đề 50, trong đó khối video hiện tại là khối sắc độ và video ở định dạng 4:4:4.

53. Phương pháp theo mệnh đề 50, trong đó video ở định dạng 4:2:0, và trong đó khối video hiện tại là khối sắc độ bắt đầu ở vị trí (x, y) , và trong đó việc xác định bao gồm xác định vector khối là không hợp lệ cho trường hợp trong đó $\text{isRec}(c, ((x+BVx) \gg 5 < 5) + 64 - (((y+BVy) \gg 5) \& 1) * 32 + (x \% 32), ((y+BVy) \gg 5 < 5) + (y \% 32))$ là true.

54. Phương pháp theo mệnh đề 50, trong đó video ở định dạng 4:2:0, và trong đó khối video hiện tại là khối sắc độ bắt đầu ở vị trí (x, y) , và trong đó việc xác định bao gồm xác định vector khối là không hợp lệ cho trường hợp trong đó nếu $\text{isRec}(c, x+BVx+\text{Chroma_CTU_size}, y)$ là true.

55. Phương pháp xử lý video, bao gồm: xác định, có chọn lọc đối với biến đổi giữa khối video hiện tại của VPDU hiện tại của vùng video và biểu diễn dòng bit của khối video hiện tại, để sử dụng K1 VPDU được xử lý trước đó từ hàng thứ nhất của vùng video và K2 VPDU được xử lý trước đó từ hàng thứ hai của vùng video; và thực hiện việc biến đổi, trong đó việc biến đổi loại trừ việc sử dụng phần còn lại của VPDU hiện tại.

56. Phương pháp theo mệnh đề 55, trong đó $K1 = 1$ và $K2 = 2$.

57. Phương pháp theo mệnh đề 55 hoặc 56, trong đó khối video hiện tại được xử lý có chọn lọc dựa trên kích thước của vùng video hoặc kích thước của VPDU hiện tại.

58. Phương pháp xử lý video, bao gồm: thực hiện kiểm tra tính hợp lệ của vector khối đối với biến đổi giữa khối video hiện tại và biểu diễn dòng bit của khối video hiện tại, trong đó vector khối được sử dụng cho IBC; và sử dụng kết quả của kiểm tra tính hợp lệ để sử dụng có chọn lọc vector khối trong khi biến đổi.

59. Phương pháp theo mệnh đề 58, trong đó bộ đệm IBC được sử dụng trong khi biến đổi, trong đó chiều rộng và chiều cao của bộ đệm IBC làm Wbuf và Hbuf, kích thước của khối video hiện tại là $W \times H$ và trong đó vector khối được

biểu diễn dưới dạng (BV_x , BV_y), và trong đó khối video hiện tại nằm trong ảnh hiện tại có các kích thước W_{pic} và H_{pic} và CTU có W_{ctu} và H_{ctu} làm chiều rộng và chiều cao, và trong đó kiểm tra tính hợp lệ sử dụng quy tắc định trước.

60. Phương pháp theo mệnh đề 58 hoặc 59, trong đó khối video hiện tại là khối độ sáng, khối sắc độ, CU, TU, khối 4×4 , khối 2×2 , hoặc khối phụ của khối cha mẹ bắt đầu từ các tọa độ pixel (X , Y).

61. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 58 đến 60, trong đó kiểm tra tính hợp lệ xem xét vector khối nằm ngoài biên của ảnh hiện tại là hợp lệ.

62. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 58 đến 60, trong đó kiểm tra tính hợp lệ xem xét vector khối nằm ngoài biên của CTU là hợp lệ.

Các mục từ 23 đến 30 trong phần trước nêu các ví dụ và các biến thể bổ sung của các mệnh đề nêu trên từ 58 đến 62.

63. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 62, trong đó việc biến đổi bao gồm tạo biểu diễn dòng bit từ khối video hiện tại.

64. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 62, trong đó việc biến đổi bao gồm tạo các giá trị pixel của khối video hiện tại từ biểu diễn dòng bit.

65. Bộ mã hóa video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu theo một hoặc nhiều mệnh đề từ 1 đến 62.

66. Bộ giải mã video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu theo một hoặc nhiều mệnh đề từ 1 đến 62.

67. Vật ghi máy tính đọc được lưu trữ mã trên đó, mã thực thi các lệnh bộ xử lý thực thi được để thực hiện phương pháp được nêu theo một hoặc nhiều mệnh đề từ 1 đến 62.

Fig.7 là sơ đồ khối của nền tảng phần cứng của thiết bị xử lý video/ảnh 700. Thiết bị 700 có thể được sử dụng để thực hiện một hoặc nhiều phương pháp được mô tả ở đây. Thiết bị 700 có thể được triển khai trong điện thoại thông minh, máy tính bảng, máy tính, bộ thu IoT, và v.v. Thiết bị 700 có thể bao gồm một hoặc nhiều bộ xử lý 702, một hoặc nhiều bộ nhớ 704 và phần cứng xử lý video 706. (Các) bộ xử lý 702 có thể được tạo cấu hình để thực hiện một hoặc nhiều

phương pháp (bao gồm, mà không bị giới hạn ở, phương pháp 600) được nêu theo sáng chế. (Các) bộ nhớ 704 có thể được sử dụng để lưu dữ liệu và mã được sử dụng để thực hiện các phương pháp và các kỹ thuật được mô tả ở đây. Phần cứng xử lý video 706 có thể được sử dụng để thực hiện, trong hệ mạch phần cứng, một số kỹ thuật được nêu theo sáng chế.

Biểu diễn dòng bit tương ứng với khối video hiện tại không cần là tập hợp bit liên tục và có thể được phân tán giữa các tiêu đề, các tập hợp tham số, và các gói lớp trừu tượng mạng (NAL).

Phần A: Phương án thực hiện lấy làm ví dụ bổ sung khác

Phần A nêu phương án thực hiện lấy làm ví dụ trong đó phiên bản hiện tại của chuẩn VVC có thể được chỉnh sửa để thực hiện một số kỹ thuật được nêu theo sáng chế.

Phần này phân tích một số ví dụ thiết kế bộ đệm IBC tham chiếu hiện tại và nêu thiết kế khác để giải quyết các vấn đề này. Bộ đệm tham chiếu IBC độc lập được nêu thay cho việc kết hợp với bộ nhớ giải mã. So với phần neo hiện tại, sơ đồ được nêu thể hiện -0,99%/-0,71%/-0,79% tốc độ BD độ sáng AI/RA/LD-B cho lớp F và -2,57%/-1,81%/-1,36% cho 4:2:0 TGM, với giảm bộ nhớ 6,7%; hoặc -1,31%/-1,01%/-0,81% cho Lớp F và -3,23%/-2,33%/-1,71% cho 4:2:0 TGM có tăng bộ nhớ 6,7%.

A1. Giới thiệu

Chế độ mã hóa IBC (hoặc tham chiếu ảnh hiện tại, tức là, trước đó CPR) được sử dụng. Được triển khai rằng IBC các mẫu tham chiếu cần được lưu trữ trong bộ nhớ trên chip và do vậy khu vực tham chiếu bị giới hạn của một CTU được định nghĩa. Để giới hạn bộ nhớ trên chip mở rộng cho bộ đệm, thiết kế hiện tại sử dụng lại bộ nhớ 64x64 để giải mã VPDU hiện tại sao cho chỉ bộ nhớ của ba khối 64x64 bổ sung cần hỗ trợ IBC. Khi kích thước CTU bằng 128x128, hiện tại khu vực tham chiếu được thể hiện trên Fig.2.

- Các điều kiện sau sẽ là true:

$$(yCb + (mvL[1] \gg 4)) \gg CtbLog2SizeY = yCb \gg \quad (8-972)$$

CtbLog2SizeY

$$(yCb + (mvL[1] \gg 4) + cbHeight - 1) \gg CtbLog2SizeY = yCb \quad (8-973)$$

>> CtbLog2SizeY

$$(xCb + (mvL[0] \gg 4)) \gg CtbLog2SizeY \geq (xCb \gg \quad (8-974)$$

CtbLog2SizeY) - 1

$$(xCb + (mvL[0] \gg 4) + cbWidth - 1) \gg CtbLog2SizeY \leq (xCb \quad (8-975)$$

>> CtbLog2SizeY)

[Ed. (SL): các điều kiện (8-218) và (8-216) có thể đã được kiểm tra bởi

6.4.X.]

– Khi $(xCb + (mvL[0] \gg 4)) \gg CtbLog2SizeY$ bằng $(xCb \gg CtbLog2SizeY) - 1$, quá trình dẫn xuất cho tính khả dụng khối như được xác định trong mệnh đề 6.4.X [Ed. (BB): Quá trình kiểm tra tính khả dụng các khối lân cận tbd] được gọi với vị trí độ sáng hiện tại $(xCurr, yCurr)$ được thiết lập bằng (xCb, yCb) và vị trí độ sáng lân cận $((xCb + (mvL[0] \gg 4) + CtbSizeY) \gg (CtbLog2SizeY - 1)) \ll (CtbLog2SizeY - 1), ((yCb + (mvL[1] \gg 4)) \gg (CtbLog2SizeY - 1)) \ll (CtbLog2SizeY - 1))$ làm các đầu vào, và đầu ra cần bằng FALSE.

Do vậy, tổng kích thước tham số là CTU.

A2. Các vấn đề tiềm năng về thiết kế hiện tại

Thiết kế hiện tại giả sử sử dụng lại bộ nhớ 64x64 để giải mã VPDU hiện tại và tham chiếu IBC được căn chỉnh để sử dụng lại bộ nhớ VPDU theo đó. Thiết kế này gộp bộ nhớ giải mã VPDU với bộ đệm IBC. Có thể có vài vấn đề sau:

1. Xử lý kích thước CTU nhỏ hơn có thể là vấn đề. Giả sử rằng kích thước CTU bằng 32x32, nó không rõ ràng liệu bộ nhớ 64x64 hiện tại để giải mã VPDU hiện tại có thể hỗ trợ hiệu quả sử dụng lại bộ nhớ mức 32x32 trong các kiến trúc khác nhau.

2. Khu vực tham chiếu thay đổi quan trọng. Do đó, quá nhiều giới hạn tương thích dòng bit được đưa vào. Điều này đặt gánh nặng thêm cho bộ mã hóa để khai triển hiệu quả khu vực tham chiếu và tránh tạo các dòng bit hợp pháp. Nó cũng tăng khả năng để có các BV không hợp lệ trong các mô đun khác nhau,

chẳng hạn, danh sách hợp nhất. Xử lý các BV không hợp lệ đó có thể đưa giới hạn tương thích bổ sung hoặc các logic bổ sung. Không chỉ đưa các gánh nặng cho bộ mã hóa hoặc bộ giải mã, cũng có thể tạo sự phân kỳ giữa mã hóa BV và mã hóa MV.

3. Thiết kế này không chia tỷ lệ tốt. Do VPDU giải mã được kết hợp với bộ đệm IBC, không dễ tăng hoặc giảm khu vực tham chiếu so với thiết kế CTU 128x128 của pixel hiện tại. Có thể giới hạn độ linh hoạt triển khai hiệu quả mã hóa tốt hơn với việc cân bằng bộ nhớ trên chip trong phát triển sau này, chẳng hạn, tiêu sử cao hơn hoặc thấp hơn.

4. Chiều sâu bit của bộ đệm tham chiếu IBC được liên kết với bộ đệm giải mã. Mặc dù các nội dung màn hình thường có chiều sâu bit thấp hơn với chiều sâu bit giải mã trong, bộ đệm vẫn cần tiêu tốn bộ nhớ để lưu trữ các bit thường biểu diễn nhiều các nhiễu làm tròn hoặc lượng tử hóa. Vấn đề này trở nên nghiêm trọng khi xem xét các cấu hình chiều sâu bit giải mã cao hơn.

A3. Thiết kế bộ đệm IBC rõ ràng

Để giải quyết các vấn đề được nêu trong phần phụ nêu trên, đề xuất có bộ đệm IBC dành riêng, không được kết hợp với bộ nhớ giải mã.

Đối với CTU 128x128, bộ đệm được định nghĩa là 128x128 có các mẫu 8-bit, khi CU (x, y) có kích thước wxh đã được giải mã, việc tái tạo trước khi lọc vòng được biến đổi thành 8 bit và được ghi vào diện tích khối wxh bắt đầu từ vị trí (x%128, y%128). Ở đây, toán tử modulo % luôn trả về số dương, tức là, for $x < 0$, $x \% L \triangleq -(-x \% L)$, chẳng hạn $-3 \% 128 = 125$.

Giả sử pixel (x,y) được tạo mã trong chế độ IBC với BV=(BVx, BVy), chính là mẫu dự báo trong bộ đệm tham chiếu IBC đặt ở ((x+BVx)%128, (y+BVy)%128) và giá trị pixel sẽ được biến đổi thành 10-bit trước khi dự báo.

Khi bộ đệm được xem xét là (W, H), sau khi giải mã CTU hoặc CU bắt đầu từ (x, y), các pixel được tái tạo trước khi lọc vòng sẽ được lưu trữ trong bộ đệm bắt đầu từ (x%W, y%H). Do vậy, sau khi giải mã CTU, bộ đệm tham chiếu IBC tương ứng sẽ được cập nhật do đó. Thiết lập này có thể xảy ra khi kích thước CTU không bằng 128x128. Chẳng hạn, đối với CTU 64x64, có kích thước bộ

đệm hiện tại, có thể được xem xét là bộ đệm 256x64. Đối với CTU 64x64, Fig.2 thể hiện trạng thái bộ đệm.

Fig.12 là hình vẽ trạng thái bộ đệm tham chiếu IBC, trong đó khối ký hiệu CTU 64x64.

Theo thiết kế hiện tại, do bộ đệm IBC khác với bộ nhớ giải mã VPDU, tất cả bộ đệm tham chiếu IBC có thể được sử dụng làm tham chiếu.

Khi chiều sâu bit của bộ đệm IBC bằng 8-bit, so với thiết kế hiện tại cần ba bộ đệm 64x64 10 bit bổ sung, tăng bộ nhớ trên chip bằng $(8*4)/(10*3)-100\%=6,7\%$.

Cần giảm thêm chiều sâu bit. Yêu cầu bộ nhớ có thể được giảm tiếp. Chẳng hạn, đối với bộ đệm 7-bit, việc tiết kiệm bộ nhớ trên chip bằng $100\%-(7*4)/(10*3)=6,7\%$.

Với thiết kế này, chỉ giới hạn tương thích dòng bit chính là khối tham chiếu nằm trong khu vực được tái tạo trong hàng CTU hiện tại của phiên hiện tại.

Khi khởi tạo đến 512 được phép ở bắt đầu của mỗi hàng CTU, tất cả giới hạn tương thích dòng bit có thể được loại bỏ.

A4. Các kết quả thực nghiệm

Theo một số phương án thực hiện, các phương pháp được bộc lộ có thể được triển khai nhờ sử dụng phần mềm VTM-4.0.

Để triển khai bộ đệm 10 bit và CTC, bộ giải mã hoàn toàn tương thích với bộ mã hóa VTM4.0 hiện tại, nghĩa là bộ giải mã được đề xuất có thể chính xác giải mã các dòng bit VTM-4.0 CTC.

Đối với triển khai bộ đệm 7 bit, các kết quả được thể hiện trên Bảng I.

Đối với triển khai bộ đệm 8 bit, các kết quả được thể hiện trên Bảng II.

Bảng I. Hiệu năng với bộ đệm 7 bit. Phần neo là VTM-4.0 có IBC bật cho tất cả các chuỗi.

	Toàn bộ bên trong				
	Trên VTM-4.0 có bật IBC				
	Y	U	V	EncT	DecT
Lớp A1	-0,01%	-0,09%	-0,10%	132%	101%
Lớp A2	0,05%	0,00%	0,06%	135%	100%
Lớp B	0,00%	-0,02%	0,01%	135%	100%
Lớp C	-0,02%	0,01%	0,03%	130%	98%
Lớp E	-0,13%	-0,16%	-0,04%	135%	99%
Tổng	-0,02%	-0,05%	0,00%	133%	100%
Lớp D	0,04%	0,04%	0,12%	127%	107%
Lớp F	-0,99%	-1,14%	-1,18%	115%	99%
4:2:0 TGM	-2,57%	-2,73%	-2,67%	104%	102%

	Truy nhập ngẫu nhiên				
	Trên VTM-4.0 có bật IBC				
	Y	U	V	EncT	DecT
Lớp A1	0,02%	-0,01%	0,01%	109%	100%
Lớp A2	0,00%	-0,04%	0,03%	111%	100%
Lớp B	-0,01%	-0,10%	-0,22%	113%	101%
Lớp C	-0,01%	0,17%	0,12%	115%	100%
Lớp E					
Tổng	0,00%	0,00%	-0,04%	112%	100%
Lớp D	0,05%	0,16%	0,20%	117%	101%
Lớp F	-0,71%	-0,77%	-0,77%	109%	99%
4:2:0 TGM	-1,81%	-1,65%	-1,64%	107%	101%

	Độ trễ thấp B				
	Trên VTM-4.0 có bật IBC				
	Y	U	V	EncT	DecT
Lớp A1					
Lớp A2					
Lớp B	0,01%	0,36%	0,30%	114%	95%
Lớp C	-0,01%	-0,12%	-0,10%	120%	98%
Lớp E	0,10%	0,20%	0,18%	107%	99%
Tổng	0,03%	0,16%	0,13%	114%	97%
Lớp D	-0,01%	1,07%	0,18%	123%	104%
Lớp F	-0,79%	-0,89%	-1,01%	110%	100%
4:2:0 TGM	-1,36%	-1,30%	-1,26%	109%	102%

Bảng II. Hiệu năng có bộ đệm 8 bit. Phân neo là VTM-4.0 có bật IBC trên toàn bộ chuỗi.

