



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

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

(11)



1-0052569

(51)^{2020.01} H04N 19/157; H04N 19/117; H04N
19/70; H04N 19/176; H04N 19/593;
H04N 19/11

(13) B

(21) 1-2021-07708

(22) 04/05/2020

(86) PCT/KR2020/005830 04/05/2020

(87) WO2020/222588 05/11/2020

(30) 10-2019-0050960 30/04/2019 KR; 10-2019-0057185 15/05/2019 KR; 10-2019-0057650 17/05/2019 KR

(45) 27/10/2025 451

(43) 25/02/2022 407A

(73) WILUS INSTITUTE OF STANDARDS AND TECHNOLOGY INC. (KR)

5F 216 Hwangsaek-ro Bundang-gu Seongnam-si Gyeonggi-do 13595, Republic of Korea

(72) KO, Geonjung (KR); KIM, Dongcheol (KR); SON, Juhjung (KR); JUNG, Jaehong (KR); KWAK, Jinsam (KR).

(74) Công ty TNHH Tầm nhìn và Liên danh (VISION & ASSOCIATES CO.LTD.)

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ XỬ LÝ TÍN HIỆU VIDEO NHỜ SỬ DỤNG ĐỘ PHÂN GIẢI VECTƠ CHUYỂN ĐỘNG THÍCH ỨNG VÀ PHƯƠNG PHÁP TẠO RA DÒNG BIT

(21) 1-2021-07708

(57) Sáng chế đề cập đến phương pháp và thiết bị xử lý tín hiệu video và, cụ thể hơn là, bao gồm các bước: phân tích, từ dòng bit, cờ cho phép độ phân giải vectơ chuyển động thích ứng (adaptive motion vector resolution, AMVR) (sps_amvr_enabled_flag) biểu thị xem độ phân giải vi sai vectơ chuyển động thích ứng có được sử dụng hay không; phân tích, từ dòng bit, cờ cho phép afin (sps_affine_enabled_flag) biểu thị xem phép bù chuyển động afin có thể sử dụng hay không; trên cơ sở cờ cho phép afin, xác định xem phép bù chuyển động afin có thể sử dụng hay không; khi phép bù chuyển động afin có thể sử dụng, xác định, trên cơ sở cờ cho phép AMVR, xem độ phân giải vi sai vectơ chuyển động thích ứng có được sử dụng hay không; và, khi độ phân giải vi sai vectơ chuyển động thích ứng được sử dụng, phân tích, từ dòng bit, cờ cho phép AMVR afin (sps_affine_amvr_enabled_flag) biểu thị xem độ phân giải vi sai vectơ chuyển động thích ứng có thể sử dụng cho phép bù chuyển động afin hay không.

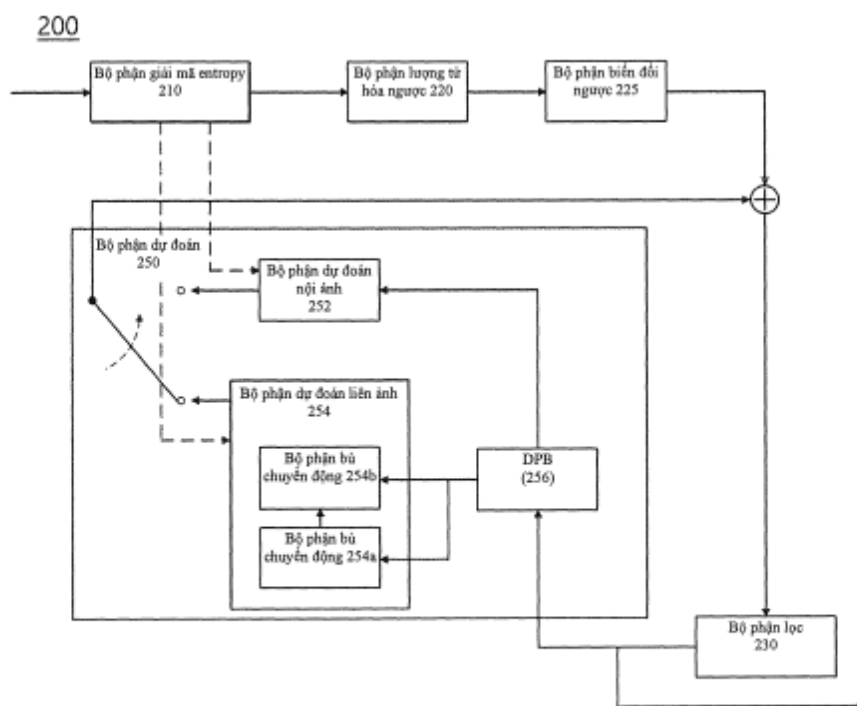


FIG. 2

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp và thiết bị xử lý tín hiệu video và, cụ thể hơn là, đề cập đến phương pháp xử lý tín hiệu video và thiết bị mã hóa và giải mã tín hiệu video.

Tình trạng kỹ thuật của sáng chế

Sự mã hóa nén đề cập đến một loạt các kỹ thuật xử lý tín hiệu để truyền thông tin được số hóa qua đường truyền thông hoặc lưu trữ thông tin dưới dạng thích hợp với vật ghi lưu trữ. Đối tượng của mã hóa nén bao gồm các đối tượng chẳng hạn như âm thanh, video và văn bản, và cụ thể là, kỹ thuật thực hiện việc mã hóa nén đối với hình ảnh được gọi là sự nén video. Việc mã hóa nén đối với tín hiệu video được thực hiện bằng cách loại bỏ thông tin dư thừa xét theo độ tương quan về không gian, độ tương quan về thời gian và độ tương quan ngẫu nhiên. Tuy nhiên, với sự phát triển gần đây của các phương tiện và phương tiện truyền dữ liệu khác nhau, cần có phương pháp và thiết bị xử lý tín hiệu video hiệu quả hơn.

Bản chất kỹ thuật của sáng chế

Vấn đề kỹ thuật

Mục đích của sáng chế là tăng hiệu quả mã hóa của tín hiệu video.

Giải pháp cho vấn đề

Phương pháp xử lý tín hiệu video theo một phương án của sáng chế bao gồm các bước: phân tích, từ dòng bit, cờ cho phép độ phân giải vector chuyển động thích ứng (adaptive motion vector resolution, AMVR) `sps_amvr_enabled_flag` biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được sử dụng hay không, phân tích, từ dòng bit, cờ cho phép afin `sps_affine_enabled_flag` biểu thị xem phép bù chuyển động afin có thể sử dụng hay không, trên cơ sở cờ cho phép afin `sps_affine_enabled_flag`, xác định xem phép bù chuyển động afin có thể sử dụng hay không, khi phép bù chuyển động afin có thể sử dụng, xác định, trên cơ sở cờ cho phép AMVR `sps_amvr_enabled_flag`, xem độ phân giải vi sai vector chuyển động thích ứng có được sử dụng hay không, và khi độ phân giải vi sai vector chuyển động thích ứng được sử dụng, phân tích, từ dòng bit, cờ cho phép AMVR afin `sps_affine_amvr_enabled_flag` biểu thị xem độ phân giải vi sai vector chuyển động thích

ứng có thể sử dụng cho phép bù chuyển động afin hay không.

Theo phương pháp xử lý tín hiệu video theo một phương án của sáng chế, một trong số cờ cho phép AMVR `sps_amvr_enabled_flag`, cờ cho phép afin `sps_affine_enabled_flag` hoặc cờ cho phép AMVR afin `sps_affine_amvr_enabled_flag` được báo hiệu dưới dạng một trong số đơn vị cây mã hóa, phân chia, tám, nhóm tám, ảnh hoặc đơn vị chuỗi.

Theo phương pháp xử lý tín hiệu video theo một phương án của sáng chế, khi phép bù chuyển động afin có thể sử dụng và độ phân giải vi sai vector chuyển động thích ứng không được sử dụng, cờ cho phép AMVR afin `sps_affine_amvr_enabled_flag` suy ra rằng độ phân giải vi sai vector chuyển động thích ứng là không thể sử dụng cho phép bù chuyển động afin.

Theo phương pháp xử lý tín hiệu video theo một phương án của sáng chế, khi phép bù chuyển động afin là không thể sử dụng, thì cờ cho phép AMVR afin `sps_affine_amvr_enabled_flag` suy ra rằng độ phân giải vi sai vector chuyển động thích ứng là không thể sử dụng cho phép bù chuyển động afin.

Phương pháp xử lý tín hiệu video theo một phương án của sáng chế còn bao gồm, khi cờ cho phép AMVR `sps_amvr_enabled_flag` biểu thị có sử dụng độ phân giải vi sai vector chuyển động thích ứng, cờ afin liên ảnh `inter_affine_flag` được thu từ dòng bit biểu thị rằng phép bù chuyển động afin không được sử dụng cho khối hiện thời, và ít nhất một trong số các chênh lệch vector chuyển động cho khối hiện thời là khác không, và bước phân tích thông tin về độ phân giải của chênh lệch vector chuyển động từ dòng bit, và bước sửa đổi các chênh lệch vector chuyển động cho khối hiện thời trên cơ sở thông tin về độ phân giải của chênh lệch vector chuyển động.

Phương pháp xử lý tín hiệu video theo một phương án của sáng chế còn bao gồm, khi cờ cho phép AMVR afin biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng có thể sử dụng cho phép bù chuyển động afin, cờ afin liên ảnh `inter_affine_flag` được thu từ dòng bit biểu thị có sử dụng phép bù chuyển động afin cho khối hiện thời, và ít nhất một trong số các chênh lệch vector chuyển động điểm điều khiển đối với khối hiện thời là khác không, bước phân tích thông tin về độ phân giải của chênh lệch vector chuyển động từ dòng bit, và bước sửa đổi các chênh lệch vector chuyển động điểm điều khiển đối với khối hiện thời trên cơ sở thông tin về độ phân giải của chênh lệch vector chuyển động.

Phương pháp xử lý tín hiệu video theo một phương án của sáng chế còn bao gồm,

bước thu thông tin `inter_pred_idc` về danh mục ảnh tham chiếu cho khối hiện thời, khi thông tin `inter_pred_idc` về danh mục ảnh tham chiếu biểu thị rằng chỉ danh mục ảnh tham chiếu thứ không không được sử dụng, bước phân tích chỉ số dự đoán vector chuyển động `mvp_11_flag` của danh mục ảnh tham chiếu thứ nhất từ dòng bit, tạo ra các ứng viên dự đoán vector chuyển động, thu bộ dự đoán vector chuyển động từ các ứng viên dự đoán vector chuyển động trên cơ sở chỉ số dự đoán vector chuyển động, và dự đoán khối hiện thời trên cơ sở bộ dự đoán vector chuyển động.

Phương pháp xử lý tín hiệu video theo một phương án của sáng chế còn bao gồm, bước thu, từ dòng bit, cờ chênh lệch vector chuyển động bằng không `mvd_11_zero_flag` biểu thị xem chênh lệch vector chuyển động và các chênh lệch vector chuyển động điểm điều khiển có được thiết lập thành không đối với danh mục ảnh tham chiếu thứ nhất hay không, trong đó bước phân tích chỉ số dự đoán vector chuyển động `mvp_11_flag` bao gồm, cờ chênh lệch vector chuyển động bằng không `mvd_11_zero_flag` là 1 và bất kể việc liệu thông tin `inter_pred_idc` về danh mục ảnh tham chiếu biểu thị rằng cả hai danh mục ảnh tham chiếu thứ không và danh mục ảnh tham chiếu thứ nhất có được sử dụng hay không, bước phân tích chỉ số dự đoán vector chuyển động `mvp_11_flag`.

Phương pháp xử lý tín hiệu video theo một phương án của sáng chế bao gồm bước phân tích, từ dòng bit, thông tin thứ nhất `six_minus_max_num_merge_cand` liên quan đến số lượng ứng viên tối đa dùng cho phép dự đoán vector chuyển động hợp nhất theo các đơn vị của các chuỗi, bước thu số lượng ứng viên hợp nhất tối đa trên cơ sở thông tin thứ nhất, bước phân tích, từ dòng bit, thông tin thứ hai biểu thị xem khối có được phân chia cho phép dự đoán liên ảnh hay không, và, khi thông tin thứ hai biểu thị 1 và số lượng ứng viên hợp nhất tối đa lớn hơn 2, bước phân tích, từ dòng bit, thông tin thứ ba liên quan đến số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia.

Phương pháp xử lý tín hiệu video theo một phương án của sáng chế còn bao gồm, khi thông tin thứ hai biểu thị 1 và số lượng ứng viên hợp nhất tối đa lớn hơn hoặc bằng 3, bước thu số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia bằng cách lấy số lượng ứng viên hợp nhất tối đa trừ thông tin thứ ba, khi thông tin thứ hai biểu thị 1 và số lượng ứng viên hợp nhất tối đa là 2, bước thiết lập số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia thành 2, và, khi thông tin thứ hai là 0 hoặc số lượng ứng viên hợp nhất tối đa là 1, thiết lập số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia thành 0.

Thiết bị xử lý tín hiệu video theo một phương án của sáng chế bao gồm bộ xử lý và bộ nhớ, trong đó, trên cơ sở các lệnh được lưu trữ trong bộ nhớ, bộ xử lý phân tích, từ dòng bit, cờ cho phép độ phân giải vector chuyển động thích ứng (AMVR) `sps_amvr_enabled_flag` biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được sử dụng hay không, phân tích, từ dòng bit, cờ cho phép afin `sps_affine_enabled_flag` biểu thị xem phép bù chuyển động afin có thể sử dụng hay không, trên cơ sở cờ cho phép afin `sps_affine_enabled_flag`, xác định xem phép bù chuyển động afin có thể sử dụng hay không, khi phép bù chuyển động afin có thể sử dụng, xác định, trên cơ sở cờ cho phép AMVR `sps_amvr_enabled_flag`, xem độ phân giải vi sai vector chuyển động thích ứng có được sử dụng hay không, và, khi độ phân giải vi sai vector chuyển động thích ứng được sử dụng, phân tích, từ dòng bit, cờ cho phép AMVR afin `sps_affine_amvr_enabled_flag` biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có thể sử dụng cho phép bù chuyển động afin hay không.

Trong thiết bị xử lý tín hiệu video theo một phương án của sáng chế, một trong số cờ cho phép AMVR `sps_amvr_enabled_flag`, cờ cho phép afin `sps_affine_enabled_flag` hoặc cờ cho phép AMVR afin `sps_affine_amvr_enabled_flag` được báo hiệu dưới dạng một trong số đơn vị cây mã hóa, phân chia, tám, nhóm tám, ảnh hoặc đơn vị chuỗi.

Trong thiết bị xử lý tín hiệu video theo một phương án của sáng chế, khi phép bù chuyển động afin có thể sử dụng và độ phân giải vi sai vector chuyển động thích ứng không được sử dụng, cờ cho phép AMVR afin `sps_affine_amvr_enabled_flag` suy ra rằng độ phân giải vi sai vector chuyển động thích ứng là không thể sử dụng cho phép bù chuyển động afin.

Trong thiết bị xử lý tín hiệu video theo một phương án của sáng chế, khi phép bù chuyển động afin là không thể sử dụng, thì cờ cho phép AMVR afin `sps_affine_amvr_enabled_flag` suy ra rằng độ phân giải vi sai vector chuyển động thích ứng là không thể sử dụng cho phép bù chuyển động afin.

Trong thiết bị xử lý tín hiệu video theo một phương án của sáng chế, trên cơ sở các lệnh được lưu trữ trong bộ nhớ, bộ xử lý, khi cờ cho phép AMVR `sps_amvr_enabled_flag` biểu thị có sử dụng độ phân giải vi sai vector chuyển động thích ứng, cờ afin liên ảnh `inter_affine_flag` được thu từ dòng bit biểu thị rằng phép bù chuyển động afin không được sử dụng cho khối hiện thời, và ít nhất một trong số các chênh lệch vector chuyển động cho khối hiện thời là khác không, phân tích thông tin về độ phân giải của chênh lệch vector chuyển động từ dòng bit, và sửa đổi các chênh lệch vector chuyển động cho khối hiện thời

trên cơ sở thông tin về độ phân giải của chèn lệch vector chuyển động.

Trong thiết bị xử lý tín hiệu video theo một phương án của sáng chế, trên cơ sở các lệnh được lưu trữ trong bộ nhớ, bộ xử lý, khi chờ cho phép AMVR afin biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng có thể sử dụng cho phép bù chuyển động afin, chờ afin liên ảnh `inter_affine_flag` được thu từ dòng bit biểu thị có sử dụng phép bù chuyển động afin cho khối hiện thời, và ít nhất một trong số các chèn lệch vector chuyển động điểm điều khiển đối với khối hiện thời là khác không, phân tích thông tin về độ phân giải của chèn lệch vector chuyển động từ dòng bit, và sửa đổi các chèn lệch vector chuyển động điểm điều khiển đối với khối hiện thời trên cơ sở thông tin về độ phân giải của chèn lệch vector chuyển động.

Trong thiết bị xử lý tín hiệu video theo một phương án của sáng chế, trên cơ sở các lệnh được lưu trữ trong bộ nhớ, bộ xử lý thu thông tin `inter_pred_idc` về danh mục ảnh tham chiếu cho khối hiện thời, khi thông tin `inter_pred_idc` về danh mục ảnh tham chiếu biểu thị rằng chỉ danh mục ảnh tham chiếu thứ không list 0 không được sử dụng, phân tích chỉ số dự đoán vector chuyển động `mvp_11_flag` của danh mục ảnh tham chiếu thứ nhất list 1 từ dòng bit, tạo ra các ứng viên dự đoán vector chuyển động, thu bộ dự đoán vector chuyển động từ các ứng viên dự đoán vector chuyển động trên cơ sở chỉ số dự đoán vector chuyển động, và dự đoán khối hiện thời trên cơ sở bộ dự đoán vector chuyển động.