	Toàn bộ bên trong				
	Trên VTM-4.0 có bật IBC				
	Y	U	V	EncT	DecT
Lớp A1	-0,01%	0,02%	-0,10%	129%	102%
Lớp A2	0,02%	-0,06%	-0,02%	134%	102%
Lớp B	-0,04%	-0,02%	-0,07%	135%	101%
Lớp C	-0,03%	0,04%	0,00%	130%	98%
Lớp E	-0,16%	-0,14%	-0,08%	134%	100%
Tổng	-0,04%	-0,03%	-0,05%	133%	100%

Lớp D	0,00%	0,04%	0,02%	126%	101%
Lớp F	-1,31%	-1,27%	-1,29%	114%	98%
4:2:0 TGM	-3,23%	-3,27%	-3,24%	101%	100%

	Truy nhập ngẫu nhiên				
	Trên VTM-4.0 có bật IBC				
	Y	U	V	EncT	DecT
Lớp A1	-0,01%	-0,08%	0,04%	107%	99%
Lớp A2	-0,03%	-0,16%	0,06%	110%	99%
Lớp B	-0,01%	-0,14%	-0,22%	111%	99%
Lớp C	-0,01%	0,15%	0,09%	115%	100%
Lớp E					
Tổng	-0,01%	-0,05%	-0,03%	111%	99%
Lớp D	0,01%	0,19%	0,22%	116%	101%
Lớp F	-1,01%	-0,99%	-1,01%	108%	99%
4:2:0 TGM	-2,33%	-2,14%	-2,19%	105%	100%

	Độ trễ thấp B				
	Trên VTM-4.0 có bật IBC				
	Y	U	V	EncT	DecT
Lớp A1					
Lớp A2					
Lớp B	0,00%	0,04%	-0,14%	113%	#NUM!
Lớp C	-0,05%	-0,28%	-0,15%	119%	98%
Lớp E	0,04%	-0,16%	0,43%	107%	#NUM!
Tổng	0,00%	-0,11%	0,00%	113%	#NUM!
Lớp D	-0,07%	1,14%	0,13%	122%	99%
Lớp F	-0,81%	-0,92%	-0,96%	111%	99%
4:2:0 TGM	-1,71%	-1,67%	-1,71%	106%	95%

Fig.17 là sơ đồ khối thể hiện ví dụ hệ thống xử lý video 1700 trong đó các kỹ thuật khác nhau được bộc lộ ở đây có thể được triển khai. Các cách triển khai khác nhau có thể bao gồm một số hoặc tất cả các thành phần của hệ thống 1700. Hệ thống 1700 có thể bao gồm đầu vào 1702 để nhận nội dung video. Nội dung video có thể được nhận ở định dạng thô hoặc không được nén, chẳng hạn, các giá trị pixel đã thành phần 8 bit hoặc 10 bit, hoặc có thể ở định dạng được nén hoặc được mã hóa. Đầu vào 1702 có thể là giao diện mạng, giao diện buýt ngoại vi, hoặc giao diện lưu trữ. Các ví dụ của giao diện mạng bao gồm các giao diện hữu tuyến chẳng hạn Ethernet, mạng quang thụ động (PON), v.v và các giao diện không dây chẳng hạn Wi-Fi hoặc các giao diện tế bào.

Hệ thống 1700 có thể bao gồm thành phần mã hóa 1704 có thể thực hiện các phương pháp tạo mã hoặc mã hóa khác nhau được nêu theo sáng chế. Thành phần mã hóa 1704 có thể giảm tốc độ bit trung bình của video từ đầu vào 1702

đến đầu ra của thành phần mã hóa 1704 để tạo biểu diễn được tạo mã của video. Do vậy, các kỹ thuật tạo mã đôi lúc được gọi là kỹ thuật nén video hoặc các kỹ thuật chuyển mã video. Đầu ra của thành phần mã hóa 1704 có thể được lưu trữ, hoặc được truyền qua kết nối truyền thông, như được biểu diễn bằng thành phần 1706. Biểu diễn dòng bit được lưu trữ hoặc được truyền thông (hoặc được tạo mã) của video được nhận ở đầu vào 1702 có thể được sử dụng bởi thành phần 1708 để tạo các giá trị pixel hoặc video có thể hiển thị được gửi đến giao diện hiển thị 1710. Quá trình của tạo video người dùng có thể xem từ biểu diễn dòng bit đôi lúc được gọi là giải nén video. Ngoài ra, trong khi các hoạt động xử lý video cụ thể được gọi là các hoạt động hoặc các công cụ thể “tạo mã”, cần hiểu rằng các công cụ mã hoặc các hoạt động được sử dụng ở bộ mã hóa và các công cụ giải mã tương ứng hoặc các hoạt động đảo ngược các kết quả mã hóa sẽ được thực hiện bằng bộ giải mã.

Các ví dụ về giao diện buýt ngoại vi hoặc giao diện hiển thị có thể bao gồm buýt nối tiếp đa năng (USB) hoặc giao diện đa phương tiện độ nét cao (HDMI) hoặc Displayport, và v.v. Các ví dụ về các giao diện lưu trữ bao gồm SATA (serial advanced technology attachment), PCI, giao diện IDE, và tương tự. Các kỹ thuật được nêu theo sáng chế có thể được triển khai trong các thiết bị điện tử khác nhau chẳng hạn điện thoại di động, máy tính xách tay, điện thoại thông minh hoặc các thiết bị khác có thể thực hiện xử lý dữ liệu số và/hoặc hiển thị video.

Fig.18 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 1 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 1802, quy trình này xác định kích thước của bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC. Ở bước 1804, quy trình này thực hiện biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Fig.19 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 4 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 1902, quy trình này xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu được tái tạo để dự báo trong chế độ IBC, trong đó bộ đệm được sử dụng để lưu trữ các mẫu được tái tạo trước bước lọc vòng. Ở bước 1904, quy trình này thực hiện việc biến đổi mà sử dụng các mẫu được tái tạo được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Fig.20 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 5 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 2002, quy trình này xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu được tái tạo để dự báo trong chế độ IBC, trong đó bộ đệm được sử dụng để lưu trữ các mẫu được tái tạo sau bước lọc vòng. Ở bước 2004, quy trình này thực hiện việc biến đổi mà sử dụng các mẫu được tái tạo được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Fig.21 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 6 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 2102, quy trình này xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu được tái tạo để dự báo trong chế độ IBC, trong đó bộ đệm được sử dụng để lưu trữ các mẫu được tái tạo cả trước bước lọc vòng và sau bước lọc vòng. Ở bước 2104, quy trình này thực hiện việc biến đổi mà sử dụng các mẫu được tái tạo được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển

động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Fig.22 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 7 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 2202, quy trình này sử dụng bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó chiều sâu bit thứ nhất của bộ đệm khác với chiều sâu bit thứ hai được sử dụng để biểu diễn dữ liệu media hình ảnh trong biểu diễn dòng bit. Ở bước 2204, quy trình này thực hiện biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Fig.23 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 8 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 2302, quy trình này khởi tạo bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó bộ đệm được khởi tạo với giá trị thứ nhất. Ở bước 2304, quy trình này thực hiện biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Fig.24 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 9 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 2402, quy trình này khởi tạo bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó, dựa trên tính khả dụng của một hoặc nhiều khối video trong dữ liệu media hình ảnh, bộ đệm được khởi tạo với các giá trị pixel của một hoặc nhiều khối video trong dữ liệu media hình ảnh. Ở bước 2404, quy trình này thực hiện biến đổi giữa khối video hiện tại mà không thuộc một hoặc nhiều khối video của dữ liệu media hình ảnh và biểu

diễn dòng bit của khối video hiện tại, nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Fig.25 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với Phương án thực hiện 14c lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 2502, quy trình này xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC. Ở bước 2504, quy trình này thực hiện việc biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu. Ở bước 2506, quy trình này tính toán tham chiếu tương ứng trong bộ đệm dựa trên vị trí tham chiếu ($P \bmod M$, $Q \bmod N$) trong đó “mod” là phép toán modulo và M và N là các số nguyên biểu diễn các chiều x và y của bộ đệm, trong đó vị trí tham chiếu (P , Q) được xác định nhờ sử dụng vector khối (BV_x , BV_y) và vị trí (x_0 , y_0), đối với pixel được đặt trong không gian ở vị trí (x_0 , y_0) và có vector khối (BV_x , BV_y) được bao gồm trong thông tin chuyển động.

Fig.26 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 14a và 14b lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 2602, quy trình này xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC. Ở bước 2604, quy trình này thực hiện việc biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu. Ở bước 2606, quy trình này tính toán tham chiếu tương ứng trong bộ đệm dựa trên vị trí tham chiếu (P , Q), trong đó vị trí

tham chiếu (P, Q) được xác định nhờ sử dụng vector khối (BVx, BVy) và vị trí (x0, y0), đối với pixel được đặt trong không gian ở vị trí (x0, y0) và có vector khối (BVx, BVy) được bao gồm trong thông tin chuyển động.

Fig.27 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 10 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 2702, quy trình này xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó các vị trí pixel trong bộ đệm được giải quyết nhờ sử dụng các số x và y. Ở bước 2704, quy trình này thực hiện, dựa trên các số x và y, việc biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

Fig.28 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 15 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 2802, quy trình này xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu. Ở bước 2804, quy trình này tính toán tham chiếu tương ứng trong bộ đệm ở vị trí tham chiếu (P, Q), trong đó vị trí tham chiếu (P, Q) được xác định nhờ sử dụng vector khối (BVx, BVy) và vị trí (x0, y0), đối với pixel được đặt trong không gian ở vị trí (x0, y0) của khối video hiện tại và có vector khối (BVx, BVy). Ở bước 2806, quy trình này tính toán lại vị trí tham chiếu nhờ sử dụng mẫu trong bộ đệm, khi xác định rằng vị trí tham chiếu (P, Q) nằm ngoài bộ đệm.

Fig.29 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 16 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 2902, quy trình này xác định, đối với biến đổi

giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu. Ở bước 2904, quy trình này tính toán tham chiếu tương ứng trong bộ đệm ở vị trí tham chiếu (P, Q), trong đó vị trí tham chiếu (P, Q) được xác định nhờ sử dụng vector khối (BVx, BVy) và vị trí (x0, y0), đối với pixel được đặt trong không gian ở vị trí (x0, y0) của khối video hiện tại so với vị trí trên bên trái của CTU bao gồm khối video hiện tại và có vector khối (BVx, BVy). Ở bước 2906, quy trình này giới hạn ít nhất một phần vị trí tham chiếu để nằm trong khoảng định trước, khi xác định rằng vị trí tham chiếu (P, Q) nằm ngoài bộ đệm.

Fig.30 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 17 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 3002, quy trình này xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu. Ở bước 3004, quy trình này tính toán tham chiếu tương ứng trong bộ đệm ở vị trí tham chiếu (P, Q), trong đó vị trí tham chiếu (P, Q) được xác định nhờ sử dụng vector khối (BVx, BVy) và vị trí (x0, y0), đối với pixel được đặt trong không gian ở vị trí (x0, y0) của khối video hiện tại so với vị trí trên bên trái của CTU bao gồm khối video hiện tại và có vector khối (BVx, BVy). Ở bước 1706, quy trình này đệm vector khối (BVx, BVy) theo vector khối của giá trị mẫu bên trong bộ đệm, khi xác định rằng vector khối (BVx, BVy) nằm ngoài bộ đệm.

Fig.31 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 30 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 3102, quy trình này thiết lập lại, trong khi biến đổi giữa video và biểu diễn dòng bit của video, bộ đệm mà lưu trữ các mẫu tham

chiều để dự báo trong chế độ IBC ở biên video. Ở bước 3104, quy trình này thực hiện việc biến đổi nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi khối video của video được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video mà không đề cập đến ảnh tham chiếu.

Fig.32 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 34 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 3202, quy trình này thực hiện biến đổi giữa khối video hiện tại và biểu diễn dòng bit của khối video hiện tại. Ở bước 3204, quy trình này cập nhật bộ đệm được sử dụng để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó bộ đệm được sử dụng đối với biến đổi giữa khối video tiếp theo và biểu diễn dòng bit của khối video tiếp theo, trong đó việc biến đổi giữa khối video tiếp theo và biểu diễn dòng bit của khối video tiếp theo được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video tiếp theo mà không đề cập đến ảnh tham chiếu.

Fig.33 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện lấy làm ví dụ 39 trong mục 4 của sáng chế. Ở bước 3302, quy trình này xác định, đối với biến đổi giữa khối video hiện tại và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà được sử dụng để lưu trữ các mẫu được tái tạo để dự báo trong chế độ IBC, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu. Ở bước 3304, quy trình này áp dụng hoạt động tiền xử lý cho các mẫu được tái tạo được lưu trữ trong bộ đệm, đáp lại xác định rằng các mẫu được tái tạo được lưu trữ trong bộ đệm cần được sử dụng để dự báo các giá trị mẫu trong khi hội thoại.

Fig.34 là lưu đồ của phương pháp lấy làm ví dụ xử lý dữ liệu hình ảnh. Các bước của lưu đồ này được nêu cùng với phương án thực hiện 42 lấy làm ví dụ trong mục 4 của sáng chế. Ở bước 3402, quy trình này xác định, có chọn lọc đối với biến đổi giữa khối video hiện tại của VPDU hiện tại của vùng video và biểu

diễn dòng bit của khối video hiện tại, liệu có sử dụng K1 VPDU được xử lý trước đó từ hàng được đánh số chẵn của vùng video và/hoặc K2 VPDU được xử lý trước đó từ hàng được đánh số lẻ của vùng video. Ở bước 3404, quy trình này thực hiện việc biến đổi, trong đó việc biến đổi loại trừ việc sử dụng phần còn lại của VPDU hiện tại, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video mà không đề cập đến ảnh tham chiếu.

Một số phương án thực hiện theo sáng chế hiện được trình bày ở định dạng dựa trên mệnh đề.

1. Phương pháp xử lý media hình ảnh, bao gồm các bước:

xác định kích thước của bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC; và

thực hiện biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu.

2. Phương pháp theo mệnh đề 1, trong đó kích thước của bộ đệm là hằng số định trước.

3. Phương pháp theo mệnh đề 1 hoặc 2, trong đó kích thước của bộ đệm là $M \times N$, trong đó M và N là các số nguyên.

4. Phương pháp theo mệnh đề 3, trong đó $M \times N$ bằng 64×64 hoặc 128×128 hoặc 64×128 hoặc 96×128 hoặc 128×96 .

5. Phương pháp theo mệnh đề 1, trong đó kích thước của bộ đệm hoặc một phần của nó bằng kích thước của CTU bao gồm khối video hiện tại hoặc một phần của nó.

6. Phương pháp theo mệnh đề 1, trong đó chiều cao của bộ đệm bằng chiều cao của CTU bao gồm khối video hiện tại.

7. Phương pháp theo mệnh đề 1, trong đó kích thước của bộ đệm tương ứng theo tỷ lệ với kích thước của CTU bao gồm khối video hiện tại.

8. Phương pháp theo mệnh đề 7, trong đó phép tương ứng theo tỷ lệ, được ký hiệu bởi các số nguyên m , n , dựa trên kích thước của CTU bao gồm khối video hiện tại sao cho khi kích thước CTU bằng 128×128 , $m=1$ và $n=1$, khi kích thước CTU bằng 64×64 , $m=4$ và $n=1$, khi kích thước CTU bằng 32×32 , $m=16$ và $n=1$, và khi kích thước CTU bằng 16×16 , $m=64$ và $n=1$, trong đó n bằng chiều cao của bộ đệm/chiều cao của CTU, và m bằng chiều rộng của bộ đệm/chiều rộng của CTU.

9. Phương pháp theo mệnh đề 1, trong đó tích số của chiều cao của bộ đệm và chiều rộng của bộ đệm là hằng số định trước.

10. Phương pháp theo mệnh đề 1, trong đó kích thước của bộ đệm tương ứng theo tỷ lệ với VPDU được sử dụng trong khi biến đổi.

11. Phương pháp theo mệnh đề 1, trong đó kích thước của bộ đệm bằng kích thước của VPDU được sử dụng trong khi biến đổi.

12. Phương pháp theo mệnh đề 1, trong đó kích thước của bộ đệm tương ứng với trường được bao gồm trong biểu diễn dòng bit.

13. Phương pháp theo mệnh đề 12, trong đó trường này được bao gồm trong biểu diễn dòng bit ở mức video hoặc chuỗi hoặc ảnh hoặc lát hoặc nhóm phiên.

14. Phương pháp theo mệnh đề 13, trong đó trường này được bao gồm trong mức VPS hoặc SPS hoặc PPS hoặc tiêu đề ảnh hoặc tiêu đề lát hoặc tiêu đề nhóm phiên.

15. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 14, trong đó bộ đệm lưu trữ các mẫu tham chiếu được liên kết với một thành phần màu hoặc nhiều thành phần màu.

16. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 15, trong đó nhiều hơn một bộ đệm được duy trì.

17. Phương pháp theo mệnh đề 16, trong đó bộ đệm thứ nhất và bộ đệm thứ hai được duy trì, bộ đệm thứ nhất có kích thước thứ nhất được sử dụng cho các mẫu tham chiếu được liên kết với các thành phần độ sáng và bộ đệm thứ hai có kích thước thứ hai được sử dụng cho các mẫu tham chiếu được liên kết với thành phần sắc độ, trong đó bộ đệm thứ nhất khác với bộ đệm thứ hai.

18. Phương pháp theo mệnh đề 17, trong đó kích thước thứ hai phụ thuộc vào kích thước thứ nhất.

19. Phương pháp theo mệnh đề 17, trong đó kích thước thứ hai độc lập với kích thước thứ nhất.

20. Phương pháp theo mệnh đề 17 hoặc 18, trong đó kích thước thứ hai phụ thuộc vào định dạng lấy mẫu phụ sắc độ của khối video hiện tại.