Trong thiết bị xử lý tín hiệu video theo một phương án của sáng chế, trên cơ sở các lệnh được lưu trữ trong bộ nhớ, bộ xử lý thu, từ dòng bit, cờ chèn lệch vector chuyển động bằng không `mvd_11_zero_flag` biểu thị xem chèn lệch vector chuyển động và các chèn lệch vector chuyển động điểm điều khiển có được thiết lập thành không đối với danh mục ảnh tham chiếu thứ nhất hay không, và cờ chèn lệch vector chuyển động bằng không `mvd_11_zero_flag` là 1 và bất kể việc liệu thông tin `inter_pred_idc` về danh mục ảnh tham chiếu biểu thị rằng cả hai danh mục ảnh tham chiếu thứ không và danh mục ảnh tham chiếu thứ nhất có được sử dụng hay không, phân tích chỉ số dự đoán vector chuyển động `mvp_11_flag`.

Thiết bị xử lý tín hiệu video theo một phương án của sáng chế bao gồm bộ xử lý và bộ nhớ, trong đó, trên cơ sở các lệnh được lưu trữ trong bộ nhớ, bộ xử lý phân tích, từ dòng bit, thông tin thứ nhất `six_minus_max_num_merge_cand` liên quan đến số lượng ứng viên tối đa dùng cho phép dự đoán vector chuyển động hợp nhất theo các đơn vị của các chuỗi, thu số lượng ứng viên hợp nhất tối đa trên cơ sở thông tin thứ nhất, phân tích, từ dòng bit,

thông tin thứ hai biểu thị xem khối có được phân chia cho phép dự đoán liên ảnh hay không, và, khi thông tin thứ hai biểu thị 1 và số lượng ứng viên hợp nhất tối đa lớn hơn 2, phân tích, từ dòng bit, thông tin thứ ba liên quan đến số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia.

Trong thiết bị xử lý tín hiệu video theo một phương án của sáng chế, trên cơ sở các lệnh được lưu trữ trong bộ nhớ, bộ xử lý, khi thông tin thứ hai biểu thị 1 và số lượng ứng viên hợp nhất tối đa lớn hơn hoặc bằng 3, thu số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia bằng cách lấy số lượng ứng viên hợp nhất tối đa trừ thông tin thứ ba, khi thông tin thứ hai biểu thị 1 và số lượng ứng viên hợp nhất tối đa là 2, thiết lập số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia thành 2, và, khi thông tin thứ hai là 0 hoặc số lượng ứng viên hợp nhất tối đa là 1, thiết lập số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia thành 0.

Phương pháp xử lý tín hiệu video theo một phương án của sáng chế bao gồm các bước: tạo ra cờ cho phép độ phân giải vector chuyển động thích ứng (AMVR) `sps_amvr_enabled_flag` biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được sử dụng hay không, tạo ra cờ cho phép afin `sps_affine_enabled_flag` biểu thị xem phép bù chuyển động afin có thể sử dụng hay không, trên cơ sở cờ cho phép afin `sps_affine_enabled_flag`, xác định xem phép bù chuyển động afin có thể sử dụng hay không, khi phép bù chuyển động afin có thể sử dụng, xác định, trên cơ sở cờ cho phép AMVR `sps_amvr_enabled_flag`, xem độ phân giải vi sai vector chuyển động thích ứng có được sử dụng hay không, khi độ phân giải vi sai vector chuyển động thích ứng được sử dụng, tạo ra cờ cho phép AMVR afin `sps_affine_amvr_enabled_flag` biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có thể sử dụng cho phép bù chuyển động afin hay không, và tạo ra dòng bit bằng cách thực hiện việc mã hóa entropy cờ cho phép AMVR `sps_amvr_enabled_flag`, cờ cho phép afin `sps_affine_enabled_flag` và cờ cho phép AMVR `sps_amvr_enabled_flag`.

Phương pháp xử lý tín hiệu video theo một phương án của sáng chế còn bao gồm các bước: tạo ra, trên cơ sở số lượng ứng viên hợp nhất tối đa, thông tin thứ nhất `six_minus_max_num_merge_cand` liên quan đến số lượng ứng viên tối đa dùng cho phép dự đoán vector chuyển động hợp nhất, tạo ra thông tin thứ hai biểu thị xem khối có thể được phân chia cho phép dự đoán liên ảnh hay không, khi thông tin thứ hai biểu thị 1 và số lượng ứng viên hợp nhất tối đa lớn hơn 2, tạo ra thông tin thứ ba liên quan đến số lượng ứng viên

chế độ hợp nhất tối đa cho khối được phân chia, và thực hiện việc mã hóa entropy thông tin thứ nhất `six_minus_max_num_merge_cand`, thông tin thứ hai và thông tin thứ ba để tạo ra dòng bit theo các đơn vị của các chuỗi.

Hiệu quả đạt được của sáng chế

Theo một phương án của sáng chế, hiệu quả mã hóa của tín hiệu video có thể được tăng lên.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ khối giản lược của thiết bị mã hóa tín hiệu video theo một phương án của sáng chế.

Fig.2 là sơ đồ khối giản lược của thiết bị giải mã tín hiệu video theo một phương án của sáng chế.

Fig.3 là sơ đồ minh họa một phương án của sáng chế để phân chia đơn vị mã hóa.

Fig.4 là sơ đồ minh họa một phương án của phương pháp biểu diễn phân cấp cấu trúc phân chia của Fig.3.

Fig.5 là sơ đồ minh họa một phương án khác của sáng chế để phân chia đơn vị mã hóa.

Fig.6 là sơ đồ minh họa phương pháp thu điểm ảnh tham chiếu cho phép dự đoán nội ảnh.

Fig.7 là sơ đồ minh họa một phương án của các chế độ dự đoán được sử dụng cho phép dự đoán nội ảnh.

Fig.8 là sơ đồ minh họa phép dự đoán liên ảnh theo một phương án của sáng chế.

Fig.9 là sơ đồ minh họa phương pháp báo hiệu vector chuyển động theo một phương án của sáng chế.

Fig.10 là sơ đồ minh họa cú pháp chênh lệch vector chuyển động theo một phương án của sáng chế.

Fig.11 là sơ đồ minh họa báo hiệu độ phân giải vector chuyển động thích ứng theo một phương án của sáng chế.

Fig.12 là sơ đồ minh họa báo hiệu độ phân giải vector chuyển động thích ứng theo một phương án của sáng chế

Fig.13 là sơ đồ minh họa báo hiệu độ phân giải vector chuyển động thích ứng theo một phương án của sáng chế.

Fig.14 là sơ đồ minh họa báo hiệu độ phân giải vector chuyển động thích ứng theo một phương án của sáng chế.

Fig.15 là sơ đồ minh họa báo hiệu độ phân giải vector chuyển động thích ứng theo một phương án của sáng chế.

Fig.16 là sơ đồ minh họa báo hiệu độ phân giải vector chuyển động thích ứng theo một phương án của sáng chế.

Fig.17 là sơ đồ minh họa phép dự đoán chuyển động affin theo một phương án của sáng chế.

Fig.18 là sơ đồ minh họa phép dự đoán chuyển động affin theo một phương án của sáng chế.

Fig.19 là hình vẽ minh họa trường vector chuyển động theo một phương án của sáng chế.

Fig.20 là sơ đồ minh họa phép dự đoán chuyển động affin theo một phương án của sáng chế.

Fig.21 là hình vẽ minh họa trường vector chuyển động theo một phương án của sáng chế.

Fig.22 là sơ đồ minh họa phép dự đoán chuyển động affin theo một phương án của sáng chế.

Fig.23 là sơ đồ minh họa một chế độ của phép dự đoán chuyển động affin theo một phương án của sáng chế.

Fig.24 là sơ đồ minh họa một chế độ của phép dự đoán chuyển động affin theo một phương án của sáng chế.

Fig.25 là sơ đồ minh họa phép suy ra bộ dự đoán chuyển động affin theo một phương án của sáng chế.

Fig.26 là sơ đồ minh họa phép suy ra bộ dự đoán chuyển động affin theo một phương án của sáng chế.

Fig.27 là sơ đồ minh họa phép suy ra bộ dự đoán chuyển động affin theo một phương

án của sáng chế.

Fig.28 là sơ đồ minh họa phép suy ra bộ dự đoán chuyển động afin theo một phương án của sáng chế.

Fig.29 là sơ đồ minh họa phương pháp tạo ra vector chuyển động điểm điều khiển theo một phương án của sáng chế.

Fig.30 là sơ đồ minh họa phương pháp xác định chênh lệch vector chuyển động thông qua phương pháp được mô tả với tham chiếu đến Fig.29.

Fig.31 là sơ đồ minh họa phương pháp tạo ra vector chuyển động điểm điều khiển theo một phương án của sáng chế.

Fig.32 là sơ đồ minh họa phương pháp xác định chênh lệch vector chuyển động thông qua phương pháp được mô tả với tham chiếu đến Fig.31.

Fig.33 là sơ đồ minh họa cú pháp chênh lệch vector chuyển động theo một phương án của sáng chế.

Fig.34 là sơ đồ minh họa cấu trúc báo hiệu mức cao hơn theo một phương án của sáng chế.

Fig.35 là sơ đồ minh họa cấu trúc cú pháp đơn vị mã hóa theo một phương án của sáng chế.

Fig.36 là sơ đồ minh họa cấu trúc báo hiệu mức cao hơn theo một phương án của sáng chế.

Fig.37 là sơ đồ minh họa cấu trúc báo hiệu mức cao hơn theo một phương án của sáng chế.

Fig.38 là sơ đồ minh họa cấu trúc cú pháp đơn vị mã hóa theo một phương án của sáng chế.

Fig.39 là sơ đồ minh họa thiết lập giá trị mặc định MVD theo một phương án của sáng chế

Fig.40 là sơ đồ minh họa thiết lập mặc định MVD theo một phương án của sáng chế.

Fig.41 là sơ đồ minh họa cấu trúc cú pháp liên quan đến AMVR theo một phương án của sáng chế.

Fig.42 là sơ đồ minh họa cấu trúc cú pháp liên quan đến phép dự đoán liên ảnh theo

một phương án của sáng chế.

Fig.43 là sơ đồ minh họa cấu trúc cú pháp liên quan đến phép dự đoán liên ảnh theo một phương án của sáng chế.

Fig.44 là sơ đồ minh họa cấu trúc cú pháp liên quan đến phép dự đoán liên ảnh theo một phương án của sáng chế.

Fig.45 là sơ đồ minh họa cú pháp liên quan đến phép dự đoán liên ảnh theo một phương án của sáng chế.

Fig.46 là sơ đồ minh họa chế độ phân chia hình tam giác theo một phương án của sáng chế.

Fig.47 là sơ đồ minh họa cú pháp dữ liệu hợp nhất theo một phương án của sáng chế.

Fig.48 là sơ đồ minh họa báo hiệu mức cao hơn theo một phương án của sáng chế.

Fig.49 là sơ đồ minh họa số lượng ứng viên tối đa được sử dụng trong TPM theo một phương án của sáng chế.

Fig.50 là sơ đồ minh họa báo hiệu mức cao hơn liên quan đến TPM theo một phương án của sáng chế.

Fig.51 là sơ đồ minh họa số lượng ứng viên tối đa được sử dụng trong TPM theo một phương án của sáng chế.

Fig.52 là sơ đồ các phần tử cú pháp liên quan đến TPM theo một phương án của sáng chế.

Fig.53 là sơ đồ minh họa báo hiệu chỉ số ứng viên TPM theo một phương án của sáng chế.

Fig.54 là sơ đồ minh họa báo hiệu chỉ số ứng viên TPM theo một phương án của sáng chế.

Fig.55 là sơ đồ minh họa báo hiệu chỉ số ứng viên TPM theo một phương án của sáng chế.

Mô tả chi tiết sáng chế

Các thuật ngữ được sử dụng trong bản mô tả này có thể là các thuật ngữ chung hiện được sử dụng rộng rãi xét theo các chức năng theo sáng chế nhưng có thể thay đổi theo các mục đích của những người có hiểu biết trung bình trong lĩnh vực kỹ thuật, các tùy chỉnh,

hoặc sự xuất hiện của công nghệ mới. Ngoài ra, trong các trường hợp nhất định, có thể có các thuật ngữ mà chủ đơn lựa chọn một cách tùy ý và, trong trường hợp này, nghĩa của chúng được mô tả trong phần mô tả tương ứng của sáng chế. Theo đó, các thuật ngữ được sử dụng trong bản mô tả này nên được hiểu dựa trên các ý nghĩa quan trọng của các thuật ngữ và các nội dung trong toàn bộ bản mô tả.

Theo sáng chế, các thuật ngữ sau có thể được hiểu dựa trên các tiêu chí sau, và thậm chí các thuật ngữ không được mô tả có thể được hiểu theo mục đích sau. Sự mã hóa có thể được hiểu là mã hóa hoặc giải mã trong một số trường hợp, thông tin là thuật ngữ bao gồm tất cả các giá trị, các thông số, các hệ số, các phần tử, v.v. và ý nghĩa của các thuật ngữ này có thể được hiểu theo cách khác trong một số trường hợp, và do đó, sáng chế không bị giới hạn ở đó. ‘Đơn vị’ được sử dụng để tham chiếu đến đơn vị cơ sở để xử lý hình ảnh (ảnh) hoặc vị trí cụ thể của ảnh, và có thể được sử dụng một cách thay thế lẫn nhau với các thuật ngữ chẳng hạn như ‘khối’, ‘phân vùng’ hoặc ‘vùng’ trong một số trường hợp. Ngoài ra, trong bản mô tả hiện tại, đơn vị có thể được sử dụng làm khái niệm bao gồm tất cả đơn vị trong số đơn vị mã hóa, đơn vị dự đoán, và đơn vị biến đổi.

Fig.1 là sơ đồ khối giản lược của thiết bị mã hóa tín hiệu video theo một phương án của sáng chế. Dựa vào Fig.1, thiết bị mã hóa 100 của sáng chế chủ yếu bao gồm bộ phận biến đổi 110, bộ phận lượng tử hóa 115, bộ phận lượng tử hóa ngược 120, bộ phận biến đổi ngược 125, bộ phận lọc 130 và bộ phận dự đoán 150 và bộ phận mã hóa entropy 160.

Bộ phận biến đổi 110 thu giá trị hệ số biến đổi bằng cách biến đổi giá trị điểm ảnh của tín hiệu video nhận được. Ví dụ, phép biến đổi cosin rời rạc (Discrete Cosine Transform, DCT) hoặc phép biến đổi wavelet có thể được sử dụng. Cụ thể là, trong phép biến đổi cosin rời rạc, việc biến đổi được thực hiện bằng cách phân chia tín hiệu ảnh đầu vào thành các khối có kích thước định trước. Trong phép biến đổi, hiệu quả mã hóa có thể thay đổi theo sự phân bố và các đặc điểm của các giá trị ở vùng biến đổi.

Bộ phận lượng tử hóa 115 lượng tử hóa đầu ra giá trị hệ số biến đổi từ bộ phận biến đổi 110. Bộ phận lượng tử hóa ngược 120 giải lượng tử hóa giá trị hệ số biến đổi, và bộ phận biến đổi ngược 125 khôi phục giá trị điểm ảnh ban đầu nhờ sử dụng giá trị hệ số biến đổi được giải lượng tử hóa.

Bộ phận lọc 130 thực hiện phép tính lọc để cải thiện chất lượng của ảnh được khôi phục. Ví dụ, bộ lọc giải khối và bộ lọc vòng thích ứng có thể được bao gồm. Ảnh được lọc được kết xuất hoặc lưu trữ trong bộ đệm ảnh được giải mã 156 để được sử dụng làm ảnh

tham chiếu.

Để cải thiện hiệu quả mã hóa, tín hiệu ảnh không được mã hóa như thông thường, mà thay vào đó là phương pháp dự đoán ảnh bằng bộ phận dự đoán 150 bằng cách sử dụng vùng mà đã được mã hóa, và thêm, vào ảnh được dự đoán, giá trị dư giữa ảnh gốc và ảnh được dự đoán, nhờ đó thu được ảnh được khôi phục, được sử dụng. Bộ phận dự đoán nội ảnh 152 thực hiện phép dự đoán nội ảnh trong ảnh hiện thời, và bộ phận dự đoán liên ảnh 154 dự đoán ảnh hiện thời bằng cách sử dụng ảnh tham chiếu được lưu trữ trong bộ đệm ảnh giải mã 156. Bộ phận dự đoán nội ảnh 152 thực hiện phép dự đoán nội ảnh từ các vùng được khôi phục trong ảnh hiện thời, và truyền thông tin mã hóa nội ảnh đến bộ phận mã hóa entropy 160. Bộ phận dự đoán liên ảnh 154 có thể bao gồm bộ phận đánh giá chuyển động 154a và bộ phận bù chuyển động 154b. Bộ phận đánh giá chuyển động 154a thu giá trị vectơ chuyển động của vùng hiện thời bằng cách tham chiếu đến vùng được khôi phục cụ thể. Bộ phận đánh giá chuyển động 154a truyền thông tin vị trí (khung tham chiếu, vectơ chuyển động, v.v.) của vùng tham chiếu đến bộ phận mã hóa entropy 160 để cho phép thông tin vị trí được bao gồm trong dòng bit. Bộ phận bù chuyển động 154b thực hiện việc bù chuyển động liên ảnh bằng cách sử dụng giá trị vectơ chuyển động được truyền từ bộ phận đánh giá chuyển động 154a.