21. Phương pháp theo mệnh đề 18, trong đó kích thước thứ hai bằng kích thước thứ nhất.

22. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 21, trong đó các mẫu tham chiếu được lưu trữ theo định dạng RGB.

23. Phương pháp theo mệnh đề 22, trong đó kích thước của bộ đệm bằng 64x64 hoặc 128x128 hoặc 64x128 hoặc 128x64.

24. Phương pháp theo mệnh đề 1, trong đó kích thước của bộ đệm dựa trên ít nhất một phần kích thước của VPDU được liên kết với việc biến đổi và/hoặc kích thước của khối cây mã được liên kết với khối video hiện tại

25. Phương pháp theo mệnh đề 24, trong đó kích thước của bộ đệm bằng kích thước của khối cây mã.

26. Phương pháp theo mệnh đề 24, trong đó chiều cao của bộ đệm bằng chiều cao của CTU được liên kết với khối video hiện tại.

27. Phương pháp theo mệnh đề 24, trong đó kích thước của bộ đệm bao gồm chiều cao, và trong đó chiều cao bằng giá trị nhỏ nhất của (i) giá trị định trước và (ii) kích thước của khối cây mã được liên kết với khối video hiện tại.

28. Phương pháp theo mệnh đề 27, trong đó giá trị định trước bằng 64.

29. Phương pháp theo mệnh đề 24, trong đó chiều rộng của bộ đệm bằng $128 \cdot 128 / vSize$, trong đó $vSize$ ký hiệu kích thước của VPDU.

30. Phương pháp theo mệnh đề 24, trong đó chiều cao của bộ đệm bằng $\min(ctbSize, 64)$, trong đó $ctbSize$ ký hiệu kích thước của khối cây mã, và trong đó $\min(x, y)$ ký hiệu giá trị nhỏ nhất của x và y .

31. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 30, trong đó việc biến đổi bao gồm tạo biểu diễn dòng bit từ khối video hiện tại.

32. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 1 đến 30, trong đó việc biến đổi bao gồm tạo các giá trị pixel của khối video hiện tại từ biểu diễn dòng bit.

33. Bộ mã hóa video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu theo một hoặc nhiều mệnh đề từ 1 đến 30.

34. Bộ giải mã video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu theo một hoặc nhiều mệnh đề từ 1 đến 30.

35. Vật ghi máy tính đọc được lưu trữ mã trên đó, mã thực thi các lệnh bộ xử lý thực thi được để thực hiện phương pháp được nêu theo một hoặc nhiều mệnh đề từ 1 đến 30.

Theo sáng chế, cụm từ “xử lý video” có thể đề cập đến kỹ thuật mã hóa video, giải mã video, kỹ thuật nén video hoặc giải nén video. Chẳng hạn, các thuật toán nén video có thể được áp dụng trong khi biến đổi từ biểu diễn pixel của video đến biểu diễn dòng bit tương ứng hoặc ngược lại. Biểu diễn dòng bit của khối video hiện tại có thể, chẳng hạn, tương ứng với các bit cùng vị trí hoặc được lan rộng ở các vị trí khác trong dòng bit, như được định nghĩa bằng cú pháp này. Chẳng hạn, khối lớn có thể được mã hóa liên quan đến các giá trị dư sai số được biến đổi và mã hóa và cũng sử dụng các bit trong các tiêu đề và các trường khác trong dòng bit.

Từ phần nêu trên, cần hiểu rằng các phương án thực hiện cụ thể của giải pháp đang được bộc lộ đã được mô tả ở đây để minh họa, mà các chỉnh sửa khác nhau có thể được thực hiện mà không xa rời phạm vi của sáng chế. Do đó, giải pháp đang được bộc lộ không bị giới hạn ngoại trừ như bởi các điểm yêu cầu bảo hộ đi kèm.

Các triển khai của đối tượng và các hoạt động chức năng được mô tả theo sáng chế có thể được triển khai trong các hệ thống khác nhau, hệ mạch điện tử số, hoặc trong phần mềm máy tính, firmware, hoặc phần cứng, bao gồm các cấu trúc được nêu trong bản mô tả và các tương đương về mặt cấu trúc, hoặc trong các tổ hợp của một hoặc nhiều trong số chúng. Các triển khai của đối tượng được nêu trong bản mô tả có thể được triển khai dưới dạng một hoặc nhiều sản phẩm chương trình máy tính, tức là, một hoặc nhiều môđun của các lệnh chương trình

máy tính được mã hóa trên vật máy tính đọc được hữu hình và bất biến để thực thi bằng, hoặc để điều khiển hoạt động của, thiết bị xử lý dữ liệu. Vật ghi máy tính đọc được có thể là bộ lưu trữ máy đọc được, nền lưu trữ máy đọc được, bộ nhớ, hợp chất ảnh hưởng tín hiệu được truyền máy đọc được, hoặc tổ hợp của một hoặc nhiều trong số chúng. Cụm từ “khối xử lý dữ liệu” hoặc “thiết bị xử lý dữ liệu” bao gồm tất cả thiết bị, bộ phận, và máy để xử lý dữ liệu, bao gồm thông qua ví dụ, bộ xử lý lập trình được, máy tính, hoặc nhiều bộ xử lý hoặc máy tính. Thiết bị có thể bao gồm, bên cạnh phần cứng, mã tạo môi trường thực thi cho chương trình máy tính đang chạy, chẳng hạn, mã tạo firmware của bộ xử lý, xếp chồng giao thức, hệ thống quản lý cơ sở dữ liệu, hệ điều hành, hoặc tổ hợp của một hoặc nhiều trong số chúng.

Chương trình máy tính (cũng được biết đến là chương trình, phần mềm, ứng dụng phần mềm, tập lệnh, hoặc mã) có thể được viết ở dạng ngôn ngữ lập trình bất kỳ, bao gồm ngôn ngữ được thông dịch hoặc biên dịch, và có thể được khai triển ở dạng bất kỳ, bao gồm dưới dạng chương trình độc lập hoặc dưới dạng môđun, thành phần, trình con, hoặc khối khác thích hợp để sử dụng trong môi trường tính toán. Chương trình máy tính không nhất thiết tương ứng với tệp tin trong hệ thống tệp tin. Chương trình có thể được lưu trữ trong một phần tệp tin chứa các chương trình hoặc dữ liệu khác (chẳng hạn, một hoặc nhiều tập lệnh được lưu trữ trong tài liệu ngôn ngữ đánh dấu), trong một tệp tin dành cho chương trình đang chạy, hoặc trong nhiều tệp tin phối hợp (chẳng hạn, các tệp tin lưu trữ một hoặc nhiều môđun, các chương trình con, hoặc các phần mã). Chương trình máy tính có thể được khai triển để được thực thi trên một máy tính hoặc trên nhiều máy tính được đặt ở một địa điểm hoặc được phân tán giữa nhiều địa điểm và được liên kết bằng mạng truyền thông.

Các tiến trình và các luồng logic được nêu trong bản mô tả có thể được thực hiện bằng một hoặc nhiều bộ xử lý lập trình được thực thi một hoặc nhiều chương trình máy tính để thực hiện các chức năng bằng cách thao tác trên dữ liệu đầu vào và tạo đầu ra. Các tiến trình và các luồng logic cũng có thể được thực hiện bằng, và thiết bị cũng có thể được triển khai dưới dạng, hệ mạch logic chuyên

dụng, chẳng hạn, mảng công dụng trường lập trình được (FPGA) hoặc mạch tích hợp ứng dụng cụ thể (ASIC).

Các bộ xử lý thích hợp để thực thi chương trình máy tính bao gồm, thông qua ví dụ, cả bộ vi xử lý đa năng và chuyên dụng, và một hoặc nhiều bộ xử lý của loại máy tính số bất kỳ. Nói chung, bộ xử lý sẽ nhận các lệnh và dữ liệu từ ROM hoặc RAM hoặc cả hai. Các phần tử cơ bản của máy tính là bộ xử lý để thực hiện các lệnh và một hoặc nhiều bộ nhớ để lưu trữ các lệnh và dữ liệu. Nói chung, máy tính cũng sẽ bao gồm, hoặc được ghép nối vận hành để nhận dữ liệu từ hoặc truyền dữ liệu đến, hoặc cả hai, một hoặc nhiều bộ lưu trữ lớn để lưu trữ dữ liệu, chẳng hạn, đĩa từ, đĩa quang từ, hoặc đĩa quang. Tuy nhiên, máy tính không cần có các thiết bị này. Các phương tiện máy tính đọc được thích hợp để lưu trữ các lệnh chương trình máy tính và dữ liệu bao gồm tất cả các dạng bộ nhớ bất biến, các phương tiện và các bộ nhớ, bao gồm thông qua ví dụ các bộ nhớ bán dẫn, chẳng hạn, EPROM, EEPROM, và các bộ nhớ nhanh. Bộ xử lý và bộ nhớ có thể được bổ sung bằng, hoặc được đưa vào, hệ mạch logic chuyên dụng.

Bản mô tả, cùng với các hình vẽ, sẽ được xem chỉ để lấy ví dụ, trong đó lấy làm ví dụ nghĩa là ví dụ. Như được sử dụng ở đây, việc sử dụng “or” sẽ bao gồm “và/hoặc”, trừ khi có thông báo khác.

Trong khi sáng chế chứa nhiều đặc tả, chúng không được hiểu như là các giới hạn trên phạm vi của sáng chế bất kỳ hoặc phần có thể được bảo hộ, mà thay vào đó như là các phần mô tả của các dấu hiệu có thể dành riêng cho các phương án thực hiện cụ thể của các sáng chế cụ thể. Các dấu hiệu cụ thể được mô tả theo sáng chế theo các phương án thực hiện riêng rẽ cũng có thể được triển khai kết hợp theo một phương án thực hiện. Ngược lại, các dấu hiệu khác nhau được mô tả theo một phương án thực hiện cũng có thể được triển khai theo nhiều phương án thực hiện riêng rẽ hoặc kết hợp phụ thích hợp bất kỳ. Ngoài ra, mặc dù các dấu hiệu có thể được mô tả nêu trên như là hoạt động theo các tổ hợp cụ thể và thậm chí như được bảo hộ ban đầu, một hoặc nhiều dấu hiệu từ tổ hợp được nêu có thể trong một số trường hợp được thực hiện từ tổ hợp, và tổ hợp được nêu có thể dành cho tổ hợp phụ hoặc biến thể của tổ hợp phụ.

Tương tự, khi các hoạt động được mô tả theo các hình vẽ theo thứ tự cụ thể, điều này không nên được hiểu như là yêu cầu rằng các hoạt động này được thực hiện theo thứ tự cụ thể được thể hiện hoặc theo thứ tự lần lượt, hoặc tất cả các hoạt động được minh họa được thực hiện, để đạt được các kết quả mong muốn. Ngoài ra, việc tách riêng các thành phần hệ thống khác nhau theo các phương án thực hiện được mô tả theo sáng chế không nên được hiểu như là yêu cầu tách theo tất cả các phương án thực hiện.

Chỉ một vài cách triển khai và các ví dụ được mô tả và các cách triển khai khác, các phần tăng cường và các biến thể có thể được thực hiện dựa trên phần được mô tả và minh họa theo sáng chế.

YÊU CẦU BẢO HỘ

1. Phương pháp xử lý dữ liệu video bao gồm các bước:

xác định, đối với biến đổi giữa khối video hiện tại của ảnh hiện tại của video và dòng bit của video, rằng chế độ dự báo được áp dụng trên khối video hiện tại;

duy trì, đối với khối video hiện tại, bộ đệm bao gồm các mẫu tham chiếu được dẫn xuất từ các khối của các giá trị mẫu của cùng vùng video của khối video hiện tại trong ảnh hiện tại,

tạo, đối với khối video hiện tại, các mẫu dự báo dựa trên xác định; và

thực hiện biến đổi giữa khối video hiện tại và dòng bit,

trong đó trong chế độ dự báo, các mẫu tham chiếu được xác định bởi vector khối trong bộ đệm được sử dụng để dẫn xuất các mẫu dự báo của khối video hiện tại, và

trong đó khối tham chiếu cụ thể trong bộ đệm được đánh dấu là không khả dụng để dẫn xuất có các giá trị nằm ngoài khoảng giá trị pixel, và

trong đó các giá trị của các mẫu tham chiếu trong khối tham chiếu cụ thể được đánh dấu là không khả dụng được gán bằng giá trị định trước -1 chỉ báo tính không khả dụng của các mẫu.

2. Phương pháp theo điểm 1, trong đó vị trí và kích thước của khối tham chiếu cụ thể được đánh dấu là không khả dụng được xác định dựa trên một hoặc nhiều trong số: kích thước của khối video hiện tại, kích thước của khối ảo, hoặc kích thước của khối cây mã bao gồm khối video hiện tại.

3. Phương pháp theo điểm 2, trong đó vị trí của khối tham chiếu cụ thể được đánh dấu là không khả dụng được xác định dựa trên $(x0, y0)$,

trong đó $(x0 \% Vsize)$ bằng 0 và $(y0 \% Vsize)$ bằng 0,

trong đó % ký hiệu phép toán phần dư,

trong đó $Vsize$ ký hiệu kích thước của khối ảo và $Vsize = \min(ctbSize, 64)$, và trong đó $ctbSize$ ký hiệu chiều rộng hoặc chiều cao của khối cây mã.

4. Phương pháp theo điểm 2, trong đó kích thước của khối tham chiếu cụ thể được đánh dấu là không khả dụng được xác định dựa trên kích thước của khối ảo nếu kích thước của khối video hiện tại nhỏ hơn kích thước của khối ảo.

5. Phương pháp theo điểm 2, trong đó kích thước của khối tham chiếu cụ thể được đánh dấu là không khả dụng được xác định dựa trên kích thước của khối video hiện tại nếu kích thước của khối video hiện tại lớn hơn kích thước của khối ảo.
6. Phương pháp theo điểm 1, trong đó bộ đệm được thiết lập lại trước khi mã hóa ảnh hoặc lát.
7. Phương pháp theo điểm 1, trong đó ràng buộc tương thích dòng bit được thỏa mãn bởi dòng bit bao gồm việc các mẫu tham chiếu được xác định bởi vector khối trong bộ đệm không có giá trị bằng giá trị định trước chỉ báo tính không khả dụng của các mẫu.
8. Phương pháp theo điểm 1, trong đó khoảng giá trị pixel được biểu diễn dưới dạng $[0, 1 < (bit_depth) - 1]$, trong đó bit_depth là số nguyên dương.
9. Phương pháp theo điểm 8, trong đó bit_depth là giá trị chính xác được sử dụng để xử lý mẫu.
10. Phương pháp theo điểm 1, trong đó việc biến đổi bao gồm mã hóa khối video hiện tại thành dòng bit.
11. Phương pháp theo điểm 1, trong đó việc biến đổi bao gồm bước giải mã khối video hiện tại từ dòng bit.
12. Thiết bị xử lý dữ liệu video bao gồm bộ xử lý và bộ nhớ bất biến có các lệnh trên đó, trong đó các lệnh khi thực thi bằng bộ xử lý, khiến bộ xử lý:
 - xác định, đối với biến đổi giữa khối video hiện tại của ảnh hiện tại của video và dòng bit của video, rằng chế độ dự báo được áp dụng trên khối video hiện tại;
 - duy trì, đối với khối video hiện tại, bộ đệm bao gồm các mẫu tham chiếu được dẫn xuất từ các khối của các giá trị mẫu của cùng vùng video của khối video hiện tại trong ảnh hiện tại,
 - tạo, đối với khối video hiện tại, các mẫu dự báo dựa trên xác định; và
 - thực hiện biến đổi giữa khối video hiện tại và dòng bit,
 - trong đó trong chế độ dự báo, các mẫu tham chiếu được xác định bởi vector khối trong bộ đệm được sử dụng để dẫn xuất các mẫu dự báo của khối video hiện tại, và

trong đó khối tham chiếu cụ thể trong bộ đệm được đánh dấu là không khả dụng để dẫn xuất có các giá trị nằm ngoài khoảng giá trị pixel, và

trong đó các giá trị của các mẫu tham chiếu trong khối tham chiếu cụ thể được đánh dấu là không khả dụng được gán bằng giá trị định trước -1 chỉ báo tính không khả dụng của các mẫu.

13. Thiết bị theo điểm 12, trong đó vị trí và kích thước của khối tham chiếu cụ thể được đánh dấu là không khả dụng được xác định dựa trên một hoặc nhiều trong số: kích thước của khối video hiện tại, kích thước của khối ảo, hoặc kích thước của khối cây mã bao gồm khối video hiện tại.

14. Thiết bị theo điểm 13, trong đó vị trí của khối tham chiếu cụ thể được đánh dấu là không khả dụng được xác định dựa trên $(x0, y0)$,

trong đó $(x0 \% Vsize)$ bằng 0 và $(y0 \% Vsize)$ bằng 0,

trong đó % ký hiệu phép toán phần dư,

trong đó Vsize ký hiệu kích thước của khối ảo và $Vsize = \min(ctbSize, 64)$,

và

trong đó ctbSize ký hiệu chiều rộng hoặc chiều cao của khối cây mã.

15. Thiết bị theo điểm 13, trong đó kích thước của khối tham chiếu cụ thể được đánh dấu không khả dụng được xác định dựa trên kích thước của khối ảo nếu kích thước của khối video hiện tại nhỏ hơn kích thước của khối ảo.

16. Thiết bị theo điểm 13, trong đó kích thước của khối tham chiếu cụ thể được đánh dấu không khả dụng được xác định dựa trên kích thước của khối video hiện tại nếu kích thước của khối video hiện tại lớn hơn kích thước của khối ảo.

17. Thiết bị theo điểm 12, trong đó bộ đệm được thiết lập lại trước khi tạo mã ảnh hoặc lát.

18. Thiết bị theo điểm 12, trong đó ràng buộc tuân thủ dòng bit được thỏa mãn bởi dòng bit bao gồm việc các mẫu tham chiếu được xác định bởi vector khối trong bộ đệm không có các giá trị bằng giá trị định trước chỉ báo sự không khả dụng của các mẫu.