Bộ phận mã hóa entropy 160 thực hiện việc mã hóa entropy đối với hệ số biến đổi được lượng tử hóa, thông tin mã hóa liên ảnh, thông tin mã hóa nội ảnh, và thông tin vùng tham chiếu được nhập vào từ bộ phận dự đoán liên ảnh 154 để tạo ra dòng bit tín hiệu video. Ở đây, trong bộ phận mã hóa entropy 160, sơ đồ mã hóa chiều dài khả biến (variable length coding, VLC), mã hóa số học, v.v. có thể được sử dụng. Sơ đồ mã hóa chiều dài khả biến (VLC) biến đổi các ký hiệu đầu vào thành các từ mã liên tiếp, và chiều dài của các từ mã có thể khả biến. Ví dụ, các ký hiệu xuất hiện thường xuyên được biểu diễn dưới dạng các từ mã ngắn và các ký hiệu không xuất hiện thường xuyên được biểu diễn dưới dạng các từ mã dài. Là sơ đồ mã hóa chiều dài khả biến, sơ đồ mã hóa chiều dài khả biến thích ứng dựa vào ngữ cảnh (context-based adaptive variable length coding, CAVLC) có thể được sử dụng. Sự mã hóa số học biến đổi các ký hiệu dữ liệu liên tiếp thành một số nguyên tố, và bit phân số tối ưu cần thiết để biểu diễn mỗi ký hiệu có thể được thu khi mã hóa số học. Mã hóa số học nhị phân thích ứng dựa vào ngữ cảnh (context-based adaptive binary arithmetic code, CABAC) có thể được sử dụng làm mã hóa số học.

Dòng bit được tạo ra được gói lại nhờ sử dụng đơn vị lớp trừu tượng mạng (network

abstraction layer, NAL) làm đơn vị cơ bản. Đơn vị NAL bao gồm đoạn phân chia được mã hóa, và đoạn phân chia bao gồm số lượng đơn vị cây mã hóa nguyên. Để giải mã dòng bit trong bộ giải mã video, dòng bit nên được chia thành các đơn vị NAL trước tiên, và sau đó mỗi đơn vị NAL được chia nên được giải mã.

Fig.2 là sơ đồ khối giản lược của thiết bị giải mã tín hiệu video 200 theo một phương án của sáng chế. Dựa vào Fig.2, thiết bị giải mã 200 của sáng chế bao gồm bộ phận giải mã entropy 210, bộ phận lượng tử hóa ngược 220, bộ phận biến đổi ngược 225, bộ phận lọc 230 và bộ phận dự đoán 250.

Bộ phận giải mã entropy 210 thực hiện việc giải mã entropy đối với dòng bit tín hiệu video để trích xuất hệ số biến đổi và thông tin chuyển động đối với mỗi vùng. Bộ phận lượng tử hóa ngược 220 giải lượng tử hóa hệ số biến đổi được giải mã entropy, và bộ phận biến đổi ngược 225 khôi phục giá trị điểm ảnh ban đầu bằng cách sử dụng hệ số biến đổi được giải lượng tử hóa.

Trong khi đó, bộ phận lọc 230 cải thiện chất lượng ảnh bằng cách thực hiện việc lọc đối với ảnh. Trong bộ phận lọc này, bộ lọc giải khối để làm giảm độ biến dạng khối và/hoặc bộ lọc vòng thích ứng để loại bỏ độ biến dạng khối toàn bộ ảnh có thể được bao gồm. Ảnh được lọc được kết xuất hoặc lưu trữ trong bộ đệm ảnh được giải mã 256 để được sử dụng làm ảnh tham chiếu cho khung tiếp theo.

Ngoài ra, bộ phận dự đoán 250 của sáng chế bao gồm bộ phận dự đoán nội ảnh 252 và bộ phận dự đoán liên ảnh 254, và khôi phục ảnh dự đoán nhờ sử dụng kiểu mã hóa, hệ số biến đổi đối với mỗi vùng, thông tin chuyển động, v.v. được giải mã thông qua bộ phận giải mã entropy 210 được mô tả ở trên.

Về mặt này, bộ phận dự đoán nội ảnh 252 thực hiện phép dự đoán nội ảnh từ mẫu được giải mã trong ảnh hiện thời. Bộ phận dự đoán liên ảnh 254 tạo ra ảnh dự đoán bằng cách sử dụng ảnh tham chiếu được lưu trữ trong bộ đệm ảnh được giải mã 256 và thông tin chuyển động. Bộ phận dự đoán liên ảnh 254 có thể được tạo cấu hình lại để bao gồm bộ phận đánh giá chuyển động 254a và bộ phận bù chuyển động 254b. Bộ phận đánh giá chuyển động 254a thu vector chuyển động biểu thị mối tương quan vị trí giữa khối hiện thời và khối tham chiếu của ảnh tham chiếu được sử dụng để mã hóa, và truyền vector chuyển động thu được đến bộ phận bù chuyển động 254b.

Đầu ra dự đoán từ bộ phận dự đoán nội ảnh 252 hoặc bộ phận dự đoán liên ảnh 254

và đầu ra giá trị điểm ảnh từ bộ phận biến đổi ngược 225 được thêm vào để tạo ra khung video được khôi phục.

Sau đây, trong hoạt động của thiết bị mã hóa 100 và thiết bị giải mã 200, phương pháp phân chia đơn vị mã hóa và đơn vị dự đoán với tham chiếu đến các hình vẽ từ Fig.3 đến Fig.5 sẽ được mô tả.

Đơn vị mã hóa có nghĩa là đơn vị cơ sở để xử lý ảnh trong quy trình xử lý tín hiệu video được mô tả ở trên, ví dụ, trong quy trình dự đoán nội ảnh/liên ảnh, biến đổi, lượng tử hóa và/hoặc mã hóa entropy. Kích thước của đơn vị mã hóa được sử dụng để mã hóa một ảnh có thể không phải là hằng số. Đơn vị mã hóa có thể có hình dạng hình chữ nhật, và một đơn vị mã hóa có thể được phân chia thành một vài đơn vị mã hóa.

Fig.3 minh họa một phương án của sáng chế để phân chia đơn vị mã hóa. Ví dụ, một đơn vị mã hóa có kích thước $2N \times 2N$ có thể lại được phân chia thành bốn đơn vị mã hóa có kích thước $N \times N$. Việc phân chia đơn vị mã hóa này có thể được thực hiện một cách đệ quy, và không phải tất cả các đơn vị mã hóa cần được phân chia theo cùng dạng. Tuy nhiên, nhằm thuận tiện khi mã hóa và quy trình xử lý, có thể có các giới hạn về kích thước của đơn vị mã hóa tối đa và/hoặc kích thước của đơn vị mã hóa tối thiểu.

Đối với một đơn vị mã hóa, thông tin biểu thị xem đơn vị mã hóa tương ứng có được phân chia hay không có thể được lưu trữ. Fig.4 minh họa một phương án của phương pháp biểu diễn phân cấp cấu trúc phân chia của đơn vị mã hóa được minh họa trên Fig.3 nhờ sử dụng giá trị cờ. Giá trị '1' có thể được cấp phát cho thông tin khi đơn vị này được phân chia, và giá trị '0' có thể được cấp phát cho thông tin khi đơn vị này không được phân chia. Như được minh họa trên Fig.4, nếu giá trị cờ biểu thị xem liệu sự phân chia là 1 hay không, đơn vị mã hóa tương ứng với nút tương ứng lại được phân chia thành 4 đơn vị mã hóa. Nếu giá trị cờ là 0, đơn vị mã hóa không còn được phân chia và quy trình xử lý đối với đơn vị mã hóa có thể được thực hiện.

Cấu trúc của đơn vị mã hóa được mô tả ở trên có thể được biểu diễn nhờ sử dụng cấu trúc cây đệ quy. Nghĩa là, đơn vị mã hóa được phân chia thành các đơn vị mã hóa khác với một ảnh hoặc đơn vị mã hóa kích thước tối đa dưới dạng gốc có số lượng nút con bằng số lượng đơn vị mã hóa được phân chia. Do đó, đơn vị mã hóa mà không còn được phân chia trở thành nút lá. Giả sử rằng chỉ phân chia hình vuông là khả thi đối với một đơn vị mã hóa, vì một đơn vị mã hóa có thể được phân chia thành tối đa bốn đơn vị mã hóa khác, cây biểu diễn đơn vị mã hóa có thể là ở dạng cây tứ phân.

Trong bộ mã hóa, kích thước tối ưu của đơn vị mã hóa được chọn theo đặc điểm (ví dụ, độ phân giải) của ảnh video hoặc xét theo hiệu quả mã hóa, và thông tin về kích thước tối ưu của đơn vị mã hóa hoặc thông tin mà kích thước tối ưu của đơn vị mã hóa có thể được suy ra có thể được bao gồm trong dòng bit. Ví dụ, kích thước của đơn vị mã hóa lớn nhất và độ sâu tối đa của cây có thể được xác định. Trong trường hợp thực hiện phân chia hình vuông, vì chiều cao và chiều rộng của đơn vị mã hóa bằng một nửa chiều cao và chiều rộng của đơn vị mã hóa của nút cha mẹ, kích thước đơn vị mã hóa tối thiểu có thể được thu bằng cách sử dụng thông tin trên. Hoặc, ngược lại, kích thước đơn vị mã hóa tối thiểu và độ sâu tối đa của cây có thể được xác định trước và sử dụng, và kích thước của đơn vị mã hóa tối đa có thể được suy ra và sử dụng bằng cách sử dụng kích thước đơn vị mã hóa tối thiểu và độ sâu tối đa của cây. Vì kích thước của đơn vị được thay đổi theo bội số của 2 trong phân chia hình vuông, kích thước thực tế của đơn vị mã hóa được biểu diễn dưới dạng giá trị logarit với cơ số 2 để tăng hiệu quả truyền dẫn.

Bộ giải mã có thể thu thông tin biểu thị xem đơn vị mã hóa hiện thời có được phân chia hay không. Nếu thông tin này được thu (được truyền) chỉ trong điều kiện cụ thể, hiệu quả có thể được tăng lên. Ví dụ, vì điều kiện rằng đơn vị mã hóa hiện thời có thể được phân chia là trường hợp trong đó kích thước của đơn vị được thu bằng cách thêm kích thước đơn vị mã hóa hiện thời ở vị trí hiện thời nhỏ hơn kích thước của ảnh và kích thước đơn vị hiện thời lớn hơn so với kích thước đơn vị mã hóa tối thiểu được thiết lập trước, thông tin biểu thị xem đơn vị mã hóa hiện thời có được phân chia hay không có thể được thu chỉ trong trường hợp này.

Nếu thông tin trên biểu thị rằng đơn vị mã hóa được phân chia, thì kích thước của đơn vị mã hóa cần được phân chia trở nên bằng một nửa đơn vị mã hóa hiện thời, và được phân chia thành bốn đơn vị mã hóa vuông trên cơ sở vị trí xử lý hiện thời. Quy trình xử lý nêu trên có thể được lặp lại đối với mỗi trong số các đơn vị mã hóa được phân chia.

Fig.5 minh họa một phương án khác của sáng chế để phân chia đơn vị mã hóa. Theo phương án khác này của sáng chế, đơn vị mã hóa dạng cây tứ phân được mô tả ở trên có thể được phân chia hơn nữa thành cấu trúc cây nhị phân của sự phân chia theo chiều ngang hoặc phân chia theo chiều dọc. Nghĩa là, phân chia cây tứ phân hình vuông có thể được áp dụng vào đơn vị mã hóa gốc trước tiên, và phân chia cây nhị phân hình chữ nhật có thể còn được áp dụng ở các nút lá của cây tứ phân. Theo một phương án, phân chia cây nhị phân có thể là sự phân chia đối xứng theo chiều ngang hoặc sự phân chia đối xứng theo chiều

đọc, nhưng sáng chế không bị giới hạn ở đó.

Trong mỗi nút phân chia của cây nhị phân, cờ biểu thị kiểu phân chia (nghĩa là, phân chia theo chiều ngang hoặc phân chia theo chiều dọc) có thể còn được báo hiệu. Theo một phương án, khi giá trị của cờ là '0', phân chia theo chiều ngang có thể được ra lệnh, và khi giá trị của cờ là '1', phân chia theo chiều dọc có thể được ra lệnh.

Tuy nhiên, phương pháp phân chia đơn vị mã hóa theo phương án của sáng chế không bị giới hạn ở các phương pháp được mô tả ở trên và phân chia theo chiều ngang/chiều dọc không đối xứng, cây tam phân được phân chia thành ba đơn vị mã hóa hình chữ nhật, v.v. có thể được áp dụng vào đó.

Phép dự đoán ảnh (bù chuyển động) để mã hóa được thực hiện đối với đơn vị mã hóa mà không còn được phân chia (nghĩa là, nút lá của cây đơn vị mã hóa). Đơn vị cơ sở để thực hiện phép dự đoán này sau đây được gọi là đơn vị dự đoán hoặc khối dự đoán.

Sau đây, thuật ngữ “đơn vị” được sử dụng trong bản mô tả này có thể được sử dụng làm thuật ngữ thay thế cho đơn vị dự đoán, mà là đơn vị cơ sở để thực hiện phép dự đoán. Tuy nhiên, sáng chế không bị giới hạn ở đó, và thuật ngữ này có thể được hiểu một cách rộng hơn là khái niệm bao gồm đơn vị mã hóa.

Để khôi phục đơn vị hiện thời mà việc giải mã được thực hiện, ảnh hiện thời bao gồm đơn vị hiện thời hoặc phần được giải mã của các ảnh khác có thể được sử dụng. Ảnh (phần chia) mà chỉ ảnh hiện thời được sử dụng để khôi phục, nghĩa là, chỉ phép dự đoán nội ảnh được thực hiện, được gọi là ảnh nội ảnh hoặc ảnh (phần chia) I, và ảnh (phần chia) mà cả hai phép dự đoán nội ảnh và phép dự đoán liên ảnh có thể được thực hiện để khôi phục được gọi là ảnh (phần chia) liên ảnh. Ảnh (phần chia), mà sử dụng nhiều nhất một vector chuyển động và chỉ số tham chiếu để dự đoán mỗi đơn vị, trong số các ảnh (phần chia) liên ảnh được gọi là ảnh dự đoán hoặc ảnh (phần chia) P, và ảnh (phần chia), mà sử dụng tối đa hai vector chuyển động và các chỉ số tham chiếu để dự đoán mỗi đơn vị, trong số các ảnh (phần chia) liên ảnh được gọi là ảnh dự đoán theo hai hướng hoặc ảnh (phần chia) B.

Đơn vị dự đoán nội ảnh thực hiện phép dự đoán nội ảnh để dự đoán giá trị điểm ảnh của đơn vị đích từ các vùng được khôi phục trong ảnh hiện thời. Ví dụ, giá trị điểm ảnh của đơn vị hiện thời có thể được dự đoán từ các điểm ảnh được khôi phục của các đơn vị nằm ở bên trái và/hoặc căn giữa trên cùng so với đơn vị hiện thời. Trong trường hợp này, các đơn vị nằm ở bên trái của đơn vị hiện thời có thể bao gồm đơn vị bên trái liền kề với đơn

vị hiện thời, đơn vị trên cùng bên trái và đơn vị dưới cùng bên trái. Ngoài ra, các đơn vị nằm ở trên cùng của đơn vị hiện thời có thể bao gồm đơn vị trên cùng liền kề với đơn vị hiện thời, đơn vị trên cùng bên trái và đơn vị trên cùng bên phải.

Trong khi đó, bộ phận dự đoán liên ảnh thực hiện phép dự đoán liên ảnh để dự đoán giá trị điểm ảnh của đơn vị đích bằng cách sử dụng thông tin về các ảnh được khôi phục khác khác với ảnh hiện thời. Trong trường hợp này, ảnh được sử dụng cho phép dự đoán được gọi là ảnh tham chiếu. Vùng tham chiếu cần được sử dụng để dự đoán đơn vị hiện thời trong quy trình dự đoán liên ảnh có thể được biểu thị nhờ sử dụng chỉ số biểu thị ảnh tham chiếu bao gồm vùng tham chiếu tương ứng, thông tin vector chuyển động, v.v..

Phép dự đoán liên ảnh có thể bao gồm phép dự đoán L0, phép dự đoán L1 và phép dự đoán theo hai hướng. Phép dự đoán L0 có nghĩa là phép dự đoán sử dụng một ảnh tham chiếu được bao gồm trong L0 (danh mục ảnh tham chiếu thứ không), và phép dự đoán L1 có nghĩa là phép dự đoán sử dụng một ảnh tham chiếu được bao gồm trong L1 (danh mục ảnh tham chiếu thứ nhất). Để thực hiện điều này, một tập hợp thông tin chuyển động (ví dụ, vector chuyển động và chỉ số ảnh tham chiếu) có thể cần thiết. Theo phương pháp dự đoán theo hai hướng, tối đa hai vùng tham chiếu có thể được sử dụng, và hai vùng tham chiếu có thể tồn tại trong cùng ảnh tham chiếu hoặc trong các ảnh khác nhau, một cách tương ứng. Nghĩa là, theo phương pháp dự đoán theo hai hướng, tối đa hai tập hợp thông tin chuyển động (ví dụ, vector chuyển động và chỉ số ảnh tham chiếu) có thể được sử dụng, và hai vector chuyển động có thể tương ứng với cùng chỉ số ảnh tham chiếu hoặc các chỉ số ảnh tham chiếu khác nhau. Trong trường hợp này, các ảnh tham chiếu có thể được hiển thị (hoặc được kết xuất) cả trước và sau ảnh hiện thời về mặt thời gian.

Đơn vị tham chiếu của đơn vị hiện thời có thể được thu nhờ sử dụng vector chuyển động và chỉ số ảnh tham chiếu. Đơn vị tham chiếu tồn tại trong ảnh tham chiếu có chỉ số ảnh tham chiếu. Ngoài ra, giá trị điểm ảnh của đơn vị được quy định bởi vector chuyển động hoặc giá trị được nội suy có thể được sử dụng làm bộ dự đoán của đơn vị hiện thời. Đối với phép dự đoán chuyển động với độ chính xác điểm ảnh theo các đơn vị của các điểm ảnh con, ví dụ, bộ lọc nội suy 8 tap có thể được sử dụng cho tín hiệu độ sáng và bộ lọc nội suy 4 tap có thể được sử dụng cho tín hiệu sắc độ. Tuy nhiên, bộ lọc nội suy dùng cho phép dự đoán chuyển động theo các đơn vị của các điểm ảnh con không bị giới hạn ở đó. Như được mô tả ở trên, phép bù chuyển động để dự đoán cấu trúc của đơn vị hiện thời từ ảnh được giải mã trước đó được thực hiện nhờ sử dụng thông tin chuyển động.

Sau đây, phương pháp dự đoán nội ảnh theo một phương án của sáng chế sẽ được mô tả một cách chi tiết hơn với tham chiếu đến Fig.6 và Fig.7. Như được mô tả ở trên, đơn vị dự đoán nội ảnh dự đoán giá trị điểm ảnh của đơn vị hiện thời bằng cách sử dụng các điểm ảnh liền kề nằm ở bên trái và/hoặc trên cùng của đơn vị hiện thời làm các điểm ảnh tham chiếu.

Như được minh họa trên Fig.6, khi kích thước của đơn vị hiện thời là $N \times N$, các điểm ảnh tham chiếu có thể được thiết lập nhờ sử dụng tối đa $(4N + 1)$ điểm ảnh liền kề nằm ở bên trái và/hoặc trên cùng của đơn vị hiện thời. Khi ít nhất một số điểm ảnh liền kề cần được sử dụng làm các điểm ảnh tham chiếu chưa được khôi phục, đơn vị dự đoán nội ảnh có thể thu các điểm ảnh tham chiếu bằng cách thực hiện quy trình đệm mẫu tham chiếu theo quy tắc được thiết lập trước. Ngoài ra, đơn vị dự đoán nội ảnh có thể thực hiện quy trình lọc mẫu tham chiếu để làm giảm lỗi trong phép dự đoán nội ảnh. Nghĩa là, các điểm ảnh tham chiếu có thể được thu bằng cách thực hiện việc lọc đối với các điểm ảnh liền kề và/hoặc các điểm ảnh được thu bởi quy trình đệm mẫu tham chiếu. Đơn vị dự đoán nội ảnh dự đoán các điểm ảnh của đơn vị hiện thời nhờ sử dụng các điểm ảnh tham chiếu được thu theo cách này.

Fig.7 minh họa một phương án của các chế độ dự đoán được sử dụng cho phép dự đoán nội ảnh. Đối với phép dự đoán nội ảnh, thông tin chế độ dự đoán nội ảnh biểu thị hướng dự đoán nội ảnh có thể được báo hiệu. Khi đơn vị hiện thời là đơn vị dự đoán nội ảnh, thiết bị giải mã tín hiệu video trích xuất thông tin chế độ dự đoán nội ảnh của đơn vị hiện thời từ dòng bit. Đơn vị dự đoán nội ảnh của thiết bị giải mã tín hiệu video thực hiện phép dự đoán nội ảnh đối với đơn vị hiện thời trên cơ sở thông tin chế độ dự đoán nội ảnh được trích xuất.

Theo một phương án của sáng chế, chế độ dự đoán nội ảnh có thể bao gồm tổng 67 chế độ. Mỗi chế độ dự đoán nội ảnh có thể được ra lệnh thông qua chỉ số được thiết lập trước (nghĩa là, chỉ số chế độ nội ảnh). Ví dụ, như được minh họa trên Fig.7, chỉ số chế độ nội ảnh 0 có thể ra lệnh chế độ phẳng, chỉ số chế độ nội ảnh 1 có thể ra lệnh chế độ DC và các chỉ số chế độ nội ảnh từ 2 đến 66 có thể ra lệnh các chế độ định hướng khác nhau (nghĩa là, các chế độ góc), một cách tương ứng. Đơn vị dự đoán nội ảnh xác định các điểm ảnh tham chiếu và/hoặc các điểm ảnh tham chiếu được nội suy cần được sử dụng cho phép dự đoán nội ảnh của đơn vị hiện thời trên cơ sở thông tin chế độ dự đoán nội ảnh của đơn vị hiện thời. Khi chỉ số chế độ nội ảnh ra lệnh chế độ hướng cụ thể, điểm ảnh tham chiếu

tương ứng với hướng cụ thể từ điểm ảnh hiện thời của đơn vị hiện thời hoặc điểm ảnh tham chiếu được nội suy được sử dụng cho phép dự đoán của điểm ảnh hiện thời. Do đó, các tập hợp khác nhau của các điểm ảnh tham chiếu và/hoặc các điểm ảnh tham chiếu được nội suy có thể được sử dụng cho phép dự đoán nội ảnh theo chế độ dự đoán nội ảnh.

Sau khi phép dự đoán nội ảnh của đơn vị hiện thời được thực hiện nhờ sử dụng các điểm ảnh tham chiếu và thông tin chế độ dự đoán nội ảnh, thiết bị giải mã tín hiệu video khôi phục các giá trị điểm ảnh của đơn vị hiện thời bằng cách thêm tín hiệu dư của đơn vị hiện thời được thu từ bộ phận biến đổi ngược và bộ dự đoán nội ảnh của đơn vị hiện thời.

Fig.8 là sơ đồ minh họa phép dự đoán liên ảnh theo một phương án của sáng chế.

Như được mô tả ở trên, khi mã hóa hoặc giải mã ảnh hoặc khối hiện thời, ảnh hoặc khối này có thể được dự đoán từ ảnh hoặc khối khác. Nghĩa là, có thể mã hóa và giải mã ảnh hoặc khối hiện thời trên cơ sở độ tương đồng với ảnh hoặc khối khác. Ảnh hoặc khối hiện thời có thể được mã hóa và được giải mã với báo hiệu trong đó phần tương tự với ảnh hoặc khối khác được lược bỏ khỏi ảnh hoặc khối hiện thời, mà sẽ được mô tả hơn nữa dưới đây. Có thể tạo ra các dự đoán theo các đơn vị của các khối.

Dựa vào Fig.8, ảnh tham chiếu tồn tại ở bên trái và ảnh hiện thời tồn tại ở bên phải, và ảnh hiện thời hoặc một phần của ảnh hiện thời có thể được dự đoán nhờ sử dụng độ tương đồng với ảnh tham chiếu hoặc một phần của ảnh tham chiếu. Khi hình chữ nhật được biểu thị bởi đường nét liền trong ảnh hiện thời của Fig.8 là khối đang cần được mã hóa và giải mã, khối hiện thời có thể được dự đoán từ hình chữ nhật được biểu thị bởi đường chấm chấm trong ảnh tham chiếu. Trong trường hợp này, thông tin biểu thị khối (khối tham chiếu) cần được tham chiếu bởi khối hiện thời có thể tồn tại, mà có thể được báo hiệu trực tiếp hoặc có thể được xây dựng bởi sự hứa hẹn bất kỳ sẽ làm giảm tổng phí báo hiệu. Thông tin biểu thị khối cần được tham chiếu bởi khối hiện thời có thể bao gồm vector chuyển động. Vector chuyển động này có thể là vector biểu thị vị trí tương đối trong ảnh giữa khối hiện thời và khối tham chiếu. Dựa vào Fig.8, phần được biểu thị bởi đường chấm chấm của ảnh tham chiếu tồn tại, và vector biểu thị cách mà khối hiện thời có thể được di chuyển đến khối cần được tham chiếu trong ảnh tham chiếu có thể là vector chuyển động. Nghĩa là, khối xuất hiện khi khối hiện thời được di chuyển theo vector chuyển động có thể là phần được biểu thị bởi đường chấm chấm trong ảnh hiện thời của Fig.8, và phần được biểu thị bởi đường chấm chấm có thể có cùng vị trí trong ảnh hiện thời với vị trí của khối tham chiếu của ảnh tham chiếu.

Ngoài ra, thông tin biểu thị khối cần được tham chiếu bởi khối hiện thời có thể bao gồm thông tin biểu thị ảnh tham chiếu. Thông tin biểu thị ảnh tham chiếu có thể bao gồm danh mục ảnh tham chiếu và chỉ số ảnh tham chiếu. Danh mục ảnh tham chiếu là danh mục bao gồm các ảnh tham chiếu, và có thể sử dụng khối tham chiếu trong ảnh tham chiếu được bao gồm trong danh mục ảnh tham chiếu. Nghĩa là, có thể dự đoán khối hiện thời từ ảnh tham chiếu được bao gồm trong danh mục ảnh tham chiếu. Ngoài ra, chỉ số ảnh tham chiếu có thể là chỉ số để biểu thị ảnh tham chiếu cần được sử dụng.

Fig.9 là sơ đồ minh họa phương pháp báo hiệu vector chuyển động theo một phương án của sáng chế.

Theo một phương án của sáng chế, có thể tạo ra vector chuyển động MV dựa trên bộ dự đoán vector chuyển động MVP. Ví dụ, bộ dự đoán vector chuyển động có thể là vector chuyển động như được minh họa dưới đây.

$$MV = MVP$$

Là một ví dụ khác, vector chuyển động có thể dựa trên chênh lệch vector chuyển động (motion vector difference, MVD) như sau. Chênh lệch vector chuyển động MVD có thể được thêm vào bộ dự đoán vector chuyển động để biểu diễn vector chuyển động chính xác.

$$MV = MVP + MVD$$

Ngoài ra, trong mã hóa video, bộ mã hóa có thể truyền thông tin vector chuyển động được xác định đến bộ giải mã, và bộ giải mã có thể tạo ra vector chuyển động từ thông tin vector chuyển động nhận được và xác định khối dự đoán. Ví dụ, thông tin vector chuyển động có thể bao gồm thông tin về bộ dự đoán vector chuyển động và chênh lệch vector chuyển động. Trong trường hợp này, các thành phần của thông tin vector chuyển động có thể thay đổi theo chế độ. Ví dụ, ở chế độ hợp nhất, thông tin vector chuyển động có thể bao gồm thông tin về bộ dự đoán vector chuyển động, nhưng có thể không bao gồm chênh lệch vector chuyển động. Là một ví dụ khác, ở chế độ dự đoán vector chuyển động nâng cao (advanced motion vector prediction, AMVP), thông tin vector chuyển động có thể bao gồm thông tin về bộ dự đoán vector chuyển động và có thể bao gồm chênh lệch vector chuyển động.

Để xác định, truyền, và nhận thông tin về bộ dự đoán vector chuyển động, bộ mã hóa và bộ giải mã có thể tạo ra các ứng viên bộ dự đoán vector chuyển động (motion vector predictor, MVP) theo cùng cách. Ví dụ, bộ mã hóa và bộ giải mã có thể tạo ra cùng ứng

viên MVP (ứng viên bộ dự đoán vector chuyển động) theo cùng thứ tự. Và, bộ mã hóa có thể truyền chỉ số `mvp_lx_flag` biểu thị MVP (bộ dự đoán vector chuyển động) được xác định từ các ứng viên MVP được tạo ra (các ứng viên dự đoán vector chuyển động) đến bộ giải mã, và bộ giải mã có thể tìm ra MVP (bộ dự đoán vector chuyển động) và MV được xác định trên cơ sở chỉ số `mvp_lx_flag` này. Chỉ số `mvp_lx_flag` có thể bao gồm chỉ số dự đoán vector chuyển động `mvp_10_flag` của danh mục ảnh tham chiếu thứ không list 0 và chỉ số dự đoán vector chuyển động `mvp_11_flag` của danh mục ảnh tham chiếu thứ nhất list 1. Phương pháp nhận chỉ số `mvp_lx_flag` sẽ được mô tả với tham chiếu đến các hình vẽ từ Fig.56 đến Fig.59.

Ứng viên MVP và phương pháp tạo ra ứng viên MVP có thể bao gồm ứng viên không gian, ứng viên thời gian, v.v.. Ứng viên không gian có thể là vector chuyển động cho khối nằm ở vị trí định trước so với khối hiện thời. Ví dụ, ứng viên không gian có thể là vector chuyển động tương ứng với khối hoặc vị trí mà liền kề với hoặc không liền kề với khối hiện thời. Ứng viên thời gian có thể là vector chuyển động tương ứng với khối trong ảnh khác với ảnh hiện thời. Theo cách khác, ứng viên MVP có thể bao gồm vector chuyển động afin, ATMVP, STMVP, tổ hợp của các vector chuyển động được mô tả ở trên, vector trung bình của các vector chuyển động được mô tả ở trên, vector chuyển động bằng không, v.v..

Ngoài ra, thông tin biểu thị ảnh tham chiếu được mô tả ở trên cũng có thể được truyền từ bộ mã hóa đến bộ giải mã. Ngoài ra, khi ảnh tham chiếu tương ứng với ứng viên MVP không tương ứng với thông tin biểu thị ảnh tham chiếu, việc thay đổi tỷ lệ vector chuyển động có thể được thực hiện. Việc thay đổi tỷ lệ vector chuyển động có thể là phép tính dựa trên số đếm thứ tự ảnh (picture order count, POC) của ảnh hiện thời, POC của ảnh tham chiếu của khối hiện thời, POC của ảnh tham chiếu của ứng viên MVP và ứng viên MVP.

Fig.10 là sơ đồ minh họa cú pháp chên lệch vector chuyển động theo một phương án của sáng chế.

Chên lệch vector chuyển động có thể được mã hóa bằng cách phân chia dấu và giá trị tuyệt đối của chên lệch vector chuyển động. Nghĩa là, dấu và giá trị tuyệt đối của chên lệch vector chuyển động có thể là cú pháp khác. Ngoài ra, giá trị tuyệt đối của chên lệch vector chuyển động có thể được mã hóa trực tiếp, nhưng có thể được mã hóa bao gồm cờ biểu thị xem giá trị tuyệt đối có lớn hơn N như được minh họa trên Fig.10 hay không. Nếu giá trị tuyệt đối lớn hơn N, thì giá trị của (giá trị tuyệt đối - N) có thể được báo hiệu cùng

nhau. Trong ví dụ của Fig.10, `abs_mvd_greater0_flag` có thể được truyền, và cờ này có thể là cờ biểu thị xem giá trị tuyệt đối có lớn hơn 0 hay không. Nếu `abs_mvd_greater0_flag` được biểu thị rằng giá trị tuyệt đối không lớn hơn 0, có thể xác định được rằng giá trị tuyệt đối là 0. Ngoài ra, nếu `abs_mvd_greater0_flag` được biểu thị rằng giá trị tuyệt đối lớn hơn 0, cú pháp khác có thể tồn tại. Ví dụ, `abs_mvd_greater1_flag` có thể tồn tại, và cờ này có thể là cờ biểu thị xem giá trị tuyệt đối có lớn hơn 1 hay không. Nếu `abs_mvd_greater1_flag` được biểu thị rằng giá trị tuyệt đối không lớn hơn 1, có thể xác định được rằng giá trị tuyệt đối là 1. Nếu `abs_mvd_greater1_flag` biểu thị rằng giá trị tuyệt đối lớn hơn 1, cú pháp khác có thể tồn tại. Ví dụ, `abs_mvd_minus2` có thể tồn tại, mà có thể là giá trị của (giá trị tuyệt đối - 2). Nó biểu thị (giá trị tuyệt đối - 2) vì giá trị tuyệt đối được xác định là lớn hơn 1 (lớn hơn 2) thông qua `abs_mvd_greater0_flag` và `abs_mvd_greater1_flag` được mô tả ở trên. Khi `abs_mvd_minus2` được nhị phân hóa với chiều dài khả biến, dùng để báo hiệu với ít bit hơn. Ví dụ, có các phương pháp nhị phân hóa chiều dài khả biến chẳng hạn như Exp-Golomb, một ngôi được cắt bỏ và rice được cắt bỏ. Ngoài ra, `mvd_sign_flag` có thể là cờ biểu thị dấu của chênh lệch vector chuyển động.

Mặc dù phương pháp mã hóa đã được mô tả thông qua chênh lệch vector chuyển động theo phương án này, thông tin khác với chênh lệch vector chuyển động cũng có thể được phân chia về dấu và giá trị tuyệt đối và giá trị tuyệt đối có thể được mã hóa với cờ biểu thị xem giá trị tuyệt đối có lớn hơn giá trị nhất định hay không và giá trị được thu bằng cách lấy giá trị tuyệt đối trừ giá trị nhất định. Ngoài ra, [0] và [1] trên Fig.10 có thể biểu diễn các chỉ số thành phần. Ví dụ, [0] và [1] có thể biểu diễn thành phần x và thành phần y.

Fig.11 là sơ đồ minh họa báo hiệu độ phân giải vector chuyển động thích ứng theo một phương án của sáng chế.

Theo một phương án của sáng chế, độ phân giải biểu thị vector chuyển động hoặc chênh lệch vector chuyển động có thể thay đổi. Nói cách khác, độ phân giải mà vector chuyển động hoặc chênh lệch vector chuyển động được mã hóa có thể thay đổi. Ví dụ, độ phân giải có thể được biểu diễn về mặt các điểm ảnh (pel). Ví dụ, vector chuyển động hoặc chênh lệch vector chuyển động có thể được báo hiệu theo các đơn vị của 1/4 (một phần tư) điểm ảnh, 1/2 (một nửa) điểm ảnh, 1 (nguyên) điểm ảnh, 2 điểm ảnh, 4 điểm ảnh, v.v.. Khi cần biểu diễn 16, nó có thể được mã hóa dưới dạng $(1/4 * 64 = 16)$ để thực hiện báo hiệu theo các đơn vị 1/4, nó có thể được mã hóa dưới dạng 4 $(1 * 16 = 16)$ để thực hiện báo hiệu theo các đơn vị 1, và nó có thể được mã hóa dưới dạng 4 $(4 * 4 = 16)$ để thực hiện báo hiệu theo các đơn

vị 4. Nghĩa là, giá trị có thể được xác định như sau.

$$\text{valueDetermined} = \text{resolution} * \text{valuePerResolution}$$

Ở đây, `valueDetermined` là giá trị cần được truyền, và có thể là vector chuyển động hoặc chênh lệch vector chuyển động theo phương án này. Ngoài ra, `valuePerResolution` có thể là giá trị được thu bằng cách biểu diễn `valueDetermined` theo các đơn vị của `[/resolution]`.