19. Vật ghi máy tính đọc được bất biến lưu trữ các lệnh khiến bộ xử lý:

xác định, đối với biến đổi giữa khối video hiện tại của ảnh hiện tại của video và dòng bit của video, rằng chế độ dự báo được áp dụng trên khối video hiện tại;

duy trì, đối với khối video hiện tại, bộ đệm bao gồm các mẫu tham chiếu được dẫn xuất từ các khối của các giá trị mẫu của cùng vùng video của khối video hiện tại trong ảnh hiện tại,

tạo, đối với khối video hiện tại, các mẫu dự báo dựa trên xác định; và thực hiện biến đổi giữa khối video hiện tại và dòng bit,

trong đó trong chế độ dự báo, các mẫu tham chiếu được xác định bởi vector khối trong bộ đệm được sử dụng để dẫn xuất các mẫu dự báo của khối video hiện tại, và

trong đó khối tham chiếu cụ thể trong bộ đệm được đánh dấu là không khả dụng để dẫn xuất có các giá trị nằm ngoài khoảng giá trị pixel, và

trong đó các giá trị của các mẫu tham chiếu trong khối tham chiếu cụ thể được đánh dấu là không khả dụng được gán bằng giá trị định trước -1 chỉ báo tính không khả dụng của các mẫu.

20. Phương pháp lưu trữ dòng bit của video bao gồm các bước:

xác định, rằng chế độ dự báo được áp dụng trên khối video hiện tại của ảnh hiện tại của video;

duy trì, đối với khối video hiện tại, bộ đệm bao gồm các mẫu tham chiếu được dẫn xuất từ các khối của các giá trị mẫu của cùng vùng video của khối video hiện tại trong ảnh hiện tại,

tạo, đối với khối video hiện tại, các mẫu dự báo dựa trên xác định;

tạo dòng bit dựa trên các mẫu dự báo, và

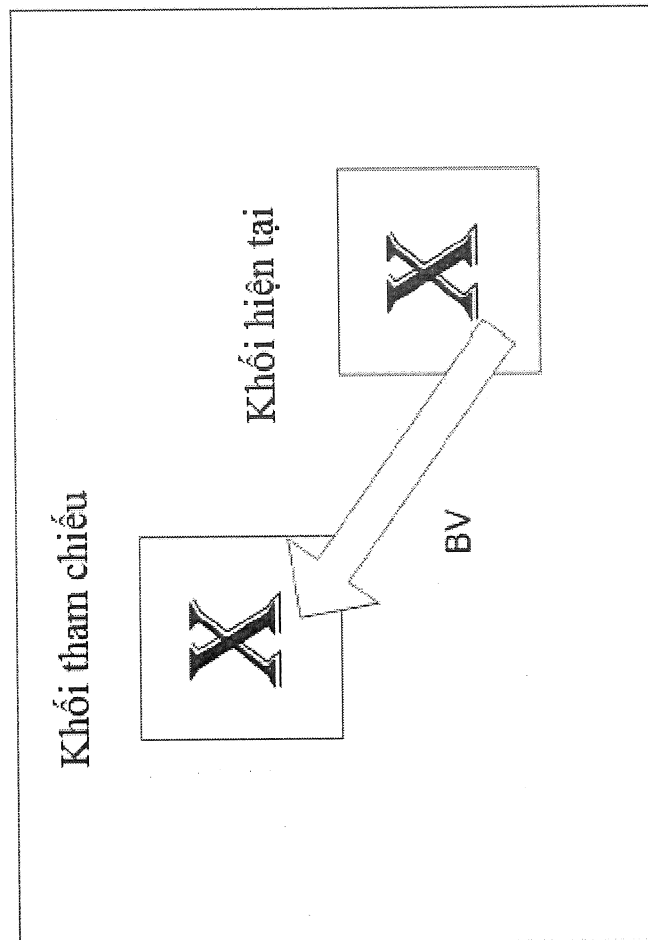
lưu trữ dòng bit trong vật ghi máy tính đọc được bất biến,

trong đó trong chế độ dự báo, các mẫu tham chiếu được xác định bởi vector khối trong bộ đệm được sử dụng để dẫn xuất các mẫu dự báo của khối video hiện tại, và

trong đó khối tham chiếu cụ thể trong bộ đệm được đánh dấu là không khả dụng để dẫn xuất có các giá trị nằm ngoài khoảng giá trị pixel, và

trong đó các giá trị của các mẫu tham chiếu trong khối tham chiếu cụ thể được đánh dấu là không khả dụng được gán bằng giá trị định trước -1 chỉ báo tính không khả dụng của các mẫu.