Trong trường hợp này, nếu giá trị được báo hiệu dưới dạng vector chuyển động hoặc chênh lệch vector chuyển động không được phân chia theo độ phân giải, có thể gửi giá trị không chính xác khác với vector chuyển động hoặc chênh lệch vector chuyển động, mà có hiệu suất dự đoán tốt nhất thông qua việc làm tròn, v.v.. Khi sử dụng độ phân giải cao, độ không chính xác có thể được làm giảm, nhưng vì giá trị được mã hóa lớn, nhiều bit có thể được sử dụng, và khi sử dụng độ phân giải thấp, độ không chính xác có thể được tăng lên, nhưng vì giá trị được mã hóa nhỏ, nên ít bit hơn có thể được sử dụng.

Cũng có thể thiết lập độ phân giải khác đi theo các đơn vị của khối, CU, phân chia, v.v.. Do đó, độ phân giải có thể được áp dụng một cách thích ứng để thích hợp với đơn vị.

Độ phân giải có thể được báo hiệu từ bộ mã hóa đến bộ giải mã. Trong trường hợp này, báo hiệu đối với độ phân giải có thể là báo hiệu được nhị phân hóa với chiều dài khả biến được mô tả ở trên. Trong trường hợp này, khi báo hiệu được thực hiện với chỉ số tương ứng với giá trị nhỏ nhất (giá trị hàng đầu), tổng phí báo hiệu được làm giảm.

Theo một phương án, các chỉ số báo hiệu có thể được so khớp theo thứ tự từ độ phân giải cao (báo hiệu chi tiết) đến độ phân giải thấp.

Fig.11 minh họa báo hiệu đối với ba kiểu độ phân giải. Trong trường hợp này, ba báo hiệu có thể là 0, 10 và 11, và mỗi trong số ba báo hiệu có thể tương ứng với độ phân giải 1, độ phân giải 2 và độ phân giải 3. Tổng phí báo hiệu thấp khi báo hiệu độ phân giải 1 vì 1 bit là cần thiết để báo hiệu độ phân giải 1 và 2 bit là cần thiết để báo hiệu các độ phân giải còn lại. Trong ví dụ của Fig.11, độ phân giải 1, độ phân giải 2 và độ phân giải 3 lần lượt là 1/4 điểm ảnh, 1 điểm ảnh và 4 điểm ảnh.

Trong phần mô tả sau, độ phân giải vector chuyển động có thể có nghĩa là độ phân giải của chênh lệch vector chuyển động.

Fig.12 là sơ đồ minh họa báo hiệu độ phân giải vector chuyển động thích ứng theo một phương án của sáng chế.

Như được mô tả với tham chiếu đến Fig.11, vì số lượng bit cần thiết để báo hiệu độ phân giải có thể thay đổi tùy thuộc vào độ phân giải, có thể thay đổi phương pháp báo hiệu theo trường hợp. Ví dụ, giá trị báo hiệu để báo hiệu độ phân giải nhất định có thể khác tùy thuộc vào trường hợp. Ví dụ, chỉ số báo hiệu và độ phân giải có thể được so khớp theo thứ tự khác tùy thuộc vào trường hợp. Ví dụ, các độ phân giải tương ứng với báo hiệu 0, 10, 110, ... có thể lần lượt là độ phân giải 1, độ phân giải 2, độ phân giải 3, ... trong trường hợp nhất định, và có thể theo thứ tự khác với độ phân giải 1, độ phân giải 2, độ phân giải 3, ..., một cách tương ứng, trong trường hợp khác. Cũng có thể xác định hai hoặc nhiều hơn hai trường hợp.

Dựa vào Fig.12, các độ phân giải tương ứng với 0, 10 và 11 có thể lần lượt là độ phân giải 1, độ phân giải 2 và độ phân giải 3 trong trường hợp 1, và có thể lần lượt độ phân giải 2, độ phân giải 1 và độ phân giải 3 trong trường hợp 2. Trong trường hợp này, có thể có hai hoặc nhiều hơn hai trường hợp.

Fig.13 là sơ đồ minh họa báo hiệu của báo hiệu độ phân giải vector chuyển động thích ứng theo một phương án của sáng chế.

Như được mô tả với tham chiếu đến Fig.12, độ phân giải vector chuyển động có thể được báo hiệu khác đi tùy thuộc vào trường hợp. Ví dụ, giả sử rằng các độ phân giải 1/4, 1 và 4 điểm ảnh tồn tại, trong một số trường hợp, báo hiệu như được mô tả với tham chiếu đến Fig.11 có thể được sử dụng, và trong một số trường hợp, báo hiệu được minh họa trên Fig.13(a) hoặc Fig.13(b) có thể được sử dụng. Có thể là chỉ hai trường hợp hoặc tất cả ba trường hợp trong số các trường hợp của Fig.11, Fig.13(a) và Fig.13(b) tồn tại. Do đó, trong một số trường hợp, có thể báo hiệu độ phân giải mà không phải là độ phân giải cao nhất với ít bit hơn.

Fig.14 là sơ đồ minh họa báo hiệu độ phân giải vector chuyển động thích ứng theo một phương án của sáng chế.

Theo một phương án của sáng chế, độ phân giải khả thi có thể được thay đổi tùy thuộc vào trường hợp trong độ phân giải vector chuyển động thích ứng được mô tả với tham chiếu đến Fig.11. Ví dụ, giá trị độ phân giải có thể thay đổi tùy thuộc vào trường hợp. Theo một phương án, có thể sử dụng giá trị độ phân giải trong số độ phân giải 1, độ phân giải 2, độ phân giải 3, độ phân giải 4, ... trong trường hợp nhất định và sử dụng giá trị độ phân giải trong số độ phân giải A, độ phân giải B, độ phân giải C, độ phân giải D, ... trong trường hợp khác. Ngoài ra, có thể có đoạn giao không trống giữa {độ phân giải 1, độ phân giải 2,

độ phân giải 3, độ phân giải 4, ...} và {độ phân giải B, độ phân giải C, độ phân giải D, ...}. Nghĩa là, giá trị độ phân giải nhất định có thể được sử dụng trong hai hoặc nhiều hơn hai trường hợp, và trong hai hoặc nhiều hơn hai trường hợp này, tập hợp giá trị độ phân giải sử dụng được có thể là khác nhau. Ngoài ra, số lượng giá trị độ phân giải sử dụng được có thể khác nhau đối với mỗi trường hợp.

Dựa vào Fig.14, độ phân giải 1, độ phân giải 2 và độ phân giải 3 có thể được sử dụng trong trường hợp 1, và độ phân giải A, độ phân giải B và độ phân giải C có thể được sử dụng trong trường hợp 2. Ví dụ, độ phân giải 1, độ phân giải 2 và độ phân giải 3 có thể là 1/4, 1 và 4 điểm ảnh. Ngoài ra, ví dụ, độ phân giải A, độ phân giải B và độ phân giải C có thể là 1/4, 1/2 và 1 điểm ảnh.

Fig.15 là sơ đồ minh họa báo hiệu độ phân giải vector chuyển động thích ứng theo một phương án của sáng chế.

Dựa vào Fig.15, báo hiệu độ phân giải có thể được tạo ra khác đi theo ứng viên vector chuyển động hoặc ứng viên bộ dự đoán vector chuyển động. Ví dụ, có thể liên quan đến trường hợp mà được mô tả trên các hình vẽ từ Fig.12 đến Fig.14 được chọn, hoặc ứng viên nào là vector chuyển động hoặc bộ dự đoán vector chuyển động được báo hiệu. Phương pháp tạo ra báo hiệu khác nhau có thể theo phương pháp của các hình vẽ từ Fig.12 đến Fig.14.

Ví dụ, trường hợp có thể được xác định khác đi tùy thuộc vào vị trí nó được định vị trong số các ứng viên. Theo cách khác, trường hợp có thể được xác định khác đi tùy thuộc vào cách mà ứng viên đã được xây dựng.

Bộ mã hóa hoặc bộ giải mã có thể tạo ra danh mục ứng viên bao gồm ít nhất một ứng viên MV (ứng viên vector chuyển động) hoặc ít nhất một ứng viên MVP (ứng viên bộ dự đoán vector chuyển động). Có thể có khuynh hướng là MVP (bộ dự đoán vector chuyển động) ở đầu danh mục ứng viên đối với ứng viên MV (ứng viên vector chuyển động) hoặc ứng viên MVP (ứng viên bộ dự đoán vector chuyển động) có độ chính xác cao và ứng viên MVP (ứng viên bộ dự đoán vector chuyển động) ở phía sau có độ chính xác thấp trong danh mục ứng viên. Điều này có thể được chỉ định sao cho ứng viên ở đầu danh mục ứng viên được báo hiệu với ít bit hơn, và MVP (bộ dự đoán vector chuyển động) ở đầu danh mục ứng viên có độ chính xác cao hơn. Theo phương án của sáng chế, nếu độ chính xác của MVP (bộ dự đoán vector chuyển động) cao, thì giá trị chênh lệch vector chuyển động (MVD) để biểu diễn vector chuyển động với hiệu suất dự đoán tốt có thể nhỏ, và nếu độ chính xác của MVP (bộ dự đoán vector chuyển động) thấp, thì giá trị MVD (chênh lệch vector chuyển động)

để biểu diễn vector chuyển động với hiệu suất dự đoán tốt có thể lớn. Do đó, khi độ chính xác của MVP thấp, thì có thể giảm bớt các bit (ví dụ, giá trị biểu diễn giá trị chênh lệch dựa trên độ phân giải) cần thiết để biểu diễn giá trị chênh lệch vector chuyển động bằng cách báo hiệu ở độ phân giải thấp.

Với nguyên lý này, vì độ phân giải thấp có thể được sử dụng khi độ chính xác MVP (bộ dự đoán vector chuyển động) thấp, theo một phương án của sáng chế, có thể hứa hẹn rằng độ phân giải khác với độ phân giải cao nhất được báo hiệu với ít bit nhất theo ứng viên MVP (ứng viên bộ dự đoán vector chuyển động). Ví dụ, khi độ phân giải 1/4, 1, hoặc 4 điểm ảnh là khả thi, 1 hoặc 4 có thể được báo hiệu với ít bit nhất (1 bit). Dựa vào Fig.15, đối với ứng viên 1 và ứng viên 2, 1/4 điểm ảnh, mà là độ phân giải cao, được báo hiệu với ít bit nhất, và đối với ứng viên N đứng sau ứng viên 1 và ứng viên 2, độ phân giải, mà không phải là 1/4 điểm ảnh, được báo hiệu với ít bit nhất.

Fig.16 là sơ đồ minh họa báo hiệu độ phân giải vector chuyển động thích ứng theo một phương án của sáng chế.

Như được mô tả với tham chiếu đến Fig.15, báo hiệu độ phân giải vector chuyển động có thể được tạo ra khác đi tùy thuộc vào ứng viên mà là vector chuyển động hoặc bộ dự đoán vector chuyển động được xác định. Phương pháp tạo ra báo hiệu khác nhau có thể theo phương pháp của các hình vẽ từ Fig.12 đến Fig.14.

Dựa vào Fig.16, độ phân giải cao được báo hiệu với ít bit nhất đối với một số ứng viên, và độ phân giải khác với độ phân giải cao được báo hiệu với ít bit nhất đối với một số ứng viên. Ví dụ, ứng viên mà độ phân giải khác với độ phân giải cao được báo hiệu với ít bit nhất có thể là ứng viên không đúng. Ví dụ, ứng viên mà độ phân giải khác với độ phân giải cao được báo hiệu với ít bit nhất có thể là ứng viên thời gian, vector chuyển động bằng không, ứng viên không gian không liên kết, ứng viên theo việc liệu có quy trình tinh chỉnh hay không, v.v..

Ứng viên thời gian có thể là vector chuyển động từ ảnh khác. Vector chuyển động bằng không có thể là vector chuyển động trong đó tất cả các thành phần vector là 0. Ứng viên không gian không liên kết có thể là vector chuyển động được tham chiếu từ vị trí không liên kết với khối hiện thời. Quy trình tinh chỉnh có thể là quy trình tinh chỉnh bộ dự đoán vector chuyển động, ví dụ, nó có thể được tinh chỉnh thông qua so khớp mẫu tạm thời, so khớp hai hướng, v.v..

Theo một phương án của sáng chế, có thể thêm chênh lệch vector chuyển động sau khi tinh chỉnh bộ dự đoán vector chuyển động. Điều này có thể là để làm giảm giá trị chênh lệch vector chuyển động bằng cách làm cho bộ dự đoán vector chuyển động chính xác. Trong trường hợp này, báo hiệu độ phân giải vector chuyển động có thể được tạo ra khác đi đối với ứng viên không có quy trình tinh chỉnh. Ví dụ, độ phân giải mà không phải là độ phân giải cao nhất có thể được báo hiệu với ít bit nhất.

Theo một phương án khác, sau khi thêm chênh lệch vector chuyển động vào bộ dự đoán vector chuyển động, quy trình tinh chỉnh có thể được thực hiện. Trong trường hợp này, đối với ứng viên có quy trình tinh chỉnh, báo hiệu độ phân giải vector chuyển động có thể được tạo ra khác đi. Ví dụ, độ phân giải mà không phải là độ phân giải cao nhất có thể được báo hiệu với ít bit nhất. Điều này có thể là do, vì quy trình tinh chỉnh sẽ được thực hiện sau khi thêm chênh lệch vector chuyển động, có thể làm giảm lỗi dự đoán thông qua quy trình tinh chỉnh thậm chí nếu chênh lệch MV (chênh lệch vector chuyển động) không được báo hiệu chính xác nhất có thể (sao cho lỗi dự đoán nhỏ).

Là một ví dụ khác, khi ứng viên được chọn khác với ứng viên khác cao hơn hoặc bằng giá trị nhất định, thì báo hiệu độ phân giải vector chuyển động có thể được thay đổi.

Theo một phương án khác, tùy thuộc vào ứng viên mà vector chuyển động hoặc bộ dự đoán vector chuyển động được xác định, quy trình tinh chỉnh vector chuyển động tiếp theo có thể được tạo ra khác đi. Quy trình tinh chỉnh vector chuyển động có thể là quy trình để tìm kiếm vector chuyển động chính xác hơn. Ví dụ, quy trình tinh chỉnh vector chuyển động có thể là quy trình tìm kiếm khớp với khối hiện thời theo sự bổ nhiệm được thiết lập từ điểm tham chiếu (ví dụ, so khớp mẫu tạm thời hoặc so khớp hai hướng). Điểm tham chiếu có thể là vị trí tương ứng với vector chuyển động hoặc bộ dự đoán vector chuyển động được xác định. Trong trường hợp này, độ dịch chuyển từ điểm tham chiếu có thể thay đổi theo sự đồng thuận được thiết lập, và việc tạo ra quy trình tinh chỉnh vector chuyển động khác có thể là để tạo ra độ dịch chuyển từ điểm tham chiếu khác. Ví dụ, quy trình tinh chỉnh vector chuyển động có thể bắt đầu với quy trình tinh chỉnh chi tiết đối với ứng viên chính xác, và bắt đầu với quy trình tinh chỉnh ít chi tiết hơn đối với ứng viên không chính xác. Ứng viên chính xác và ứng viên không chính xác có thể được xác định theo vị trí trong danh mục ứng viên hoặc cách mà ứng viên được tạo ra. Cách mà ứng viên được tạo ra có thể liên quan đến vị trí mà ứng viên không gian được đưa đến. Ngoài ra, sự tham chiếu có thể được thực hiện đến phần mô tả của Fig.16 đối với điều này. Ngoài ra, việc tinh chỉnh

chi tiết hoặc việc tinh chỉnh ít chi tiết hơn có thể là liệu khối thích hợp có được tìm thấy hay không trong khi di chuyển ít hoặc trong khi di chuyển nhiều từ điểm tham chiếu. Ngoài ra, trong trường hợp tìm kiếm khối thích hợp trong khi di chuyển nhiều, có thể thêm quy trình tìm kiếm trong khi di chuyển từng chút một từ khối thích hợp nhất được tìm thấy trong khi di chuyển nhiều.

Theo một phương án khác, báo hiệu độ phân giải vector chuyển động có thể được thay đổi trên cơ sở POC của ảnh hiện thời và POC của ảnh tham chiếu của vector chuyển động hoặc ứng viên bộ dự đoán vector chuyển động. Phương pháp tạo ra báo hiệu khác nhau có thể theo phương pháp của các hình vẽ từ Fig.12 đến Fig.14.

Ví dụ, khi chênh lệch giữa số đếm thứ tự ảnh (POC) của ảnh hiện thời và POC của ảnh tham chiếu của vector chuyển động hoặc ứng viên bộ dự đoán vector chuyển động lớn, vector chuyển động hoặc bộ dự đoán vector chuyển động có thể không chính xác, và độ phân giải khác với độ phân giải cao có thể được báo hiệu với ít bit nhất.

Là một ví dụ khác, báo hiệu độ phân giải vector chuyển động có thể được thay đổi dựa trên việc liệu việc thay đổi tỷ lệ vector chuyển động có cần được thực hiện hay không. Ví dụ, khi vector chuyển động hoặc bộ dự đoán vector chuyển động được chọn là ứng viên mà thực hiện việc thay đổi tỷ lệ vector chuyển động được thực hiện, độ phân giải khác với độ phân giải cao có thể được báo hiệu với ít bit nhất. Trường hợp thực hiện việc thay đổi tỷ lệ vector chuyển động có thể là trường hợp trong đó ảnh tham chiếu cho khối hiện thời và ảnh tham chiếu của ứng viên được tham chiếu là khác nhau.

Fig.17 là sơ đồ minh họa phép dự đoán chuyển động afin theo một phương án của sáng chế.

Theo phương pháp dự đoán thông thường như được mô tả với tham chiếu đến Fig.8, có thể dự đoán khối hiện thời từ khối từ vị trí được di chuyển mà không xoay hoặc thay đổi tỷ lệ. Nó có thể được dự đoán từ khối tham chiếu có cùng kích thước, hình dạng và góc với khối hiện thời. Fig.8 chỉ là mô hình chuyển động tịnh tiến. Tuy nhiên, nội dung chứa trong hình ảnh thực tế có thể có chuyển động phức tạp hơn, và hiệu suất dự đoán có thể được cải thiện hơn nữa nếu được dự đoán từ các hình dạng khác nhau.

Dựa vào Fig.17, khối đang được dự đoán được biểu thị bởi đường nét liền trong ảnh hiện thời. Khối hiện thời có thể được dự đoán bằng cách tham chiếu đến khối có hình dạng, kích thước và góc khác với khối đang được dự đoán, và khối tham chiếu được biểu thị bởi

đường chấm chấm trong ảnh tham chiếu. Vị trí giống với vị trí trong ảnh của khối tham chiếu được biểu thị bởi đường chấm chấm trong ảnh hiện thời. Trong trường hợp này, khối tham chiếu có thể là khối được biểu diễn bằng phép biến đổi afin trong khối hiện thời. Như vậy, có thể biểu diễn kéo dài (thay đổi tỷ lệ), xoay, cắt, phản chiếu, phép chiếu trực giao, v.v..

Số lượng thông số biểu diễn chuyển động afin và phép biến đổi afin có thể thay đổi. Nếu nhiều thông số hơn được sử dụng, thì nhiều chuyển động khác nhau có thể được biểu diễn hơn so với khi ít thông số hơn được sử dụng, nhưng có khả năng là tổng phí có thể xuất hiện trong báo hiệu hoặc phép tính, v.v..

Ví dụ, phép biến đổi afin có thể được biểu diễn bởi 6 thông số. Theo cách khác, phép biến đổi afin có thể được biểu diễn bởi ba vector chuyển động điểm điều khiển.

Fig.18 là sơ đồ minh họa phép dự đoán chuyển động afin theo một phương án của sáng chế.

Có thể biểu diễn sự di chuyển phức tạp nhờ sử dụng phép biến đổi afin như trên Fig.17, nhưng để làm giảm tổng phí báo hiệu và phép tính cho điều này, phép dự đoán chuyển động afin hoặc phép biến đổi afin đơn giản hơn có thể được sử dụng. Bằng cách giới hạn sự di chuyển, phép dự đoán chuyển động afin đơn giản hơn có thể được tạo ra. Việc giới hạn sự di chuyển có thể giới hạn hình dạng của khối hiện thời đang được biến đổi thành khối tham chiếu.

Dựa vào Fig.18, phép dự đoán chuyển động afin có thể được thực hiện nhờ sử dụng các vector chuyển động điểm điều khiển v_0 và v_1 . Việc sử dụng hai vector v_0 và v_1 có thể tương đương với sử dụng bốn thông số. Với hai vector v_0 và v_1 hoặc bốn thông số, có thể biểu thị hình dạng mà của khối tham chiếu khối hiện thời được dự đoán. Bằng cách sử dụng phép biến đổi afin đơn giản này, có thể biểu diễn sự di chuyển của việc xoay và thay đổi tỷ lệ (thu/phóng) của khối. Dựa vào Fig.18, khối hiện thời được biểu thị bởi đường nét liền có thể được dự đoán từ vị trí được biểu thị bởi đường chấm chấm trên Fig.18 trong ảnh tham chiếu. Mỗi điểm (điểm ảnh) của khối hiện thời có thể được ánh xạ đến điểm khác thông qua phép biến đổi afin.

Fig.19 là hình vẽ minh họa trường vector chuyển động theo một phương án của sáng chế. Vector chuyển động điểm điều khiển v_0 trên Fig.18 có thể là (v_{0x}, v_{0y}) và có thể là vector chuyển động của điểm điều khiển góc trên cùng bên trái. Ngoài ra, vector chuyển động

điểm điều khiển $v1$ có thể là (v_{1x}, v_{1y}) và có thể là vector chuyển động của điểm điều khiển góc trên cùng bên phải. Trong trường hợp này, vector chuyển động (v_x, v_y) ở vị trí (x, y) có thể là như được minh họa trên Fig.19. Do đó, vector chuyển động đối với mỗi vị trí điểm ảnh hoặc vị trí nhất định có thể được đánh giá theo sự biểu diễn của Fig.19, mà là dựa trên $v0$ và $v1$.

Ngoài ra, (x, y) trong sự biểu diễn của Fig.19 có thể là tọa độ tương đối trong khối. Ví dụ, (x, y) có thể là vị trí khi vị trí trên cùng bên trái của khối là $(0, 0)$.

Nếu $v0$ là vector chuyển động điểm điều khiển đối với vị trí $(x0, y0)$ trên ảnh và $v1$ là vector chuyển động điểm điều khiển đối với vị trí $(x1, y1)$ trên ảnh, nếu nó được dự định để biểu diễn vị trí (x, y) trong khối sử dụng các tọa độ giống với các vị trí của $v0, v1$, nó có thể được biểu diễn bằng cách lần lượt thay đổi x và y thành $(x - x0)$ và $(y - y0)$ trong sự biểu diễn của Fig.19. Ngoài ra, w (chiều rộng của khối) có thể là $(x1 - x0)$.

Fig.20 là sơ đồ minh họa phép dự đoán chuyển động afin theo một phương án của sáng chế.

Theo một phương án của sáng chế, chuyển động afin có thể được biểu diễn nhờ sử dụng các vector chuyển động điểm điều khiển hoặc các thông số.

Dựa vào Fig.20, phép dự đoán chuyển động afin có thể được thực hiện nhờ sử dụng các vector chuyển động điểm điều khiển $v0, v1$ và $v2$. Việc sử dụng ba vector $v0, v1, v2$ có thể tương đương với sử dụng sáu thông số. Ba vector $v0, v1$ và $v2$ hoặc sáu thông số có thể biểu thị hình dạng của khối tham chiếu mà khối hiện thời được dự đoán. Dựa vào Fig.20, khối hiện thời được biểu thị bởi đường nét liền có thể được dự đoán từ vị trí được biểu thị bởi đường chấm chấm trên Fig.20 trong ảnh tham chiếu. Mỗi điểm (điểm ảnh) của khối hiện thời có thể được ánh xạ đến điểm khác thông qua phép biến đổi afin.

Fig.21 là hình vẽ minh họa trường vector chuyển động theo một phương án của sáng chế. Trên Fig.20, vector chuyển động điểm điều khiển $v0$ có thể là (mv_0^x, mv_0^y) và có thể là vector chuyển động của điểm điều khiển góc trên cùng bên trái, vector chuyển động điểm điều khiển $v1$ có thể là (mv_1^x, mv_1^y) và có thể là vector chuyển động của điểm điều khiển góc trên cùng bên phải, và vector chuyển động điểm điều khiển $v2$ có thể là (mv_2^x, mv_2^y) và có thể là vector chuyển động của điểm điều khiển góc dưới cùng bên trái. Trong trường hợp này, vector chuyển động (mv^x, mv^y) của vị trí (x, y) có thể là như được minh họa trên Fig.21. Do đó, vector chuyển động đối với mỗi vị trí điểm ảnh hoặc vị

trí nhất định có thể được đánh giá theo sự biểu diễn của Fig.21, mà là dựa trên v_0 , v_1 và v_2 .

Ngoài ra, (x, y) trong sự biểu diễn của Fig.21 có thể là tọa độ tương đối trong khối. Ví dụ, (x, y) có thể là vị trí khi vị trí trên cùng bên trái của khối là $(0, 0)$. Do đó, nếu giả sử rằng v_0 là vector chuyển động điểm điều khiển đối với vị trí (x_0, y_0) , v_1 là vector chuyển động điểm điều khiển đối với vị trí (x_1, y_1) , và v_2 là vector chuyển động điểm điều khiển đối với vị trí (x_2, y_2) và được dự định để biểu diễn (x, y) nhờ sử dụng các tọa độ giống với các vị trí của v_0 , v_1 và v_2 , có thể được biểu diễn bằng cách lần lượt thay đổi x và y thành $(x - x_0)$ và $(y - y_0)$ trong sự biểu diễn của Fig.21. Ngoài ra, w (chiều rộng của khối) có thể là $(x_1 - x_0)$, và h (chiều cao của khối) có thể là $(y_2 - y_0)$.

Fig.22 là sơ đồ minh họa phép dự đoán chuyển động afin theo một phương án của sáng chế.

Như được mô tả ở trên, trường vector chuyển động tồn tại, và vector chuyển động có thể được tính toán đối với mỗi điểm ảnh, nhưng nhằm đơn giản, phép biến đổi afin có thể được thực hiện trên cơ sở khối con như được minh họa trên Fig.22. Ví dụ, hình chữ nhật nhỏ trên Fig.22(a) là khối con, vector chuyển động biểu diễn của khối con có thể được xây dựng, và vector chuyển động biểu diễn có thể được sử dụng cho các điểm ảnh của khối con. Ngoài ra, do sự di chuyển phức tạp đã được biểu diễn trên Fig.17, Fig.18, Fig.20, v.v., khối con có thể tương ứng với khối tham chiếu bằng cách biểu diễn sự di chuyển này, hoặc nó có thể được tạo ra đơn giản hơn bằng cách áp dụng chỉ chuyển động tịnh tiến vào khối con. Trên Fig.22(a), v_0 , v_1 và v_2 có thể là các vector chuyển động điểm điều khiển.

Trong trường hợp này, kích thước của khối con có thể là $M \times N$, và M và N có thể giống nhau như được minh họa trên Fig.22(b). Và, Mv_{Pre} có thể là độ chính xác phân số vector chuyển động. Ngoài ra, (v_{0x}, v_{0y}) , (v_{1x}, v_{1y}) , và (v_{2x}, v_{2y}) có thể lần lượt là các vector chuyển động của các điểm điều khiển trên cùng bên trái, trên cùng bên phải và dưới cùng bên trái. (v_{2x}, v_{2y}) có thể là vector chuyển động của điểm điều khiển dưới cùng bên trái của khối hiện thời, ví dụ, trong trường hợp 4 thông số, nó có thể là MV (motion vector - vector chuyển động) cho điểm điều khiển dưới cùng bên trái được tính toán bởi sự biểu diễn của Fig.19.

Ngoài ra, khi xây dựng vector chuyển động biểu diễn của khối con, có thể tính toán vector chuyển động biểu diễn nhờ sử dụng vị trí mẫu trung tâm của khối con. Ngoài ra, khi xây dựng vector chuyển động của khối con, vector chuyển động có độ chính xác cao hơn so với vector chuyển động bình thường có thể được sử dụng, và do đó, bộ lọc nội suy bù chuyển

động có thể được áp dụng.

Theo một phương án khác, kích thước của khối con không khả biến và có thể được cố định thành kích thước cụ thể. Ví dụ, kích thước khối con có thể được cố định thành kích thước 4*4.

Fig.23 là sơ đồ minh họa một chế độ của phép dự đoán chuyển động affin theo một phương án của sáng chế.

Theo một phương án của sáng chế, có thể có chế độ liên ảnh affin làm một ví dụ của phép dự đoán chuyển động affin. Có thể có cờ biểu thị rằng chế độ là chế độ liên ảnh affin. Dựa vào Fig.23, có thể có các khối ở các vị trí A, B, C, D và E gần v_0 và v_1 , và các vector chuyển động mà lần lượt tương ứng với các khối có thể được gọi là v_A , v_B , v_C , v_D , v_E . Nhờ sử dụng cờ này, có thể xây dựng danh mục ứng viên cho các vector chuyển động hoặc các bộ dự đoán vector chuyển động sau.

$$\{(v_0, v_1) | v_0 = \{v_A, v_B, v_C\}, v_1 = \{v_D, v_E\}\}$$

Nghĩa là, cặp (v_0, v_1) có thể được xây dựng với v_0 được chọn từ v_A , v_B và v_C và v_1 được chọn từ v_D và v_E . Trong trường hợp này, vector chuyển động có thể được thay đổi tỷ lệ theo số đếm thứ tự ảnh (POC) của tham chiếu của khối lân cận, POC của tham chiếu cho CU hiện thời (đơn vị mã hóa hiện thời; khối hiện thời) và POC của CU hiện thời. Khi danh mục ứng viên được xây dựng với cùng các cặp vector chuyển động như được mô tả ở trên, có thể báo hiệu ứng viên mà của danh mục ứng viên được chọn và việc liệu ứng viên này có được chọn hay không. Ngoài ra, nếu danh mục ứng viên không được làm đầy đủ, có thể làm đầy danh mục ứng viên với các ứng viên dự đoán liên ảnh khác. Ví dụ, danh mục này có thể được làm đầy nhờ sử dụng các ứng viên dự đoán vector chuyển động nâng cao (AMVP). Ngoài ra, có thể báo hiệu chênh lệch cần được sử dụng để hiệu chỉnh thay vì sử dụng trực tiếp v_0 và v_1 được chọn từ danh mục ứng viên làm vector chuyển động điểm điều khiển của phép dự đoán chuyển động affin, nhờ đó có khả năng xây dựng vector chuyển động điểm điều khiển tốt hơn. Nghĩa là, trong bộ giải mã, có thể sử dụng v_0' và v_1' được xây dựng bằng cách thêm chênh lệch vào v_0 và v_1 được chọn từ danh mục ứng viên làm các vector chuyển động điểm điều khiển cho phép dự đoán chuyển động affin.

Theo một phương án, cũng có thể sử dụng chế độ liên ảnh affin cho đơn vị mã hóa (CU) có kích thước lớn hơn hoặc bằng kích thước cụ thể.

Fig.24 là sơ đồ minh họa một chế độ của phép dự đoán chuyển động affin theo một

phương án của sáng chế.

Theo một phương án của sáng chế, có thể có chế độ hợp nhất afin làm một ví dụ của phép dự đoán chuyển động afin. Có thể có cờ biểu thị rằng chế độ này là chế độ hợp nhất afin. Theo chế độ hợp nhất afin, khi phép dự đoán chuyển động afin được sử dụng xung quanh khối hiện thời, vectơ chuyển động điểm điều khiển của khối hiện thời có thể được tính toán từ các vectơ chuyển động của khối hoặc xung quanh khối. Ví dụ, khi kiểm tra xem khối lân cận có sử dụng phép dự đoán chuyển động afin hay không, khối lân cận mà trở thành ứng viên có thể là như được minh họa trên Fig.24(a). Ngoài ra, có thể kiểm tra xem phép dự đoán chuyển động afin có được sử dụng theo thứ tự A, B, C, D và E hay không, và khi khối sử dụng phép dự đoán chuyển động afin được tìm thấy, vectơ chuyển động điểm điều khiển của khối hiện thời có thể được tính toán nhờ sử dụng các vectơ chuyển động của khối hoặc xung quanh khối. A, B, C, D và E có thể lần lượt là bên trái, ở trên, ở trên bên phải, dưới cùng bên trái, ở trên bên trái, như được minh họa trên Fig.24(a).

Theo một phương án, khi khối ở vị trí A sử dụng phép dự đoán chuyển động afin như được minh họa trên Fig.24(b), v_0 và v_1 có thể được tính toán nhờ sử dụng các vectơ chuyển động của khối hoặc xung quanh khối. Các vectơ chuyển động xung quanh khối có thể là v_2 , v_3 và v_4 .

Theo phương án nêu trên, thứ tự của các khối lân cận cần được tham chiếu được xác định. Tuy nhiên, hiệu suất của vectơ chuyển động điểm điều khiển được suy ra từ vị trí cụ thể không luôn luôn tốt hơn. Do đó, theo một phương án khác, có thể báo hiệu xem có suy ra vectơ chuyển động điểm điều khiển hay không bằng cách tham chiếu đến khối nằm ở vị trí nào. Ví dụ, các vị trí ứng viên của sự suy ra vectơ chuyển động điểm điều khiển được xác định theo thứ tự A, B, C, D và E của Fig.24(a), và có thể báo hiệu vị trí ứng viên cần được tham chiếu đến.

Theo một phương án khác, khi suy ra vectơ chuyển động điểm điều khiển, độ chính xác có thể được tăng lên bằng cách lấy nó từ khối ở gần cho mỗi vectơ chuyển động điểm điều khiển. Ví dụ, tham chiếu đến Fig.24, khối bên trái có thể được tham chiếu đến khi suy ra v_0 , và khối ở trên có thể được tham chiếu đến khi suy ra v_1 . Theo cách khác, A, D, hoặc E có thể được tham chiếu đến khi suy ra v_0 , và B hoặc C có thể được tham chiếu đến khi suy ra v_1 .

Fig.25 là sơ đồ minh họa phép suy ra bộ dự đoán chuyển động afin theo một phương án của sáng chế.

Các vector chuyển động điểm điều khiển có thể cần thiết cho phép dự đoán chuyển động afin, và trường vector chuyển động, nghĩa là, vector chuyển động cho khối con hoặc vị trí nhất định, có thể được tính toán dựa trên các vector chuyển động điểm điều khiển. Vector chuyển động điểm điều khiển có thể được gọi là vector nguồn.

Trong trường hợp này, MV điểm điều khiển (vector chuyển động điểm điều khiển) có thể dựa trên bộ dự đoán. Ví dụ, bộ dự đoán có thể là MV điểm điều khiển (vector chuyển động điểm điều khiển). Là một ví dụ khác, MV điểm điều khiển có thể được tính toán dựa trên bộ dự đoán và chênh lệch. Cụ thể là, MV điểm điều khiển (vector chuyển động điểm điều khiển) có thể được tính toán bằng cách thêm chênh lệch vào bộ dự đoán hoặc lấy bộ dự đoán trừ đi chênh lệch.