1/34



Ảnh hiện tại

Fig.1

2/34

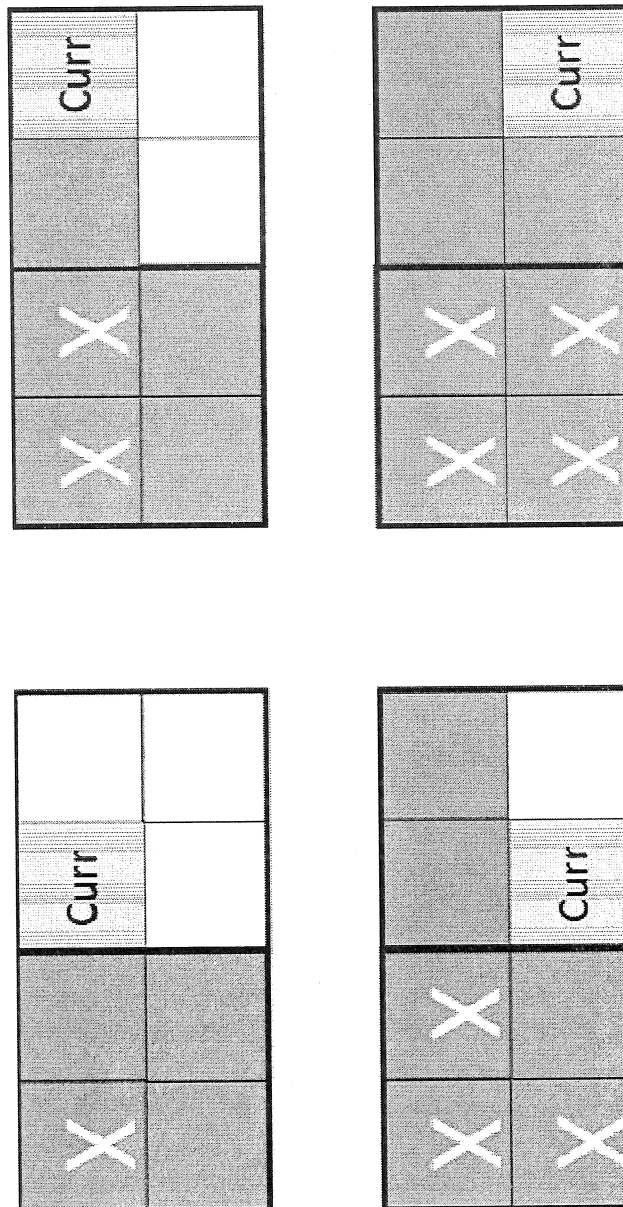


Fig.2

3/34

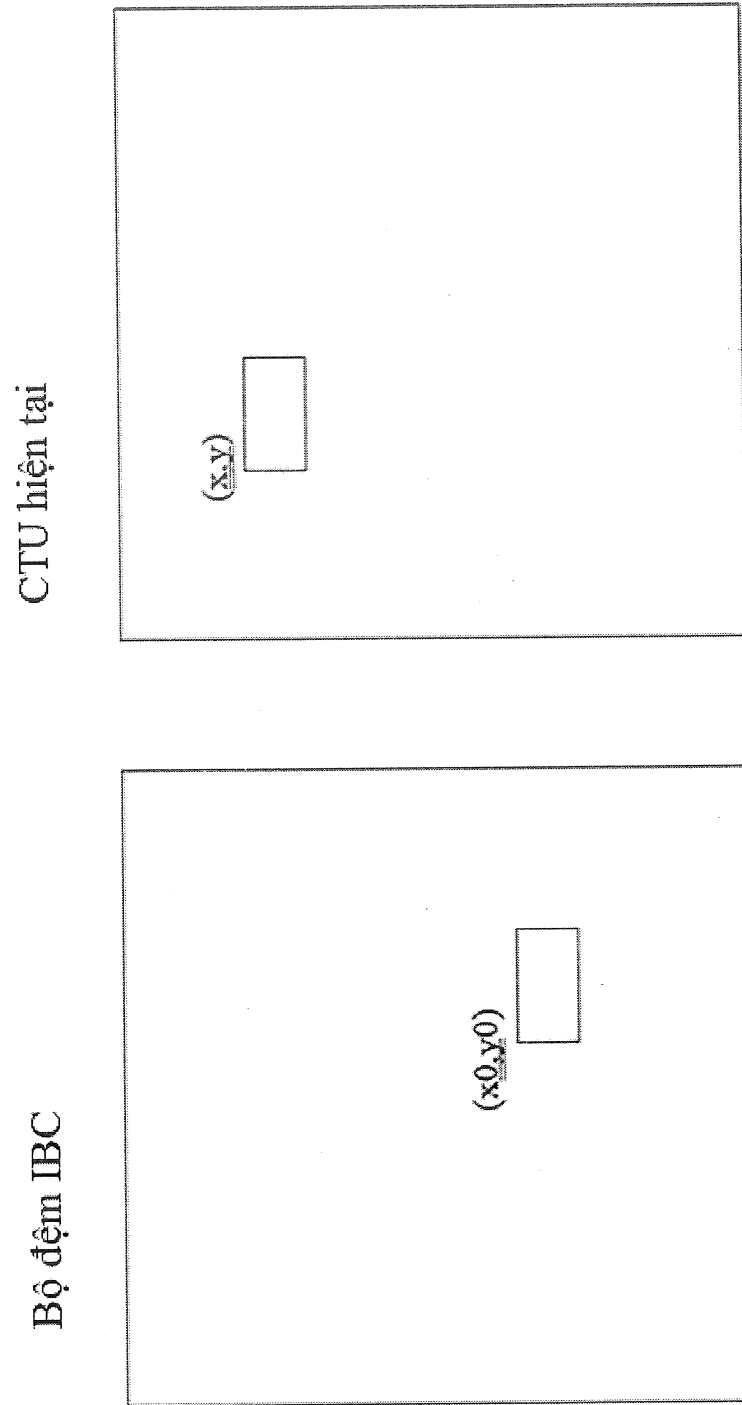


Fig.3

4/34

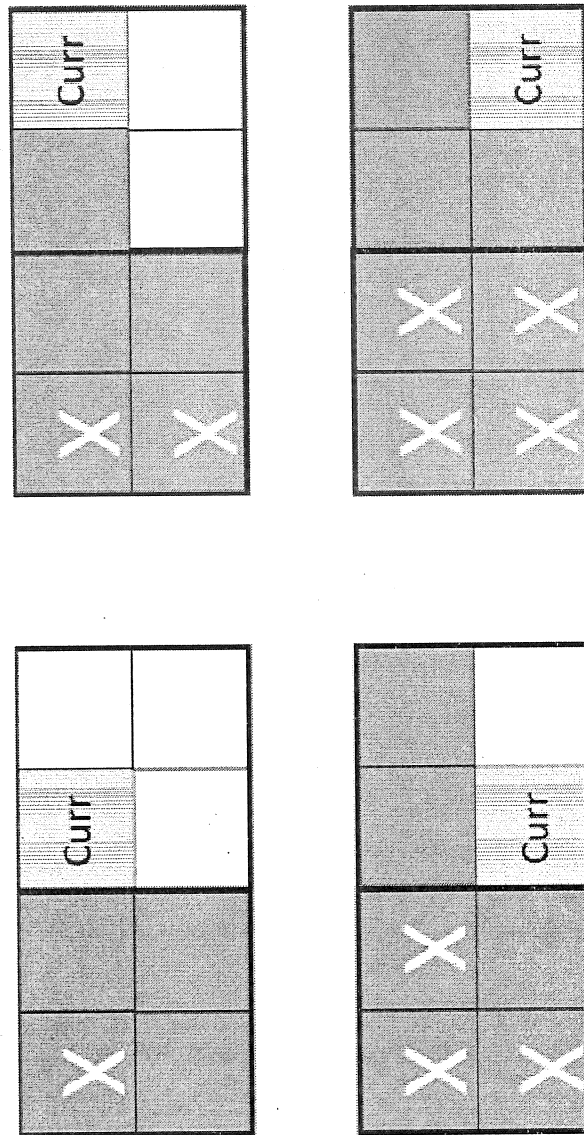


Fig.4

5/34

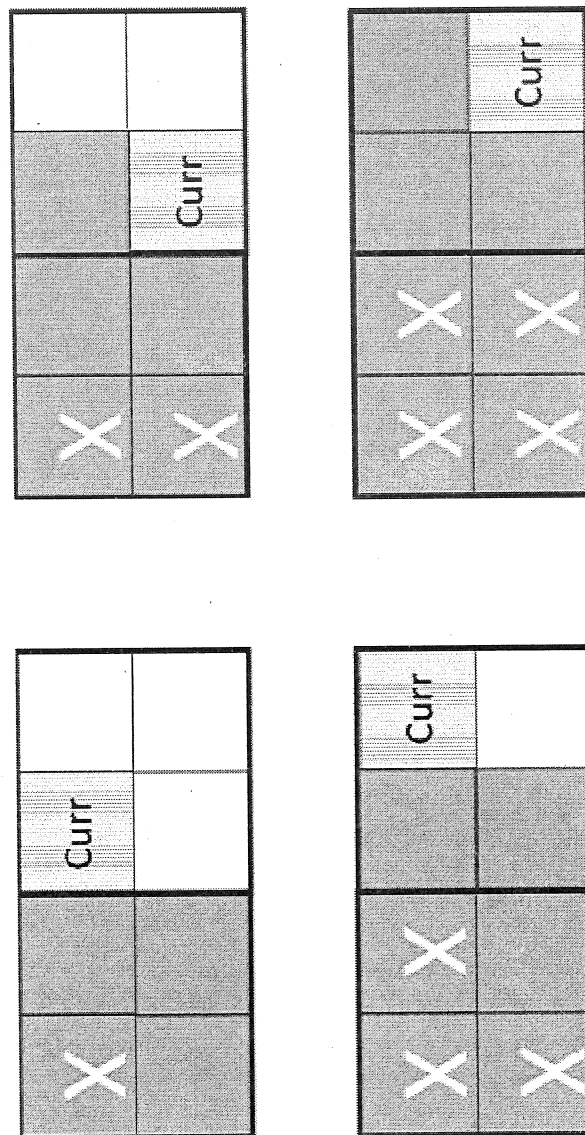


Fig.5

6/34

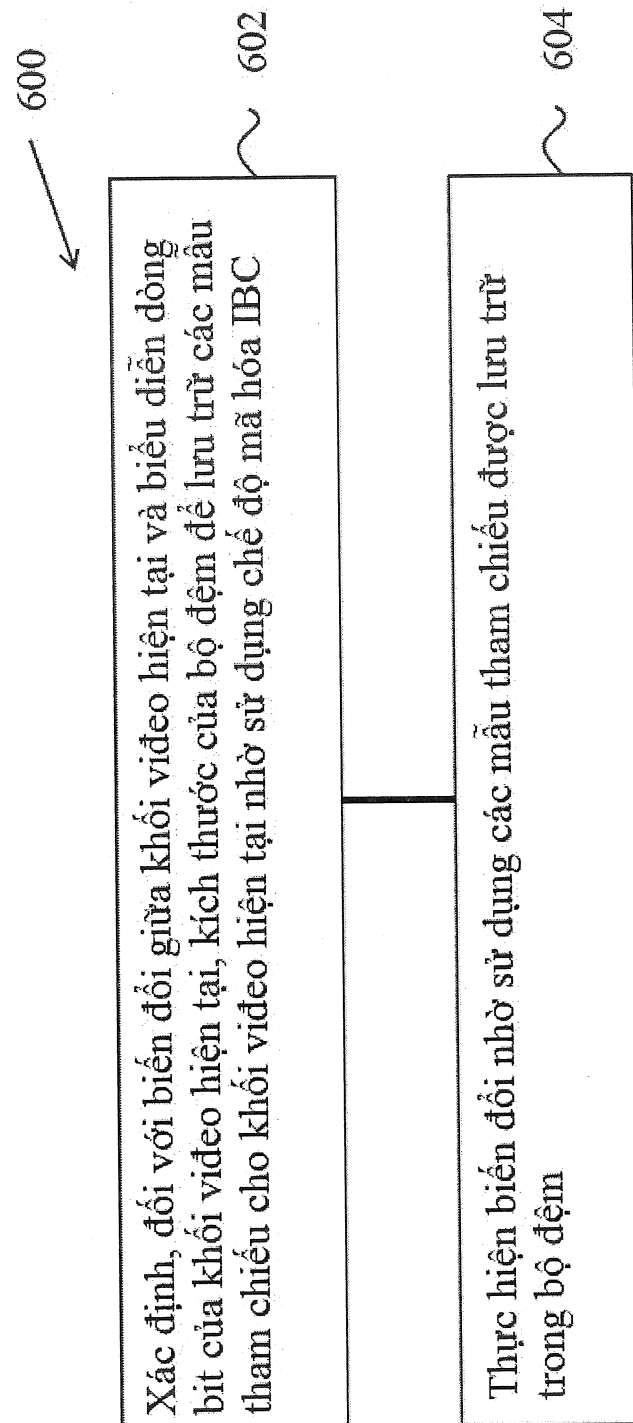


Fig.6

7/34

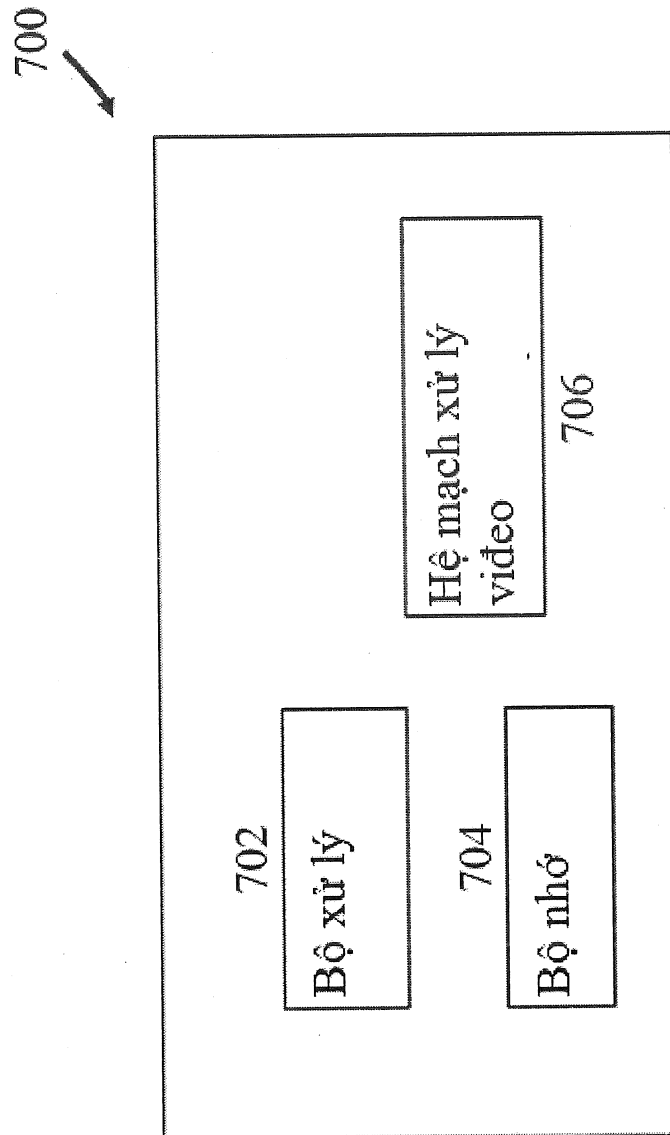


Fig. 7

8/34

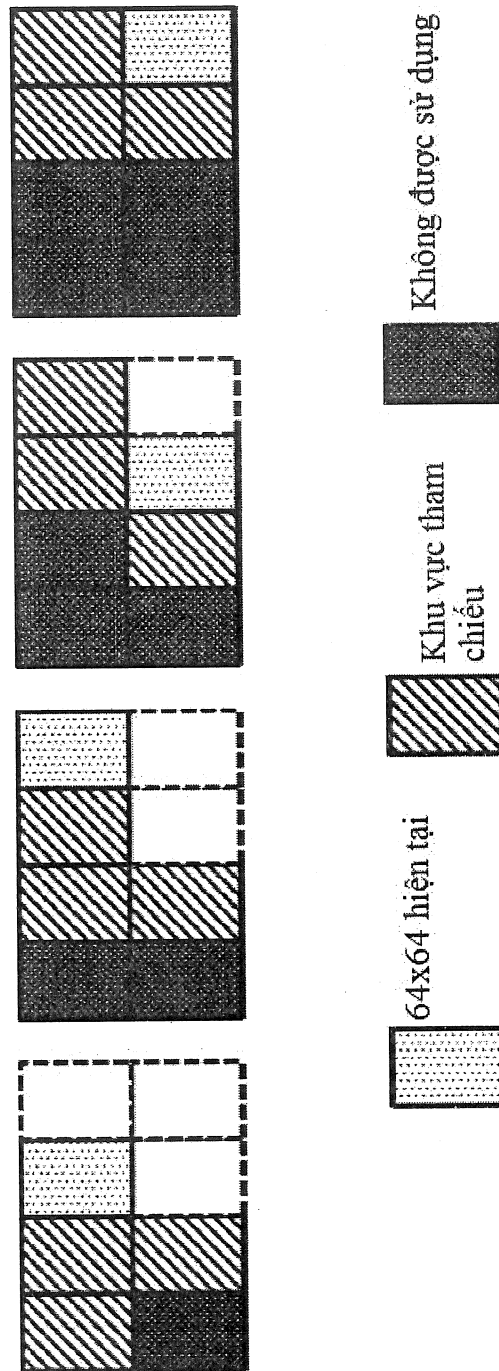


Fig.8

9/34

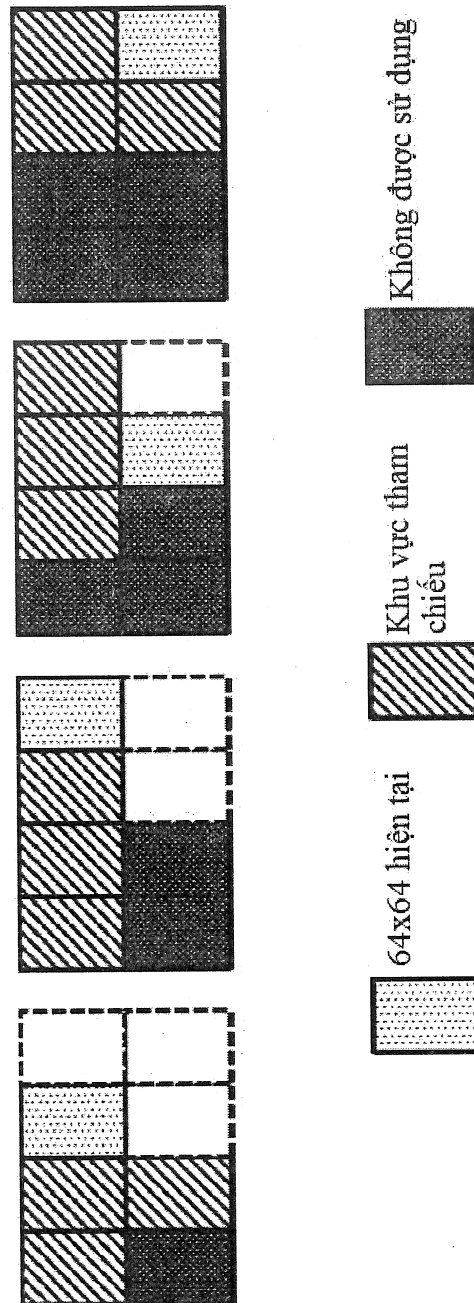


Fig.9

10/34

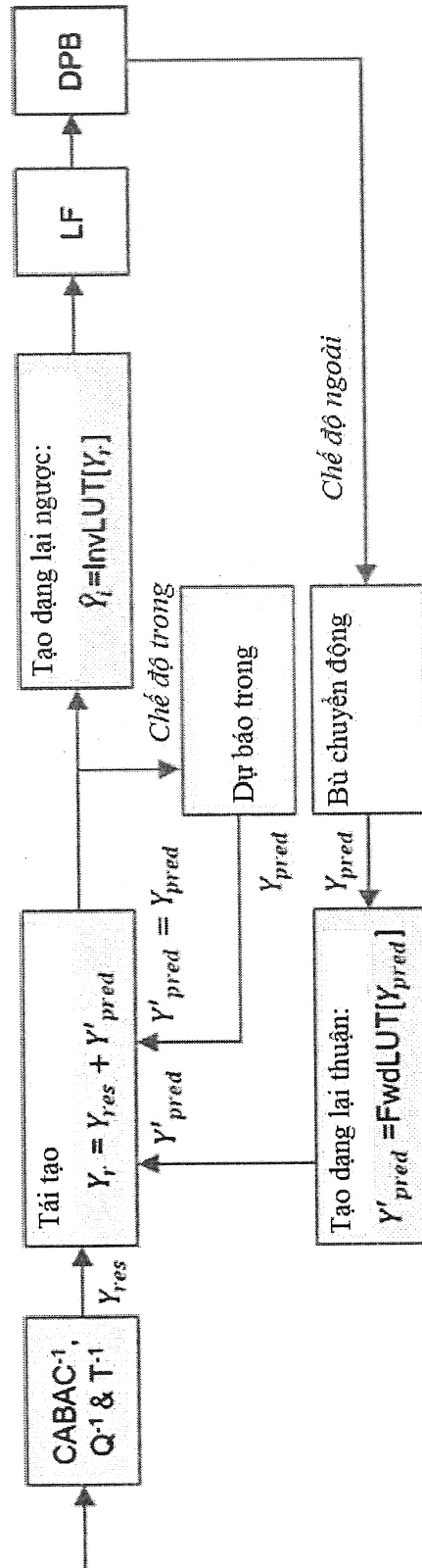


Fig.10

11/34

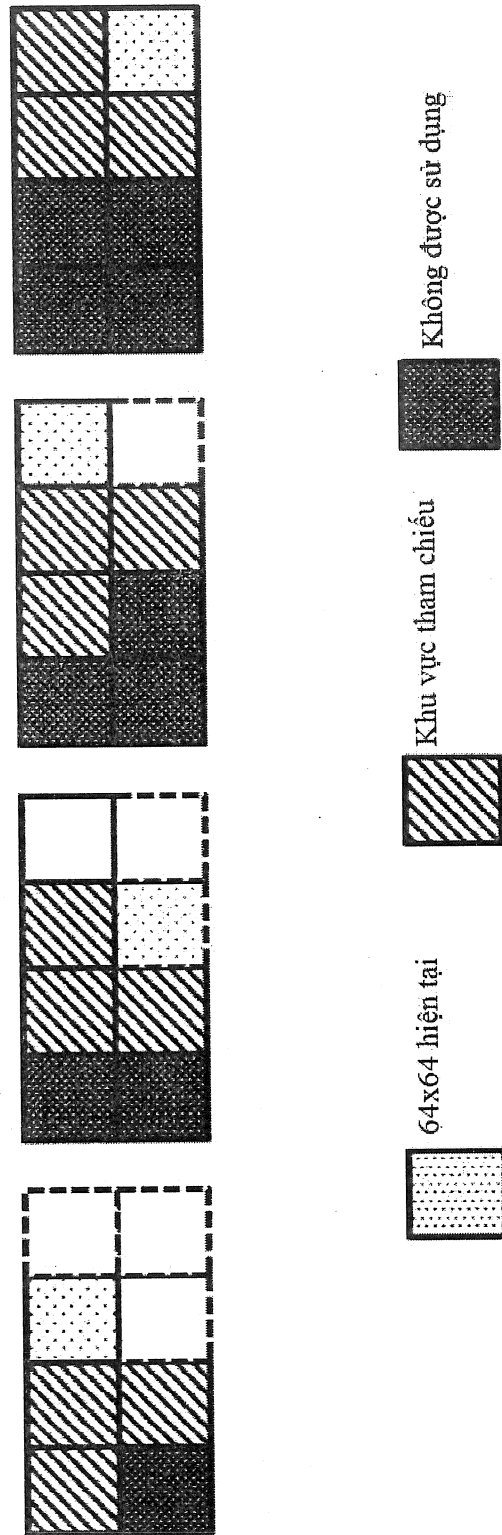


Fig.11

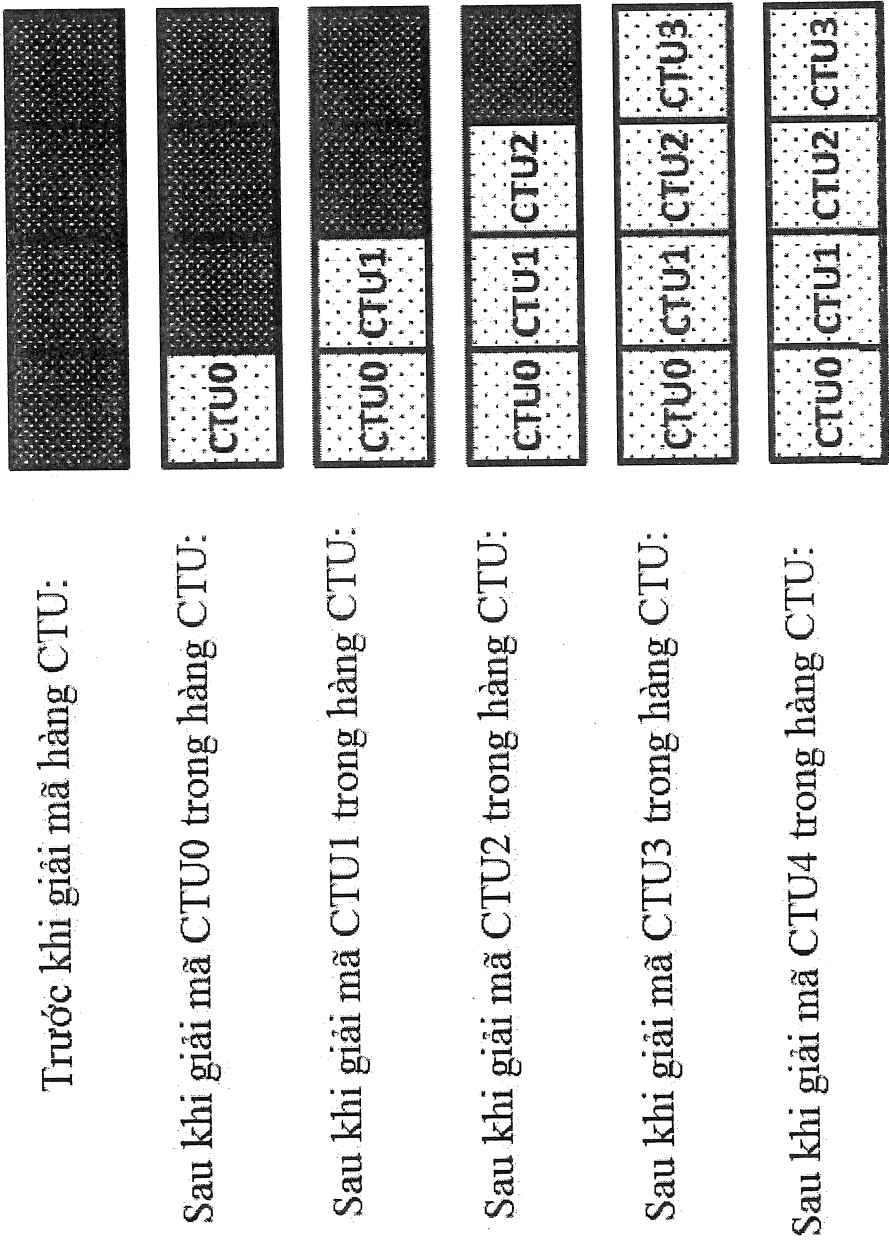


Fig.12

13/34

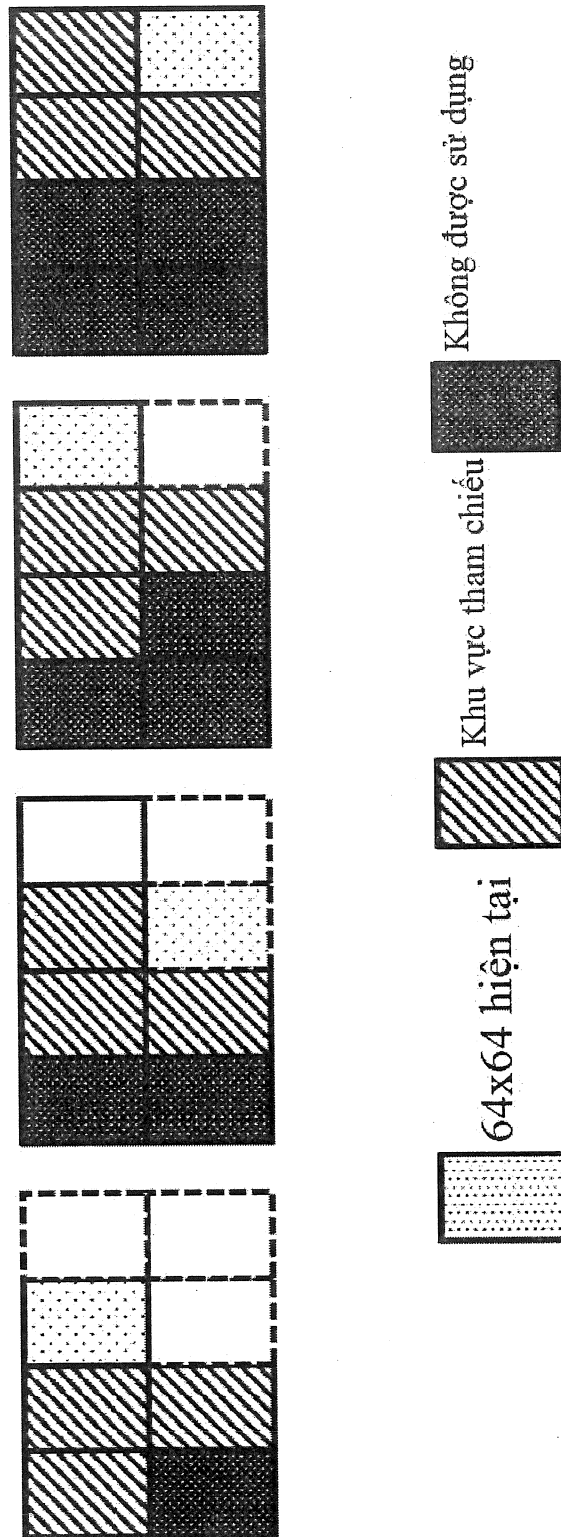


Fig.13

14/34

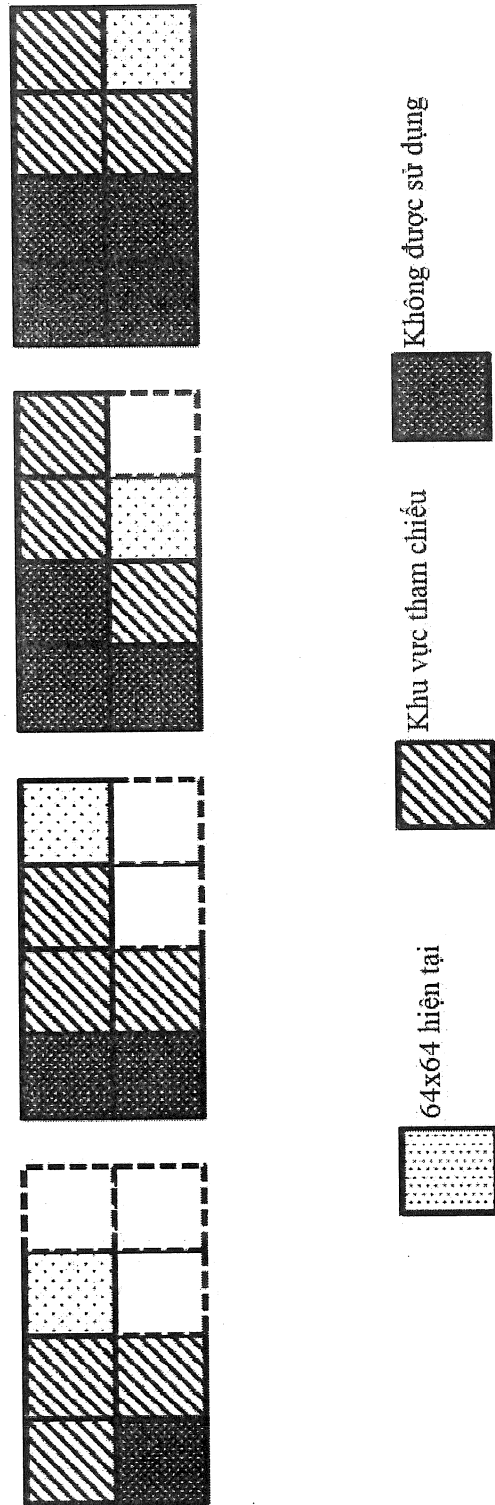


Fig.14

15/34

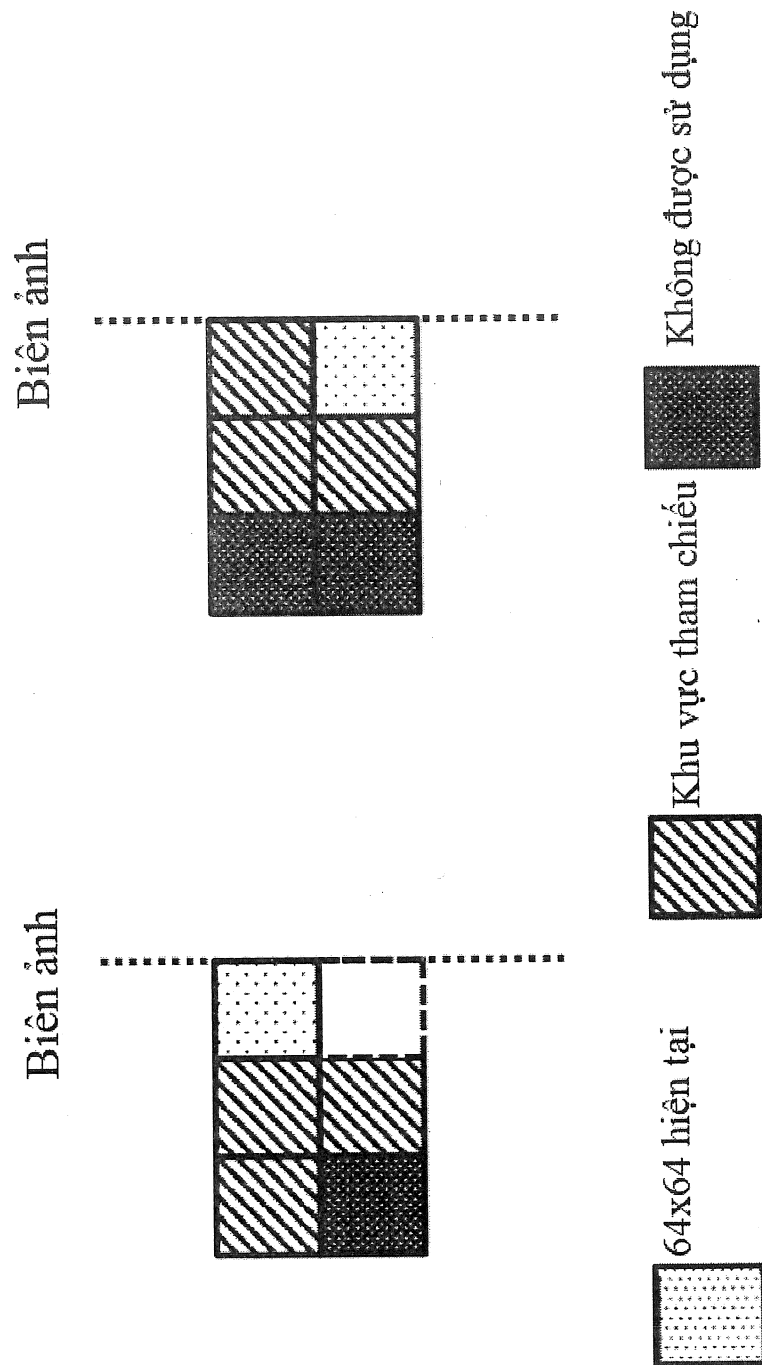


Fig.15

16/34

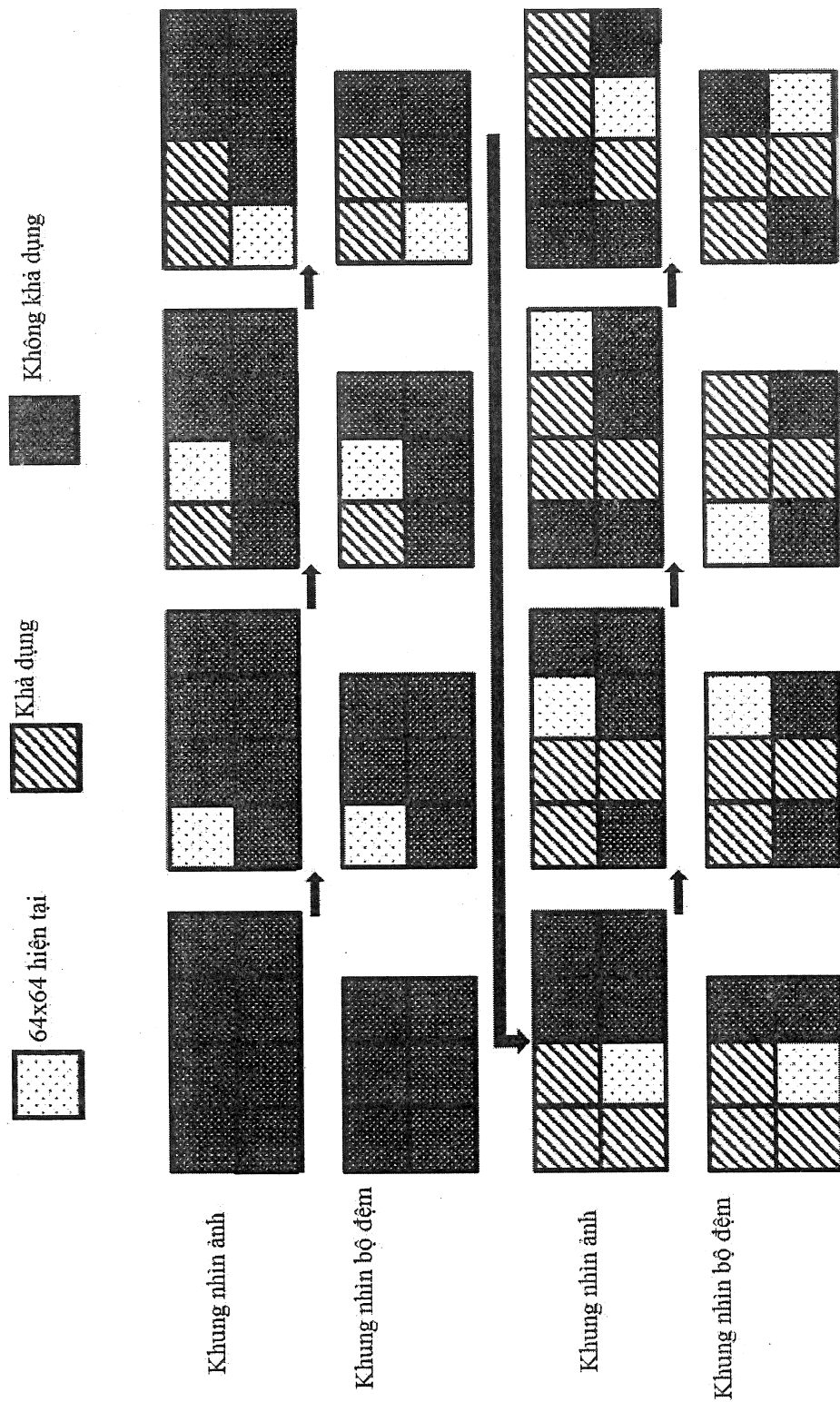


Fig.16

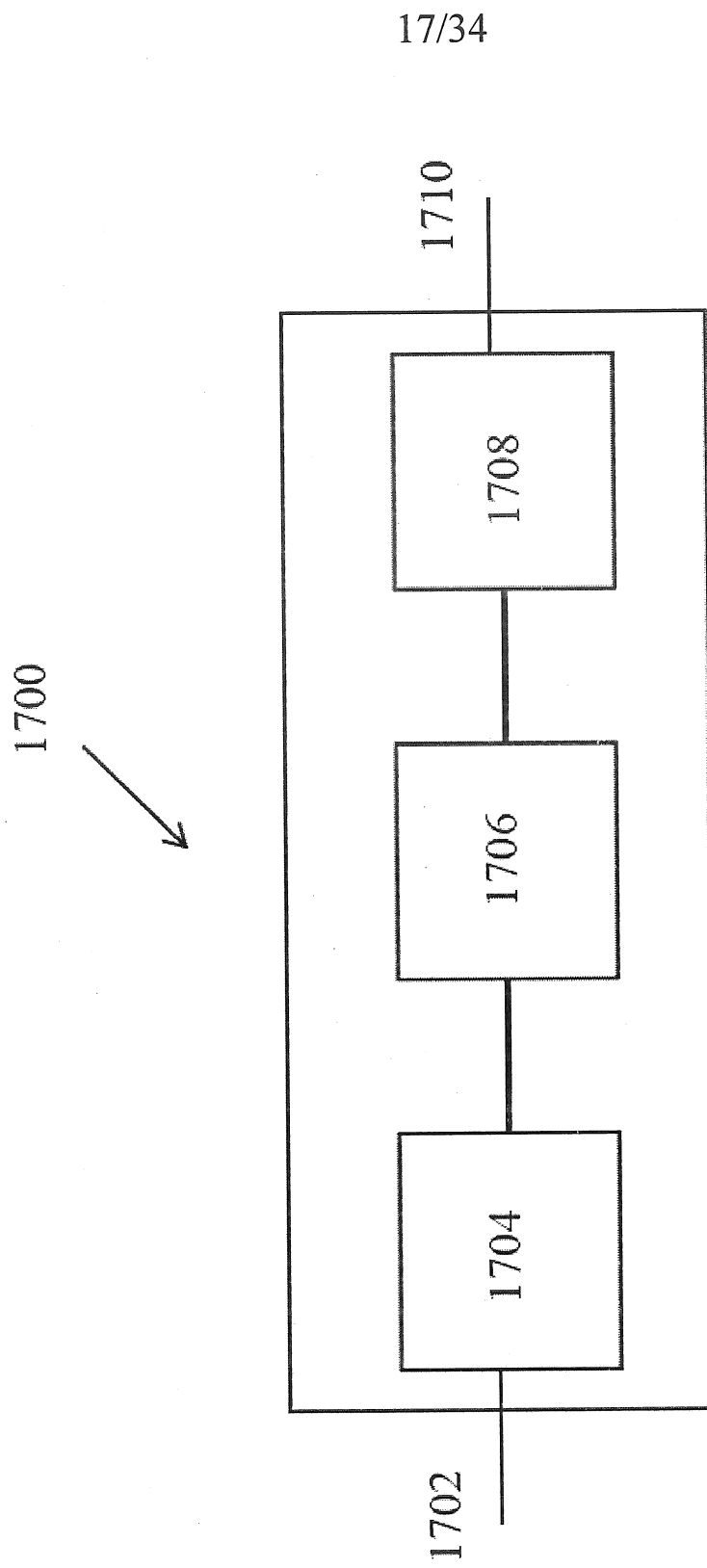


Fig.17

18/34

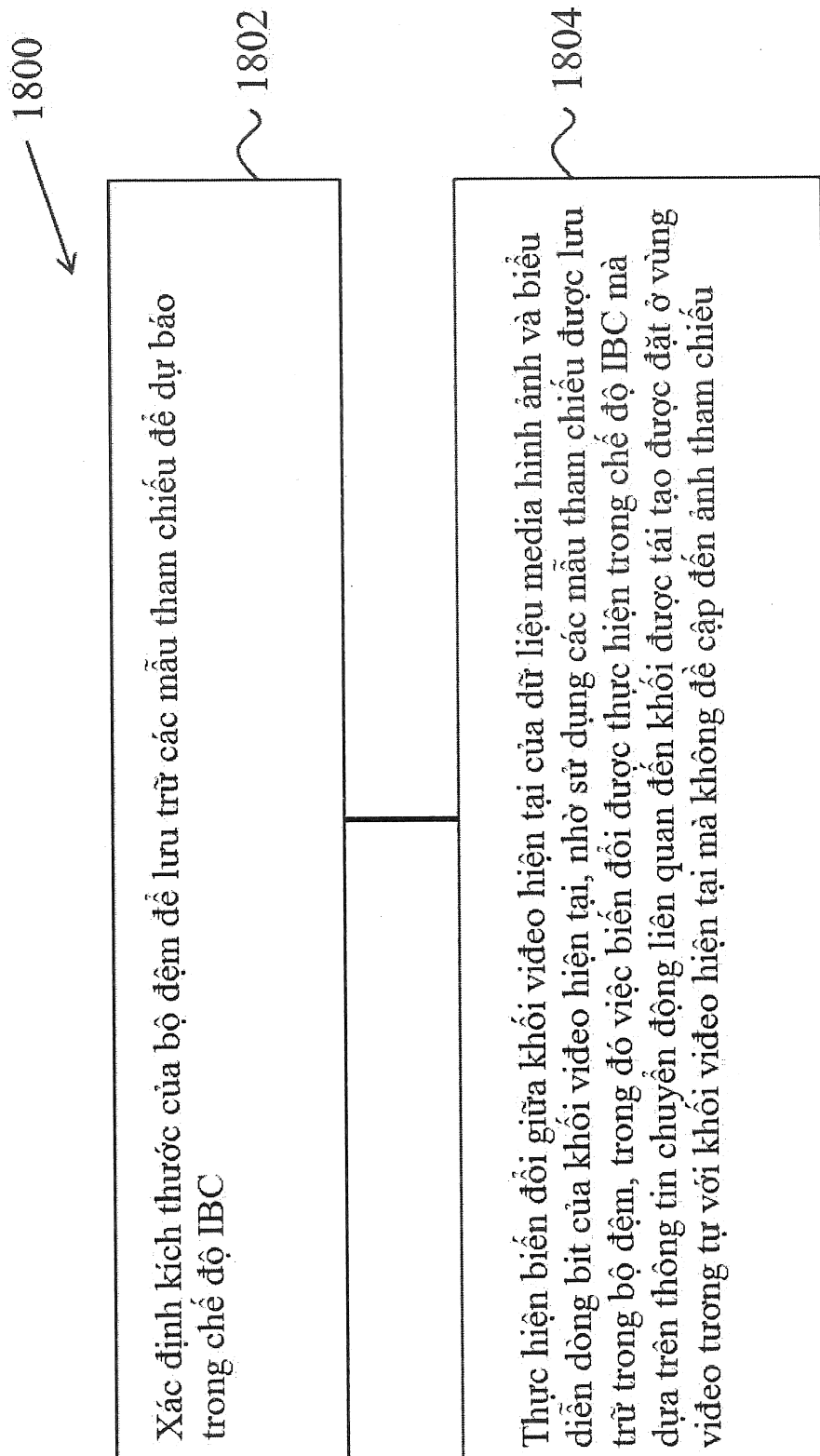


Fig.18

19/34

1900



1902

Xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu được tái tạo để dự báo trong chế độ IBC, trong đó bộ đệm được sử dụng để lưu trữ các mẫu được tái tạo trước bước lọc vòng

1904

Thực hiện biến đổi mà sử dụng các mẫu được tái tạo được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không viện dẫn đến ảnh tham chiếu

Fig.19

20/34

2000
↙

2002
~

2004
~

Xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu được tái tạo để dự báo trong chế độ IBC, trong đó bộ đệm được sử dụng để lưu trữ các mẫu được tái tạo sau bước lọc vòng

Thực hiện biến đổi mà sử dụng các mẫu được tái tạo được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu

Fig.20

21/34

2100
↙

Xác định, đối với biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, bộ đệm mà lưu trữ các mẫu được tái tạo để dự báo trong chế độ IBC, trong đó bộ đệm được sử dụng để lưu trữ các mẫu được tái tạo cả trước bước lọc vòng và sau bước lọc vòng

2102

Thực hiện biến đổi mà sử dụng các mẫu được tái tạo được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu

2104

Fig.21

22/34

2200
↙

Sử dụng bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó chiều sâu bit thứ nhất của bộ đệm khác với chiều sâu bit thứ hai được sử dụng để biểu diễn dữ liệu media hình ảnh trong biểu diễn dòng bit