Trong trường hợp này, trong quá trình xây dựng bộ dự đoán của MV điểm điều khiển, có thể suy ra nó từ các MV điểm điều khiển (vector chuyển động điểm điều khiển) hoặc các MV (các vector chuyển động) của khối lân cận mà phép dự đoán chuyển động afin (phép bù chuyển động afin (motion compensation, MC)) được thực hiện. Ví dụ, nếu khối tương ứng với vị trí được thiết lập trước trải qua phép dự đoán chuyển động afin, bộ dự đoán cho phép bù chuyển động afin của khối hiện thời có thể được suy ra từ các MV điểm điều khiển hoặc các MV của khối. Dựa vào Fig.25, các vị trí được thiết lập trước có thể là A0, A1, B0, B1 và B2. Theo cách khác, các vị trí được thiết lập trước có thể bao gồm vị trí liền kề với khối hiện thời và vị trí không liền kề với khối hiện thời. Ngoài ra, có thể tham chiếu đến các MV điểm điều khiển (vector chuyển động điểm điều khiển) hoặc các MV (các vector chuyển động) ở vị trí được thiết lập trước (không gian), và có thể tham chiếu đến các MV điểm điều khiển thời gian hoặc các MV ở vị trí được thiết lập trước.

Ứng viên của phép bù chuyển động (motion compensation, MC) afin có thể được xây dựng theo cách giống với theo phương án của Fig.25, và ứng viên này cũng có thể được gọi là ứng viên được thừa hưởng. Theo cách khác, ứng viên này có thể được gọi là ứng viên hợp nhất. Ngoài ra, khi tham chiếu đến các vị trí được thiết lập trước theo phương pháp của Fig.25, tham chiếu có thể được tạo ra theo thứ tự được thiết lập trước.

Fig.26 là sơ đồ minh họa phép suy ra bộ dự đoán chuyển động afin theo một phương án của sáng chế.

Các vector chuyển động điểm điều khiển có thể cần thiết cho phép dự đoán chuyển động afin, và trường vector chuyển động, nghĩa là, vector chuyển động cho khối con hoặc vị trí nhất định, có thể được tính toán dựa trên các vector chuyển động điểm điều khiển. Vector

chuyển động điểm điều khiển cũng có thể được gọi là vector nguồn.

Trong trường hợp này, MV điểm điều khiển (vector chuyển động điểm điều khiển) có thể dựa trên bộ dự đoán. Ví dụ, bộ dự đoán có thể là MV điểm điều khiển (vector chuyển động điểm điều khiển). Là một ví dụ khác, MV điểm điều khiển (vector chuyển động điểm điều khiển) có thể được tính toán dựa trên bộ dự đoán và chênh lệch. Cụ thể là, MV điểm điều khiển (vector chuyển động điểm điều khiển) có thể được tính toán bằng cách thêm chênh lệch vào bộ dự đoán hoặc lấy bộ dự đoán trừ chênh lệch.

Trong trường hợp này, có thể suy ra nó từ các MV lân cận trong quá trình xây dựng bộ dự đoán của MV điểm điều khiển. Trong trường hợp này, các MV lân cận có thể bao gồm các MV (các vector chuyển động) mà không trải qua phép bù chuyển động (MC) afin. Ví dụ, khi suy ra mỗi MV điểm điều khiển (vector chuyển động điểm điều khiển) của khối hiện thời, MV ở vị trí được thiết lập trước cho mỗi MV điểm điều khiển có thể được sử dụng làm bộ dự đoán của MV điểm điều khiển. Ví dụ, vị trí được thiết lập trước có thể là phần được bao gồm trong khối liên kề với phần của vị trí được thiết lập trước.

Dựa vào Fig.26, các MV điểm điều khiển (vector chuyển động điểm điều khiển) mv_0 , mv_1 và mv_2 có thể được xác định. Trong trường hợp này, theo một phương án của sáng chế, MV (vector chuyển động) tương ứng với các vị trí được thiết lập trước A, B và C có thể được sử dụng làm bộ dự đoán cho mv_0 . Ngoài ra, MV (vector chuyển động) tương ứng với các vị trí được thiết lập trước D và E có thể được sử dụng làm bộ dự đoán cho mv_1 . MV (vector chuyển động) tương ứng với các vị trí được thiết lập trước F và G có thể được sử dụng làm bộ dự đoán cho mv_2 .

Ngoài ra, khi xác định mỗi bộ dự đoán của các MV điểm điều khiển (vector chuyển động điểm điều khiển) mv_0 , mv_1 và mv_2 theo phương án của Fig.26, thứ tự tham chiếu đến vị trí được thiết lập trước cho mỗi vị trí điểm điều khiển có thể được xác định. Ngoài ra, có thể có các vị trí được thiết lập trước được gọi là các bộ dự đoán của MV điểm điều khiển cho mỗi điểm điều khiển, và tổ hợp khả thi của các vị trí được thiết lập trước có thể được xác định.

Ứng viên của MC afin (phép bù chuyển động afin) có thể được xây dựng theo cách giống với theo phương án của Fig.26, và ứng viên này cũng có thể được gọi là ứng viên được xây dựng. Theo cách khác, ứng viên này có thể được gọi là ứng viên liên ảnh hoặc các ứng viên ảo. Ngoài ra, khi tham chiếu đến các vị trí được thiết lập trước theo phương pháp của Fig.41, sự tham chiếu có thể được tạo ra theo thứ tự được thiết lập trước.

Theo một phương án của sáng chế, nhờ sử dụng các phương án được mô tả với tham chiếu đến các hình vẽ từ Fig.23 đến 26 hoặc tổ hợp của chúng, danh mục ứng viên của MC afin (phép bù chuyển động afin) hoặc danh mục ứng viên MV điểm điều khiển của MC afin (phép bù chuyển động afin) có thể được tạo ra.

Fig.27 là sơ đồ minh họa phép suy ra bộ dự đoán chuyển động afin theo một phương án của sáng chế.

Như được mô tả với tham chiếu đến các hình vẽ từ Fig.24 đến Fig.25, MV điểm điều khiển (vector chuyển động điểm điều khiển) dùng cho phép dự đoán chuyển động afin của khối hiện thời có thể được suy ra từ khối lân cận mà trải qua phép dự đoán chuyển động afin. Trong trường hợp này, cùng phương pháp như trên Fig.27 có thể được sử dụng. Trên Fig.27, các MV hoặc MV điểm điều khiển (vector chuyển động) của trên cùng bên trái, trên cùng bên phải, và dưới cùng bên trái của khối lân cận mà trải qua phép dự đoán chuyển động afin lần lượt là (v_{E0x}, v_{E0y}) , (v_{E1x}, v_{E1y}) và (v_{E2x}, v_{E2y}) . Ngoài ra, các tọa độ của trên cùng bên trái, trên cùng bên phải, và dưới cùng bên trái của khối lân cận mà trải qua phép dự đoán chuyển động afin có thể lần lượt là (x_{E0}, y_{E0}) , (x_{E1}, y_{E1}) và (x_{E2}, y_{E2}) . Trong trường hợp này, (v_{0x}, v_{0y}) và (v_{1x}, v_{1y}) , mà là các bộ dự đoán của các MV điểm điều khiển (vector chuyển động điểm điều khiển) hoặc các MV điểm điều khiển (vector chuyển động điểm điều khiển) của khối hiện thời, có thể được tính toán theo Fig.27.

Fig.28 là sơ đồ minh họa phép suy ra bộ dự đoán chuyển động afin theo một phương án của sáng chế.

Như được mô tả ở trên, nhiều MV chuyển động điểm điều khiển (vector chuyển động điểm điều khiển) hoặc nhiều bộ dự đoán MV điểm điều khiển (bộ dự đoán vector chuyển động điểm điều khiển) có thể cần thiết cho phép bù chuyển động afin. Trong trường hợp này, MV chuyển động điểm điều khiển (vector chuyển động điểm điều khiển) hoặc bộ dự đoán MV điểm điều khiển (bộ dự đoán vector chuyển động điểm điều khiển) khác có thể được suy ra từ MV chuyển động điểm điều khiển (vector chuyển động điểm điều khiển) hoặc bộ dự đoán MV điểm điều khiển (bộ dự đoán vector chuyển động điểm điều khiển) nhất định.

Ví dụ, khi hai MV điểm điều khiển (vector chuyển động điểm điều khiển) hoặc hai bộ dự đoán MV điểm điều khiển (vector chuyển động điểm điều khiển các bộ dự đoán) được xây dựng bằng phương pháp được mô tả trên các hình vẽ trước đó, MV điểm điều khiển (vector chuyển động điểm điều khiển) khác hoặc bộ dự đoán MV điểm điều khiển (bộ dự

đoán vector chuyển động điểm điều khiển) khác có thể được tạo ra dựa trên điều này.

Dựa vào Fig.28, phương pháp tạo ra mv_0 , mv_1 và mv_2 , mà là các bộ dự đoán MV điểm điều khiển (vector chuyển động điểm điều khiển các bộ dự đoán) hoặc các MV điểm điều khiển (vector chuyển động điểm điều khiển) của trên cùng bên trái, trên cùng bên phải, và dưới cùng bên trái, được minh họa. Trên hình vẽ, x và y lần lượt biểu diễn thành phần x và thành phần y , và kích thước khối hiện thời có thể là $w \times h$.

Fig.29 là sơ đồ minh họa phương pháp tạo ra vector chuyển động điểm điều khiển theo một phương án của sáng chế.

Theo một phương án của sáng chế, có thể xác định MV điểm điều khiển (vector chuyển động điểm điều khiển) bằng cách xây dựng bộ dự đoán cho MV điểm điều khiển (vector chuyển động điểm điều khiển) để thực hiện MC afin (phép bù chuyển động afin) đối với khối hiện thời và thêm chênh lệch vào đó. Theo một phương án, có thể xây dựng bộ dự đoán của MV điểm điều khiển bằng phương pháp được mô tả với tham chiếu đến các hình vẽ từ Fig.23 đến Fig.26. Chênh lệch có thể được báo hiệu từ bộ mã hóa đến bộ giải mã.

Dựa vào Fig.29, các chênh lệch đối với các MV điểm điều khiển (vector chuyển động điểm điều khiển) có thể tồn tại. Ngoài ra, các chênh lệch đối với các MV điểm điều khiển (vector chuyển động điểm điều khiển) có thể lần lượt được báo hiệu. Fig.29(a) minh họa phương pháp xác định mv_0 và mv_1 , mà là các MV điểm điều khiển của mô hình 4 thông số, và Fig.29(b) minh họa phương pháp xác định mv_0 , mv_1 và mv_2 , mà là các MV điểm điều khiển của mô hình 6 thông số. Các MV điểm điều khiển (vector chuyển động điểm điều khiển) được xác định bằng cách thêm mvd_0 , mvd_1 và mvd_2 , mà là các chênh lệch đối với các MV điểm điều khiển (vector chuyển động điểm điều khiển), vào bộ dự đoán.

Các số hạng được biểu thị với thanh phía trên trên Fig.29 có thể là các bộ dự đoán của các MV điểm điều khiển (vector chuyển động điểm điều khiển).

Fig.30 là sơ đồ minh họa phương pháp xác định chênh lệch vector chuyển động thông qua phương pháp được mô tả với tham chiếu đến Fig.29.

Là một phương án, chênh lệch vector chuyển động có thể được báo hiệu bằng phương pháp được mô tả với tham chiếu đến Fig.10. Và, chênh lệch vector chuyển động được xác định bởi được phương pháp báo hiệu có thể là $1Mvd$ của Fig.30. Ngoài ra, giá trị chẳng hạn như chênh lệch vector chuyển động mvd được báo hiệu trên Fig.29, nghĩa là, giá trị chẳng hạn như mvd_0 , mvd_1 và mvd_2 có thể là $1Mvd$ trên Fig.30. Như được mô tả với tham chiếu

đến Fig.29, mvd (chênh lệch vector chuyển động) được báo hiệu có thể được xác định là chênh lệch với bộ dự đoán của MV điểm điều khiển (vector chuyển động điểm điều khiển), và chênh lệch được xác định có thể là $MvdL0$ và $MvdL1$ của Fig.30. $L0$ có thể biểu thị danh mục tham chiếu 0 (danh mục ảnh tham chiếu thứ không), và $L1$ có thể biểu thị danh mục tham chiếu 1 (danh mục ảnh tham chiếu thứ nhất). $compIdx$ là chỉ số thành phần, và có thể biểu thị thành phần x , y , v.v..

Fig.31 là sơ đồ minh họa phương pháp tạo ra vector chuyển động điểm điều khiển theo một phương án của sáng chế.

Theo một phương án của sáng chế, có thể xác định MV điểm điều khiển (vector chuyển động điểm điều khiển) bằng cách xây dựng bộ dự đoán cho MV điểm điều khiển để thực hiện MC afin (phép bù chuyển động afin) đối với khối hiện thời và thêm chênh lệch vào đó. Theo một phương án, có thể xây dựng bộ dự đoán của MV điểm điều khiển (vector chuyển động điểm điều khiển) bằng phương pháp được mô tả với tham chiếu đến các hình vẽ từ Fig.23 đến Fig.26. Chênh lệch có thể được báo hiệu từ bộ mã hóa đến bộ giải mã.

Dựa vào Fig.31, bộ dự đoán cho chênh lệch đối với mỗi MV điểm điều khiển (vector chuyển động điểm điều khiển) có thể tồn tại. Ví dụ, chênh lệch của MV điểm điều khiển (vector chuyển động điểm điều khiển) khác có thể được xác định dựa trên chênh lệch của MV điểm điều khiển (vector chuyển động điểm điều khiển) nhất định. Điều này có thể dựa trên độ tương đồng giữa các chênh lệch đối với các MV điểm điều khiển (vector chuyển động điểm điều khiển). Vì các chênh lệch là tương tự nhau, nếu bộ dự đoán được xác định, thì có thể tạo ra chênh lệch nhỏ với bộ dự đoán. Trong trường hợp này, bộ dự đoán chênh lệch đối với MV điểm điều khiển (vector chuyển động điểm điều khiển) có thể được báo hiệu, và chênh lệch với bộ dự đoán chênh lệch đối với MV điểm điều khiển (vector chuyển động điểm điều khiển) có thể được báo hiệu.

Fig.31(a) minh họa phương pháp xác định $mv0$ và $mv1$, mà là các MV điểm điều khiển (vector chuyển động điểm điều khiển) của mô hình 4 thông số, và Fig.31(b) minh họa phương pháp xác định $mv0$, $mv1$ và $mv2$, mà là các MV điểm điều khiển (vector chuyển động điểm điều khiển) của mô hình 6 thông số.

Dựa vào Fig.31, đối với chênh lệch đối với mỗi MV điểm điều khiển, chênh lệch đối với MV điểm điều khiển và MV điểm điều khiển được xác định dựa trên chênh lệch $mvd0$ của $mv0$, mà là MV điểm điều khiển 0. $mvd0$, $mvd1$ và $mvd2$ được minh họa trên Fig.31 có thể được báo hiệu từ bộ mã hóa đến bộ giải mã. So với phương pháp được mô tả với

tham chiếu đến Fig.29, theo phương pháp của Fig.31, các giá trị của $mvd1$ và $mvd2$ mà được báo hiệu có thể là khác nhau thậm chí nếu cùng $mv0$, $mv1$ và $mv2$ như trên Fig.29 và cùng các bộ dự đoán được sử dụng. Nếu chênh lệch với các bộ dự đoán của các MV điểm điều khiển $mv0$, $mv1$ và $mv2$ là tương tự, khi phương pháp của Fig.31 được sử dụng, có khả năng là các giá trị tuyệt đối của $mvd1$ và $mvd2$ nhỏ hơn khi phương pháp của Fig.29 được sử dụng, và do đó tổng phí báo hiệu của $mvd1$ và $mvd2$ có thể được làm giảm. Dựa vào Fig.31, chênh lệch với bộ dự đoán của $mv1$ có thể được xác định là $(mvd1 + mvd0)$, và chênh lệch với bộ dự đoán của $mv2$ có thể được xác định là $(mvd2 + mvd0)$.

Các số hạng được biểu thị với thanh phía trên trên Fig.31 có thể là các bộ dự đoán của MV các điểm điều khiển.

Fig.32 là sơ đồ minh họa phương pháp xác định chênh lệch vector chuyển động thông qua phương pháp được mô tả với tham chiếu đến Fig.31.

Là một phương án, chênh lệch vector chuyển động có thể được báo hiệu bằng phương pháp được mô tả với tham chiếu đến Fig.10 hoặc Fig.33. Ngoài ra, chênh lệch vector chuyển động được xác định dựa trên thông số được báo hiệu có thể là $1Mvd$ của Fig.32. Ngoài ra, mvd được báo hiệu trên Fig.31, nghĩa là, giá trị chẳng hạn như $mvd0$, $mvd1$, $mvd2$ có thể là $1Mvd$ trên Fig.32.

$MvdLX$ của Fig.32 có thể là chênh lệch giữa mỗi MV điểm điều khiển (vector chuyển động điểm điều khiển) và bộ dự đoán. Nghĩa là, nó có thể là $(mv - mvp)$. Trong trường hợp này, như được mô tả với tham chiếu đến Fig.31, đối với MV điểm điều khiển $0\ mv_0$, chênh lệch vector chuyển động được báo hiệu có thể được sử dụng trực tiếp cho chênh lệch $MvdLX$ cho MV điểm điều khiển và, đối với các MV điểm điều khiển khác (mv_1 , mv_2), $MvdLX$, mà là chênh lệch giữa các MV điểm điều khiển, có thể được xác định và được sử dụng dựa trên chênh lệch vector chuyển động được báo hiệu ($mvd1$ và $mvd2$ trên Fig.31) và chênh lệch vector chuyển động được báo hiệu ($mvd0$ trên Fig.31) đối với MV điểm điều khiển $0\ mv_0$.

LX trên Fig.32 có thể biểu thị danh mục tham chiếu X (danh mục ảnh tham chiếu X). $compIdx$ là chỉ số thành phần, và có thể biểu thị thành phần x , y , v.v.. $cpIdx$ có thể biểu thị chỉ số điểm điều khiển. $cpIdx$ có thể có nghĩa là 0 , 1 hoặc 0 , 1 , 2 được minh họa trên Fig.31.