2202

Thực hiện biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu

2204

Fig.22

23/34

2300
↙

Fig.23

Khởi tạo bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó bộ đệm được khởi tạo với giá trị thử nhất

2302

2304

Thực hiện biến đổi giữa khối video hiện tại của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu

24/34

2400



Fig. 24

Khởi tạo bộ đệm để lưu trữ các mẫu tham chiếu để dự báo trong chế độ IBC, trong đó, dựa trên tính khả dụng của một hoặc nhiều khối video trong dữ liệu media hình ảnh, bộ đệm được khởi tạo với các giá trị pixel của một hoặc nhiều khối video trong dữ liệu media hình ảnh

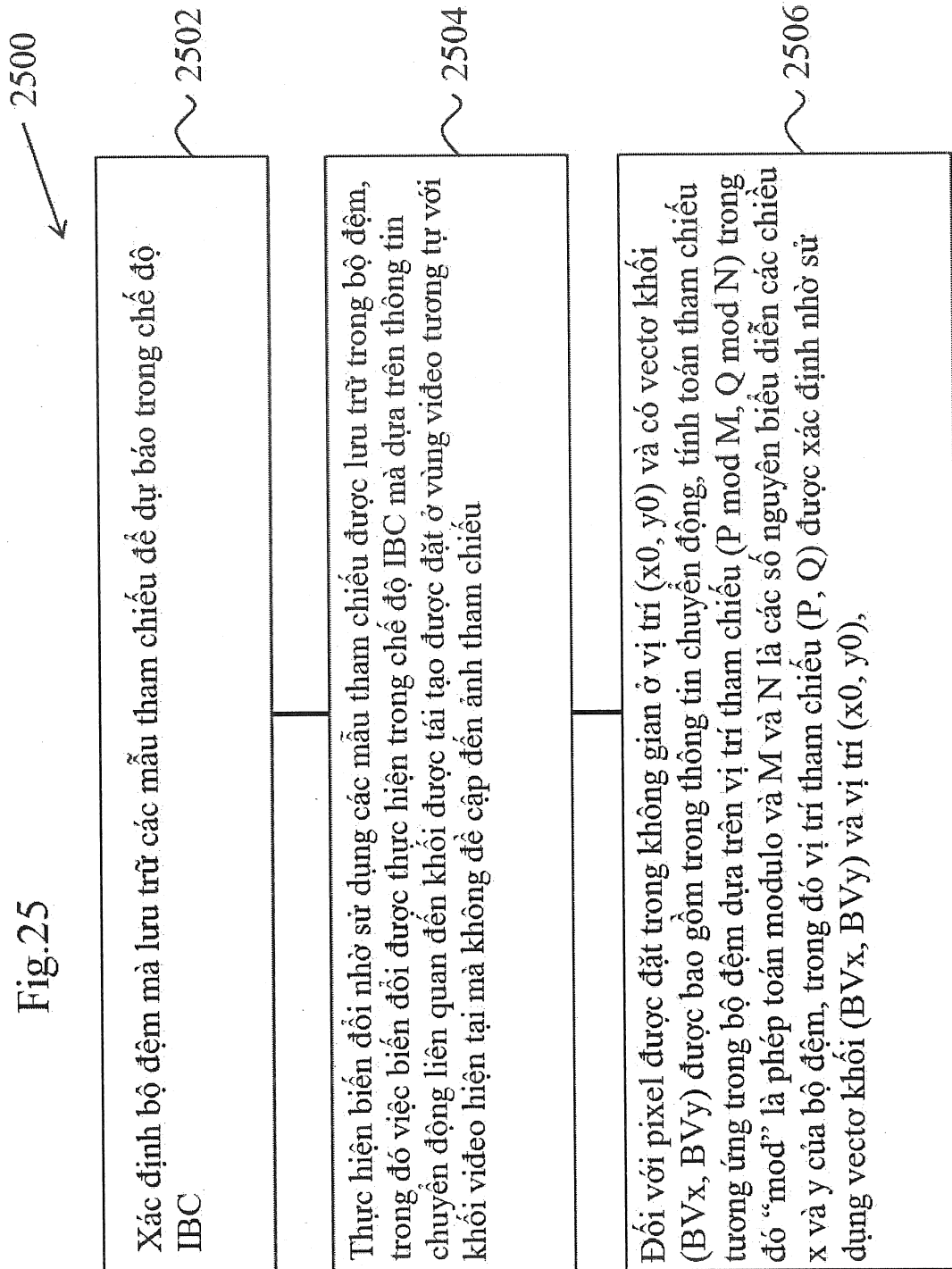
2402

2404

Thực hiện biến đổi giữa khối video hiện tại mà không thuộc một hoặc nhiều khối video của dữ liệu media hình ảnh và biểu diễn dòng bit của khối video hiện tại, nhờ sử dụng các mẫu tham chiếu được lưu trữ trong bộ đệm, trong đó việc biến đổi được thực hiện trong chế độ IBC mà dựa trên thông tin chuyển động liên quan đến khối được tái tạo được đặt ở vùng video tương tự với khối video hiện tại mà không đề cập đến ảnh tham chiếu

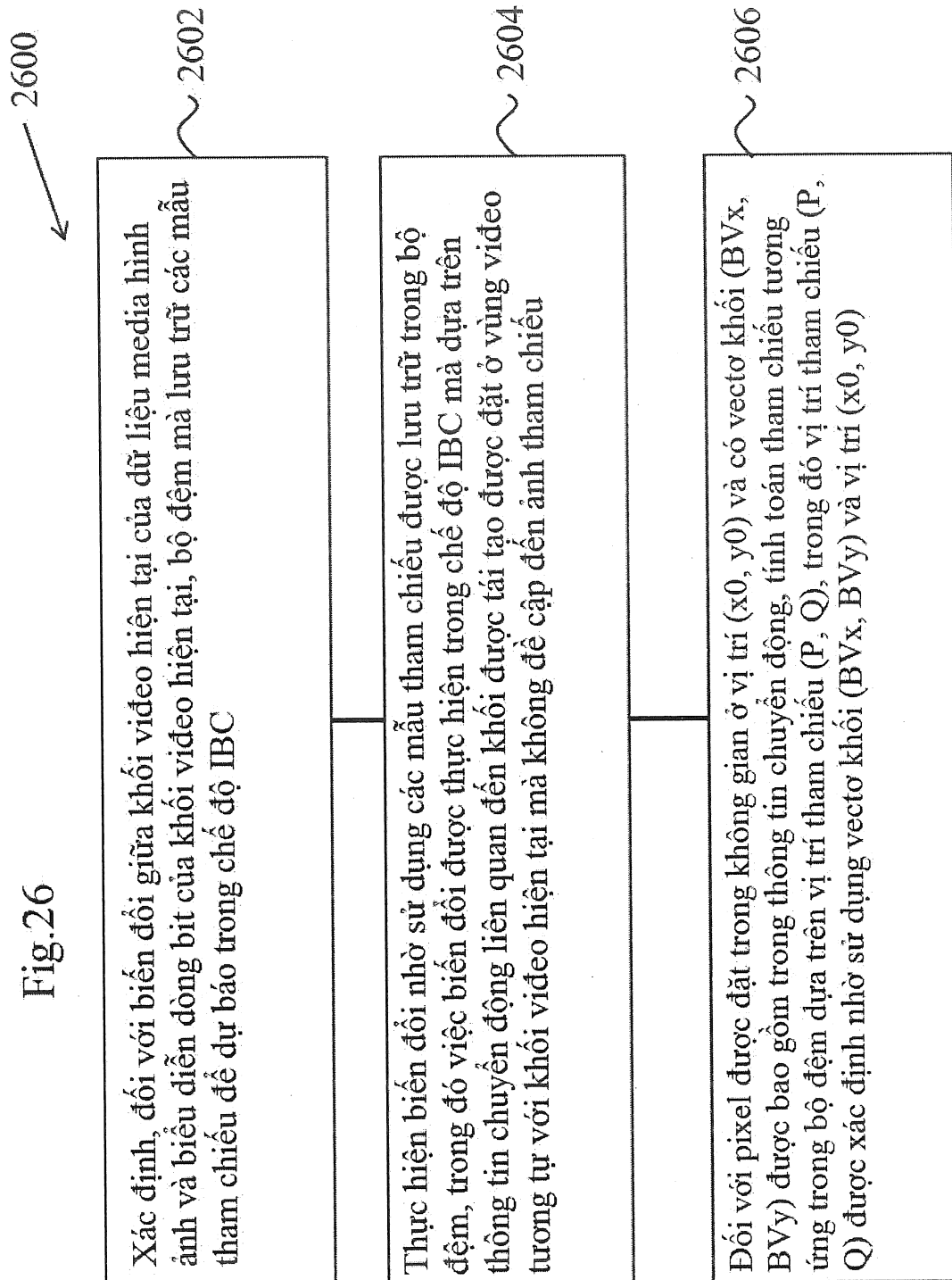
25/34

Fig.25



26/34

Fig.26



27/34

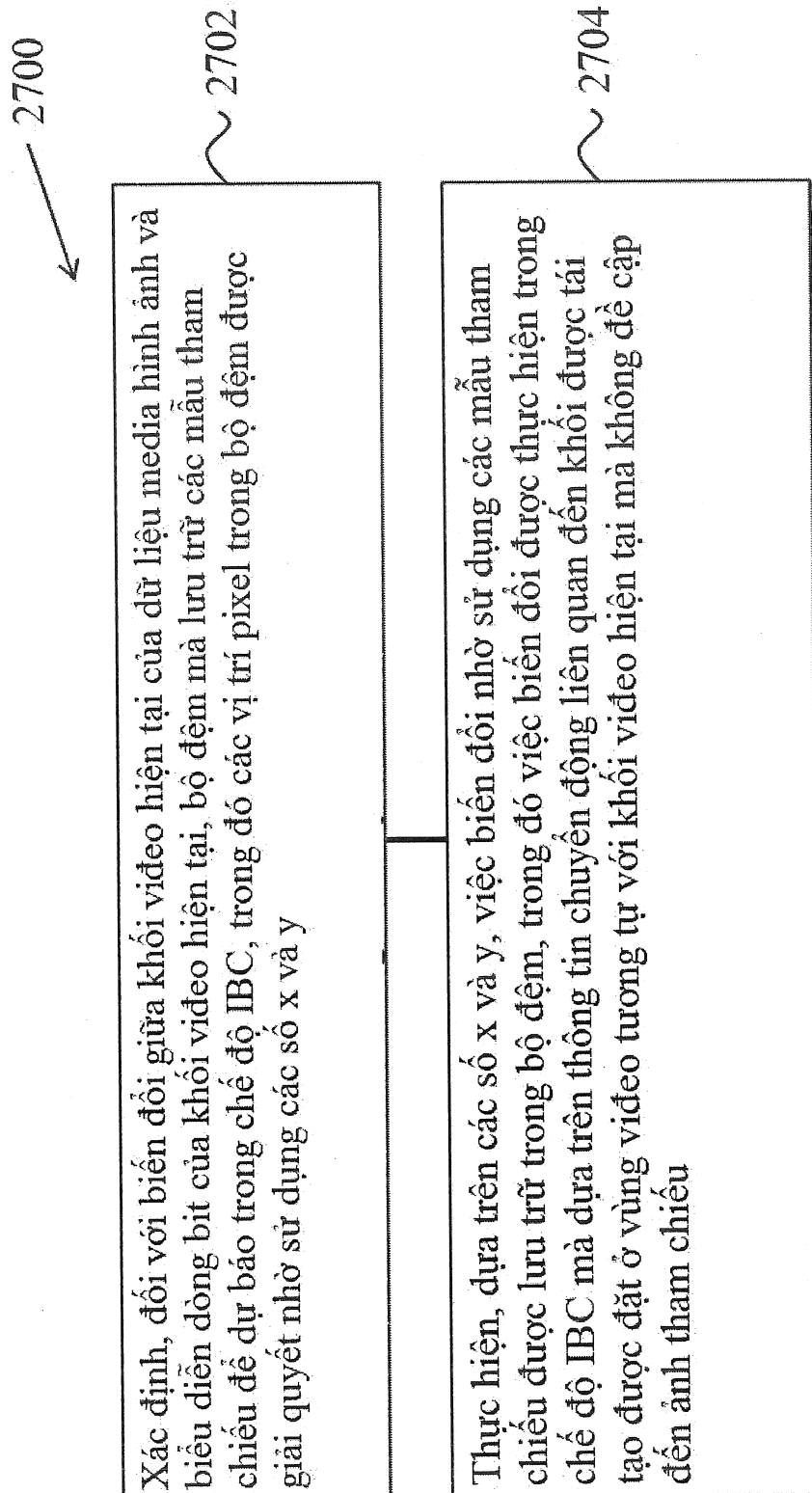
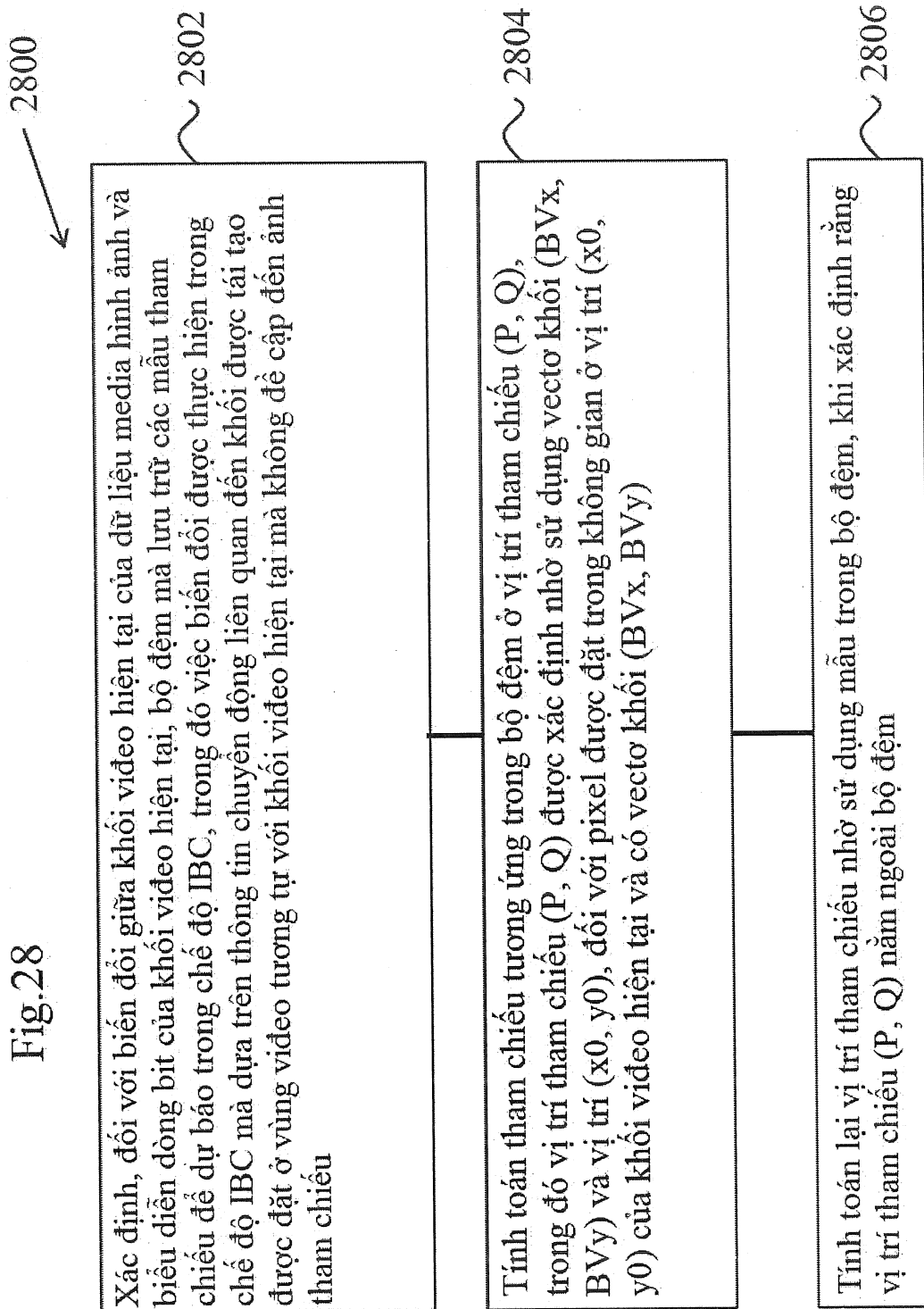


Fig.27

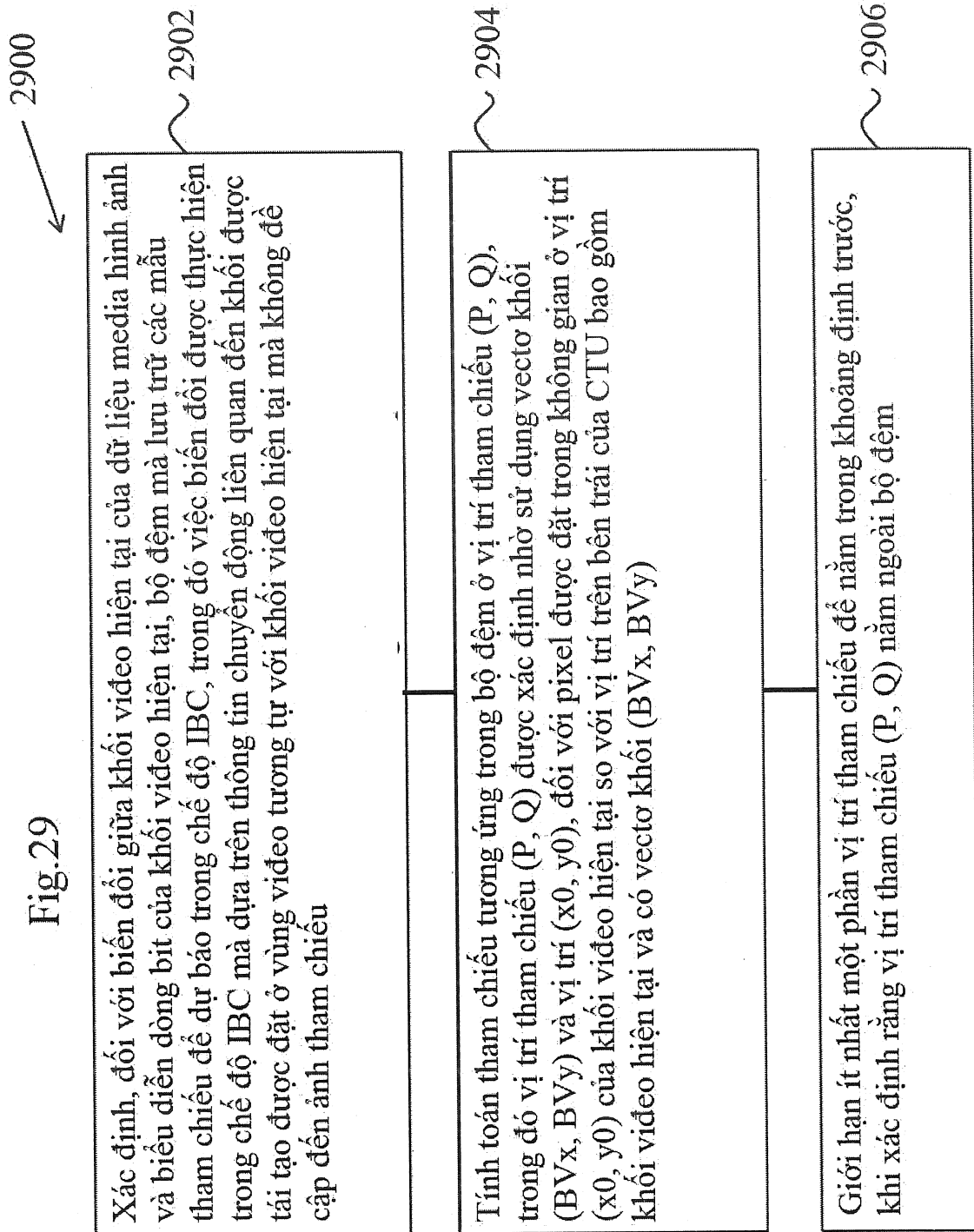
28/34

Fig.28



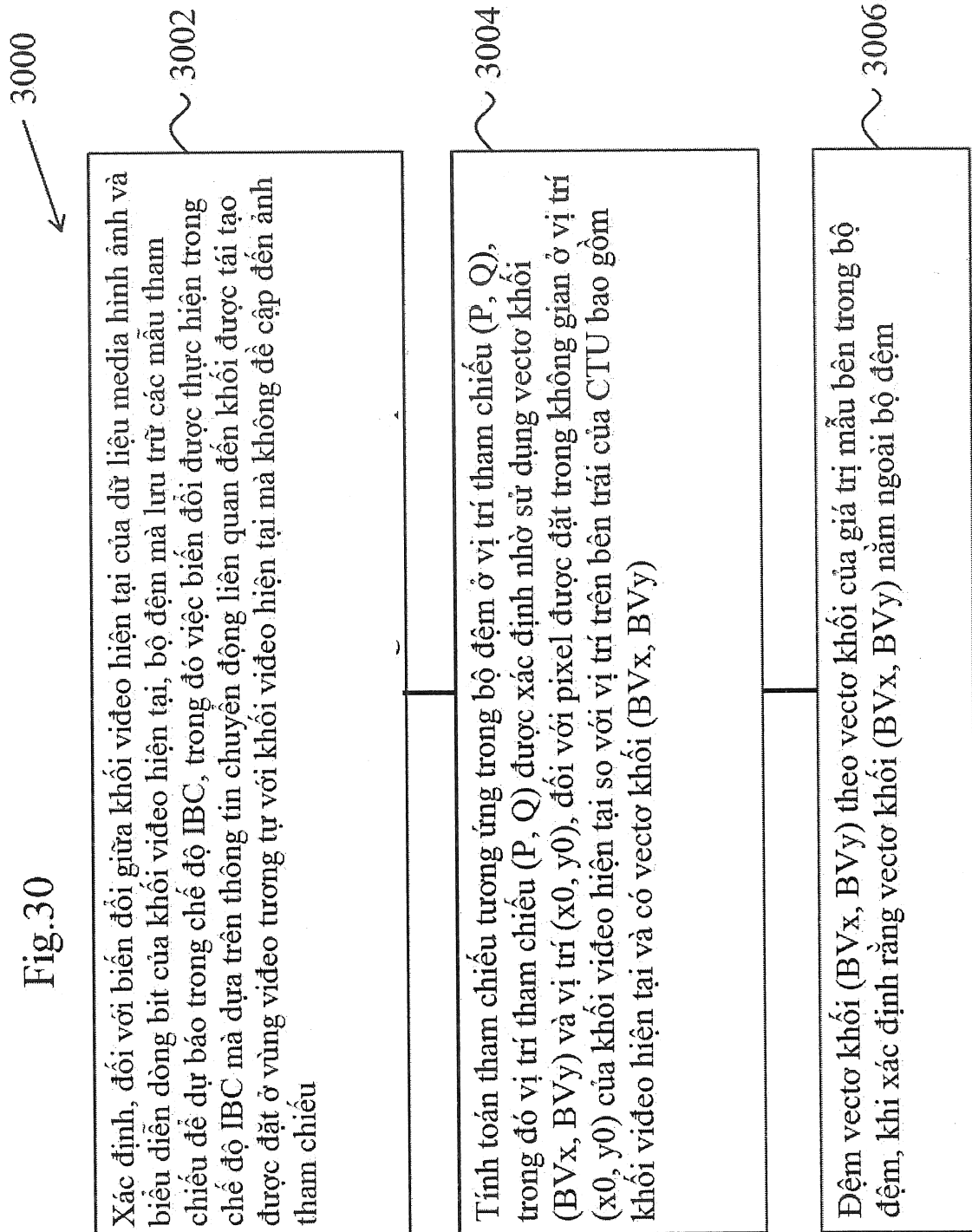
29/34

Fig.29



30/34

Fig.30



31/34

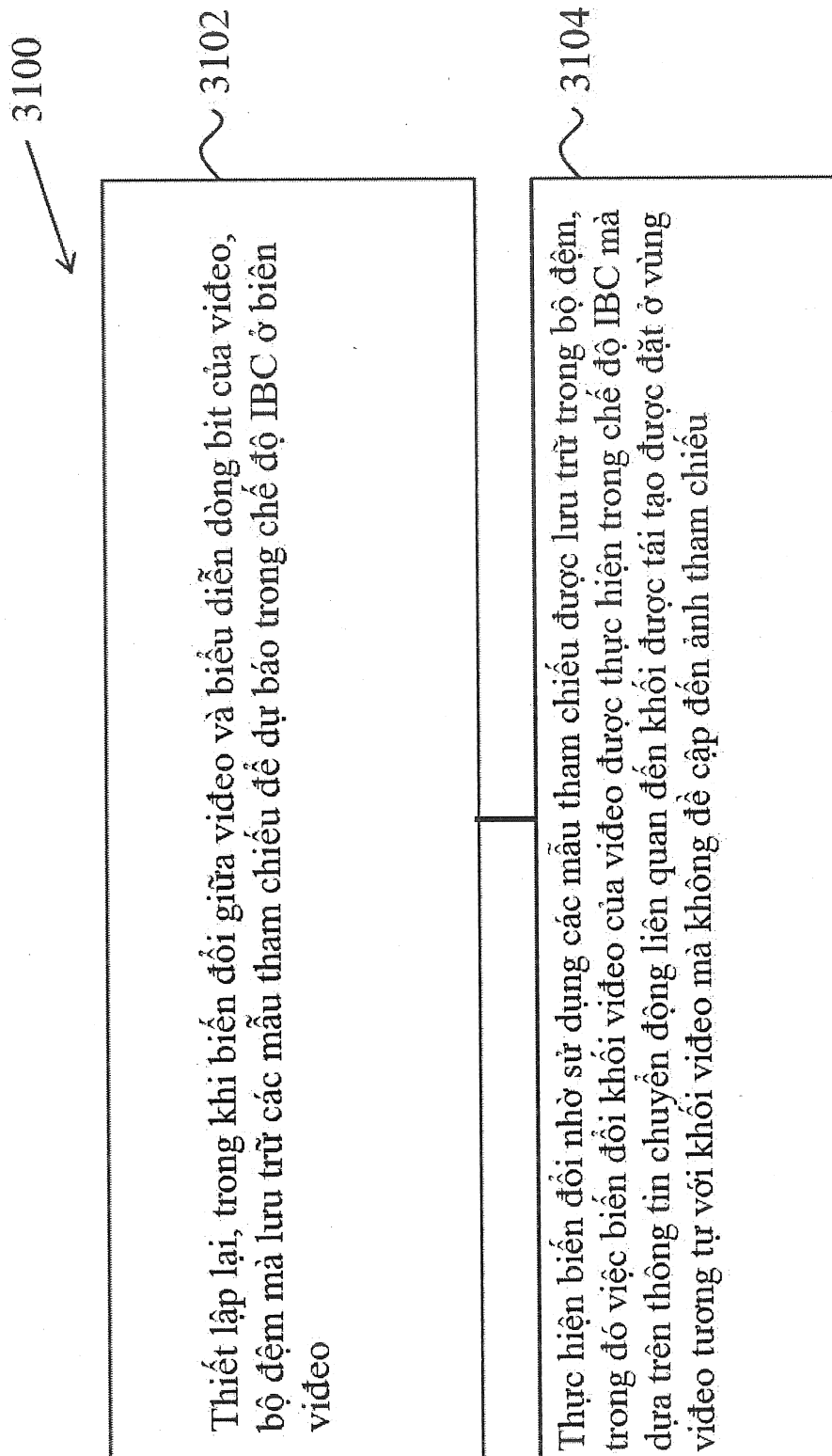
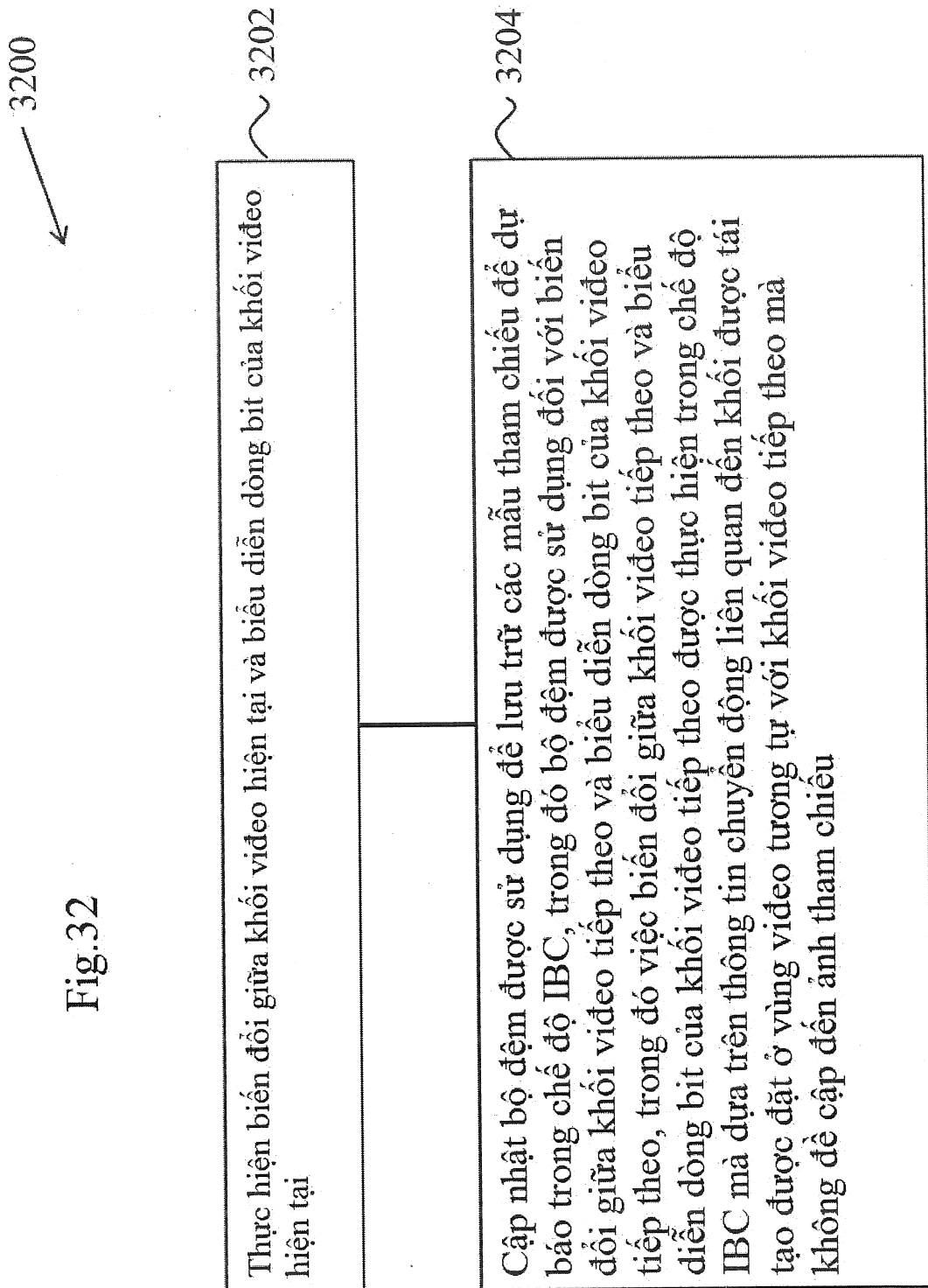


Fig.31

32/34

Fig. 32



33/34

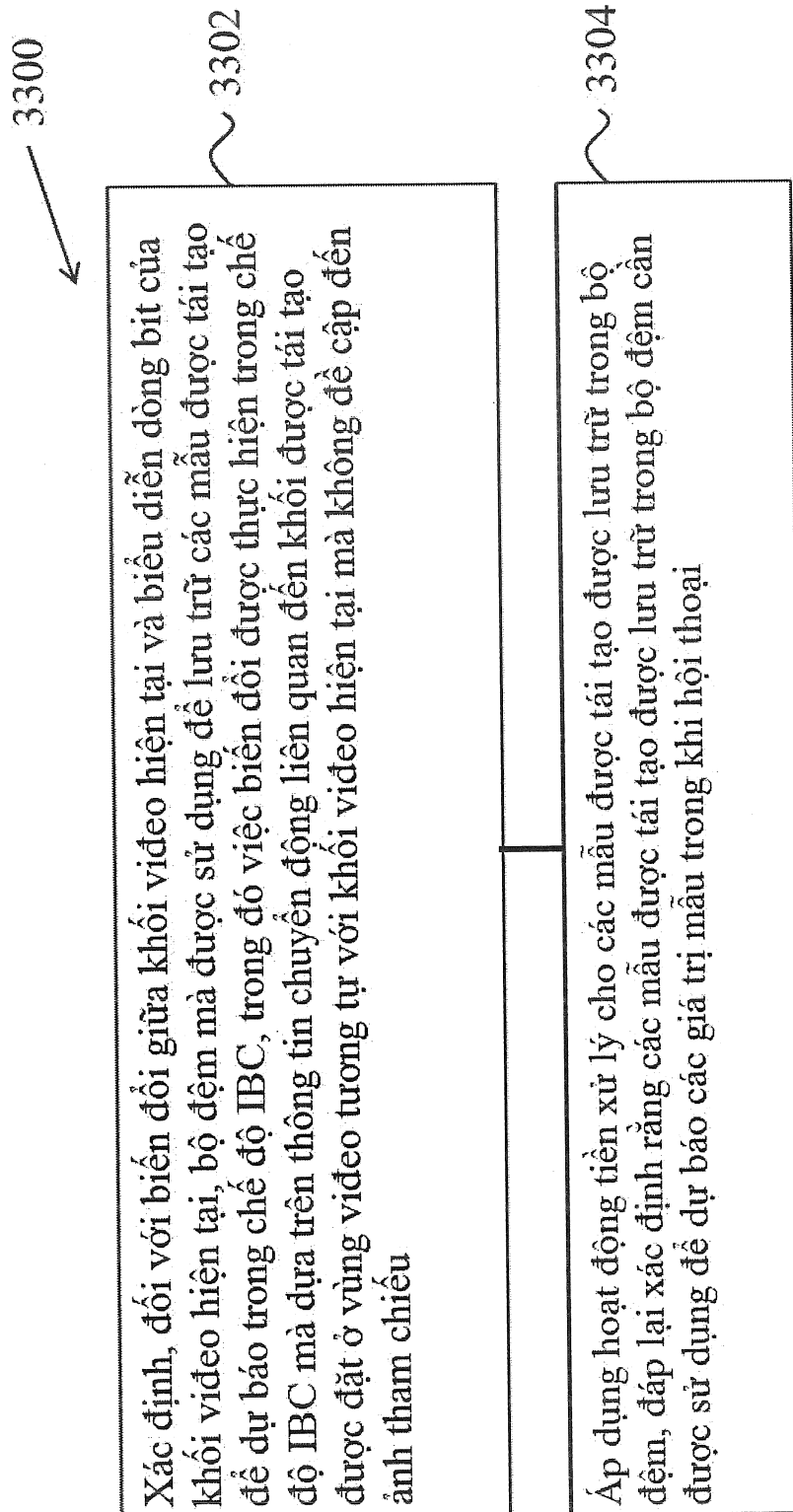


Fig.33

34/34

Fig.34