Độ phân giải của chênh lệch vector chuyển động có thể được xem xét theo các giá trị

được minh họa trên Fig.10, Fig.12, Fig.13, v.v.. Ví dụ, khi độ phân giải là R, thì giá trị của $IMvd \cdot R$ có thể được sử dụng cho $IMvd$ trên hình vẽ.

Fig.33 là sơ đồ minh họa cú pháp chênh lệch vector chuyển động theo một phương án của sáng chế.

Dựa vào Fig.33, chênh lệch vector chuyển động có thể được mã hóa theo cách tương tự với cách được mô tả với tham chiếu đến Fig.10. Trong trường hợp này, việc mã hóa có thể được thực hiện riêng theo chỉ số điểm điều khiển $cpIdx$.

Fig.34 là sơ đồ minh họa cấu trúc báo hiệu mức cao hơn theo một phương án của sáng chế.

Theo một phương án của sáng chế, một hoặc nhiều báo hiệu mức cao hơn có thể tồn tại. Báo hiệu mức cao hơn có thể có nghĩa là báo hiệu ở mức cao hơn. Mức cao hơn có thể là đơn vị bao gồm đơn vị bất kỳ. Ví dụ, mức cao hơn của khối hiện thời hoặc đơn vị mã hóa hiện thời có thể bao gồm CTU, phần chia, tấm, nhóm tấm, ảnh, chuỗi, v.v.. Báo hiệu mức cao hơn có thể ảnh hưởng đến mức thấp hơn của mức cao hơn tương ứng. Ví dụ, nếu mức cao hơn là chuỗi, thì nó có thể ảnh hưởng CTU, phần chia, tấm, nhóm tấm, và ảnh các đơn vị mà là các mức thấp hơn của chuỗi. Ở đây, sự ảnh hưởng là báo hiệu mức cao hơn ảnh hưởng đến mã hóa hoặc giải mã đối với các mức thấp hơn.

Ngoài ra, báo hiệu mức cao hơn có thể bao gồm báo hiệu biểu thị chế độ nào có thể được sử dụng. Dựa vào Fig.34, báo hiệu mức cao hơn có thể bao gồm `sps_modeX_enabled_flag`. Theo một phương án, có thể xác định được xem chế độ `modeX` có thể được sử dụng hay không dựa trên `sps_modeX_enabled_flag`. Ví dụ, khi `sps_modeX_enabled_flag` là giá trị nhất định, thì chế độ `modeX` có thể không được sử dụng. Ngoài ra, có thể sử dụng chế độ `modeX` khi `sps_modeX_enabled_flag` là giá trị khác bất kỳ. Ngoài ra, khi `sps_modeX_enabled_flag` là giá trị khác bất kỳ, thì có thể đánh giá xem chế độ `modeX` có được sử dụng hay không dựa trên báo hiệu khác. Ví dụ, giá trị nhất định có thể là 0, và giá trị khác có thể là 1. Tuy nhiên, sáng chế không bị giới hạn ở đó, và giá trị nhất định có thể là 1 và giá trị khác có thể là 0.

Theo một phương án của sáng chế, báo hiệu biểu thị xem phép bù chuyển động afin có thể được sử dụng hay không có thể tồn tại. Ví dụ, báo hiệu này có thể là báo hiệu mức cao hơn. Dựa vào Fig.34, báo hiệu này có thể là `sps_affine_enabled_flag` (cờ cho phép afin). Dựa vào Fig.2 và Fig.7, báo hiệu có thể có nghĩa là tín hiệu được truyền từ bộ mã hóa đến

bộ giải mã thông qua dòng bit. Bộ giải mã có thể phân tích cờ cho phép afin từ dòng bit.

Ví dụ, nếu `sps_affine_enabled_flag` (cờ cho phép afin) là 0, thì cú pháp có thể được giới hạn sao cho phép bù chuyển động afin không được sử dụng. Ngoài ra, khi `sps_affine_enabled_flag` (cờ cho phép afin) là 0, thì `inter_affine_flag` (cờ afin liên ảnh) và `cu_affine_type_flag` (cờ loại afin đơn vị mã hóa) có thể không tồn tại.

Ví dụ, `inter_affine_flag` (cờ afin liên ảnh) có thể là báo hiệu biểu thị xem MC afin (phép bù chuyển động afin) có được sử dụng hay không trong khối. Dựa vào Fig.2 và Fig.7, báo hiệu có thể có nghĩa là tín hiệu được truyền từ bộ mã hóa đến bộ giải mã thông qua dòng bit. Bộ giải mã có thể phân tích `inter_affine_flag` (cờ afin liên ảnh) từ dòng bit.

Ngoài ra, `cu_affine_type_flag` (cờ loại afin đơn vị mã hóa) có thể là báo hiệu biểu thị loại MC afin (phép bù chuyển động afin) nào được sử dụng trong khối. Ngoài ra, ở đây, loại này có thể biểu thị xem nó là mô hình afin 4 thông số hay mô hình afin 6 thông số. Ngoài ra, khi `sps_affine_enabled_flag` (cờ cho phép afin) là 1, thì có thể là phép bù chuyển động afin được sử dụng.

Phép bù chuyển động afin có thể có nghĩa là phép bù chuyển động dựa trên mô hình afin hoặc phép bù chuyển động dựa trên mô hình afin cho phép dự đoán liên ảnh.

Ngoài ra, theo một phương án của sáng chế, báo hiệu biểu thị rằng loại cụ thể của chế độ nào có thể được sử dụng có thể tồn tại. Ví dụ, báo hiệu biểu thị xem loại cụ thể của phép bù chuyển động afin có thể được sử dụng có thể tồn tại. Ví dụ, báo hiệu này có thể là báo hiệu mức cao hơn. Dựa vào Fig.34, báo hiệu này có thể là `sps_affine_type_flag`. Ngoài ra, loại cụ thể có thể có nghĩa là mô hình afin 6 thông số. Ví dụ, nếu `sps_affine_type_flag` là 0, thì cú pháp có thể được giới hạn sao cho mô hình afin 6 thông số không được sử dụng. Ngoài ra, khi `sps_affine_type_flag` là 0, thì `cu_affine_type_flag` có thể không tồn tại. Ngoài ra, nếu `sps_affine_type_flag` là 1, thì mô hình afin 6 thông số có thể được sử dụng. Nếu `sps_affine_type_flag` không tồn tại, thì giá trị của nó có thể được suy ra là bằng 0.

Ngoài ra, theo một phương án của sáng chế, báo hiệu biểu thị rằng loại cụ thể của chế độ nhất định có thể được sử dụng có thể tồn tại khi báo hiệu biểu thị rằng chế độ nhất định có thể được sử dụng tồn tại. Ví dụ, khi giá trị báo hiệu biểu thị xem chế độ nhất định có thể được sử dụng hay không là 1, thì có thể phân tích báo hiệu biểu thị xem loại cụ thể của chế độ nhất định có thể được sử dụng hay không. Ví dụ, khi giá trị báo hiệu biểu thị xem chế độ nhất định có thể được sử dụng hay không là 0, thì không thể phân tích báo hiệu

biểu thị xem loại cụ thể của chế độ nhất định có thể được sử dụng hay không. Ví dụ, báo hiệu biểu thị xem chế độ nhất định có thể được sử dụng hay không có thể bao gồm `sps_affine_enabled_flag` (cờ cho phép afin). Ngoài ra, báo hiệu biểu thị rằng loại cụ thể của chế độ nhất định có thể được sử dụng có thể bao gồm `sps_affine_type_flag` (cờ cho phép afin). Dựa vào Fig.34, khi `sps_affine_enabled_flag` (cờ cho phép afin) là 1, thì có thể phân tích `sps_affine_type_flag`. Ngoài ra, khi `sps_affine_enabled_flag` (cờ cho phép afin) là 0, thì không thể phân tích `sps_affine_type_flag`, và có thể suy ra giá trị của nó là 0.

Ngoài ra, có thể sử dụng độ phân giải vector chuyển động thích ứng (AMVR) như được mô tả ở trên. Có thể sử dụng tập độ phân giải của AMVR theo cách khác tùy thuộc vào trường hợp. Ví dụ, có thể sử dụng tập độ phân giải của AMVR theo cách khác theo chế độ dự đoán. Ví dụ, tập độ phân giải AMVR có thể là khác khi sử dụng phép dự đoán liên ảnh thông thường chẳng hạn như AMVP và khi sử dụng MC afin (phép bù chuyển động afin). Ngoài ra, AMVR được áp dụng vào phép dự đoán liên ảnh thông thường chẳng hạn như AMVP có thể được áp dụng vào chênh lệch vector chuyển động. Theo cách khác, AMVR được áp dụng vào phép dự đoán liên ảnh thông thường chẳng hạn như dự đoán vector chuyển động nâng cao (AMVP) có thể được áp dụng vào bộ dự đoán vector chuyển động. Ngoài ra, AMVR được áp dụng vào MC afin (phép bù chuyển động afin) có thể được áp dụng vào vector chuyển động điểm điều khiển hoặc chênh lệch vector chuyển động điểm điều khiển.

Ngoài ra, theo một phương án của sáng chế, báo hiệu biểu thị xem AMVR có thể được sử dụng hay không có thể tồn tại. Báo hiệu này có thể là báo hiệu mức cao hơn. Dựa vào Fig.34, `sps_amvr_enabled_flag` (cờ cho phép AMVR) có thể tồn tại. Dựa vào Fig.2 và Fig.7, báo hiệu có thể có nghĩa là tín hiệu được truyền từ bộ mã hóa đến bộ giải mã thông qua dòng bit. Bộ giải mã có thể phân tích cờ cho phép AMVR từ dòng bit.

`sps_amvr_enabled_flag` (cờ cho phép AMVR) theo một phương án của sáng chế có thể biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được sử dụng hay không. Ngoài ra, `sps_amvr_enabled_flag` (cờ cho phép AMVR) theo một phương án của sáng chế có thể biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có thể được sử dụng hay không. Ví dụ, theo một phương án của sáng chế, khi `sps_amvr_enabled_flag` (cờ cho phép AMVR) là 1, thì AMVR có thể sử dụng được để mã hóa vector chuyển động. Ngoài ra, theo một phương án của sáng chế, khi `sps_amvr_enabled_flag` (cờ cho phép AMVR) là 1, thì AMVR có thể được sử dụng để mã hóa vector chuyển động. Ngoài ra, khi

`sps_amvr_enabled_flag` (cờ cho phép AMVR) là 1, thì báo hiệu khác để biểu thị độ phân giải nào được sử dụng có thể tồn tại. Ngoài ra, khi `sps_amvr_enabled_flag` (cờ cho phép AMVR) là 0, thì AMVR có thể không sử dụng được để mã hóa vector chuyển động. Ngoài ra, khi `sps_amvr_enabled_flag` (cờ cho phép AMVR) là 0, thì AMVR có thể không sử dụng được để mã hóa vector chuyển động. Theo một phương án của sáng chế, AMVR tương ứng với `sps_amvr_enabled_flag` (cờ cho phép AMVR) có thể có nghĩa là nó được sử dụng cho phép dự đoán liên ảnh thông thường. Ví dụ, AMVR tương ứng với `sps_amvr_enabled_flag` (cờ cho phép AMVR) có thể không có nghĩa là nó được sử dụng cho MC afin. Ngoài ra, việc có sử dụng MC afin (phép bù chuyển động afin) hay không có thể được biểu thị bởi `inter_affine_flag` (cờ afin liên ảnh). Nghĩa là, AMVR tương ứng với `sps_amvr_enabled_flag` (cờ cho phép AMVR) có nghĩa là nó được sử dụng khi `inter_affine_flag` (cờ afin liên ảnh) là 0, hoặc có thể không có nghĩa là nó được sử dụng khi `inter_affine_flag` (cờ afin liên ảnh) là 1

Ngoài ra, theo một phương án của sáng chế, báo hiệu biểu thị xem AMVR có thể được sử dụng hay không cho MC afin (phép bù chuyển động afin) có thể tồn tại. Báo hiệu này có thể là báo hiệu mức cao hơn. Dựa vào Fig.34, báo hiệu `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) biểu thị xem AMVR có thể được sử dụng hay không cho MC afin (phép bù chuyển động afin) có thể tồn tại. Dựa vào Fig.2 và Fig.7, báo hiệu có thể có nghĩa là tín hiệu được truyền từ bộ mã hóa đến bộ giải mã thông qua dòng bit. Bộ giải mã có thể phân tích cờ cho phép AMVR afin từ dòng bit.

`sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) có thể biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được sử dụng hay không cho phép bù chuyển động afin. Ngoài ra, `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) có thể biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có thể được sử dụng hay không cho phép bù chuyển động afin. Theo một phương án, khi `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) là 1, thì AMVR có thể được sử dụng để mã hóa vector chuyển động chế độ liên ảnh afin. Ngoài ra, khi `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) là 1, thì AMVR có thể sử dụng được để mã hóa vector chuyển động chế độ liên ảnh afin. Ngoài ra, khi `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) là 0, thì AMVR không thể được sử dụng để mã hóa vector chuyển động chế độ liên ảnh afin. Khi `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) là 0, thì AMVR có thể không sử

dùng được để mã hóa vector chuyển động chế độ liên ảnh afin.

Ví dụ, khi `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) là 1, thì có thể sử dụng AMVR tương ứng với trường hợp trong đó `inter_affine_flag` (cờ afin liên ảnh) là 1 có thể được sử dụng để mã hóa vector chuyển động chế độ liên ảnh afin. Ngoài ra, khi `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) là 1, thì báo hiệu khác để biểu thị độ phân giải nào được sử dụng có thể tồn tại. Ngoài ra, khi `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) là 0, thì AMVR tương ứng với trường hợp trong đó `inter_affine_flag` (cờ afin liên ảnh) là 1 có thể không được sử dụng để mã hóa vector chuyển động chế độ liên ảnh afin.

Fig.35 là sơ đồ minh họa cấu trúc cú pháp đơn vị mã hóa theo một phương án của sáng chế.

Như được mô tả với tham chiếu đến Fig.34, báo hiệu khác để biểu thị độ phân giải dựa trên báo hiệu mức cao hơn nhờ sử dụng AMVR có thể tồn tại. Dựa vào Fig.34, báo hiệu khác để biểu thị độ phân giải có thể bao gồm `amvr_flag` hoặc `amvr_precision_flag`. `amvr_flag` hoặc `amvr_precision_flag` có thể là thông tin về độ phân giải của chênh lệch vector chuyển động.

Theo một phương án, báo hiệu được biểu thị khi `amvr_flag` là 0 có thể tồn tại. Ngoài ra, khi `amvr_flag` là 1, thì `amvr_precision_flag` có thể tồn tại. Ngoài ra, khi `amvr_flag` là 1, thì độ phân giải cũng có thể được xác định dựa trên `amvr_precision_flag`. Ví dụ, nếu `amvr_flag` là 0, thì có thể là 1/4 độ phân giải. Ngoài ra, nếu `amvr_flag` không tồn tại, thì giá trị `amvr_flag` có thể được suy ra dựa trên `CuPredMode`. Ví dụ, khi `CuPredMode` là `MODE_IBC`, thì giá trị `amvr_flag` có thể được suy ra là bằng 1, và khi `CuPredMode` không phải là `MODE_IBC` hoặc `CuPredMode` (đơn vị mã hóa chế độ dự đoán) là `MODE_INTER`, thì giá trị `amvr_flag` có thể được suy ra là bằng 0.

Ngoài ra, khi `inter_affine_flag` là 0 và `amvr_precision_flag` là 0, thì độ phân giải 1 điểm ảnh có thể được sử dụng. Ngoài ra, khi `inter_affine_flag` là 1 và `amvr_precision_flag` là 0, thì độ phân giải 1/16 điểm ảnh có thể được sử dụng. Ngoài ra, khi `inter_affine_flag` là 0 và `amvr_precision_flag` là 1, thì độ phân giải 4 điểm ảnh có thể được sử dụng. Ngoài ra, khi `inter_affine_flag` là 1 và `amvr_precision_flag` là 1, thì độ phân giải 1 điểm ảnh có thể được sử dụng.

Nếu `amvr_precision_flag` là 0, thì giá trị của nó có thể được suy ra là bằng 0.

Theo một phương án, độ phân giải có thể được áp dụng bởi giá trị MvShift. Ngoài ra, MvShift có thể được xác định bởi amvr_flag và amvr_precision_flag, mà là thông tin về độ phân giải của chênh lệch vector chuyển động. Ví dụ, khi inter_affine_flag là 0, thì giá trị MvShift có thể được xác định như sau.

$$\text{MvShift} = (\text{amvr_flag} + \text{amvr_precision_flag}) \ll 1$$

Ngoài ra, giá trị chênh lệch vector chuyển động Mvd có thể được dịch dựa trên giá trị MvShift. Ví dụ, Mvd (chênh lệch vector chuyển động) có thể được dịch như sau, và theo đó, độ phân giải của AMVR có thể được áp dụng.

$$\text{MvdLX} = \text{MvdLX} \ll (\text{MvShift} + 2)$$

Là một ví dụ khác, khi inter_affine_flag (cờ afin liên ảnh) là 1, thì giá trị MvShift có thể được xác định như sau.

$$\text{MvShift} = \text{amvr_precision_flag} ? (\text{amvr_precision_flag} \ll 1) : (- (\text{amvr_flag} \ll 1))$$

Ngoài ra, giá trị chênh lệch vector chuyển động điểm điều khiển MvdCp có thể được dịch dựa trên giá trị MvShift. MvdCP có thể là chênh lệch vector chuyển động điểm điều khiển hoặc vector chuyển động điểm điều khiển. Ví dụ, MvdCp (chênh lệch vector chuyển động điểm điều khiển) được dịch như sau, và theo đó, độ phân giải của AMVR có thể được áp dụng.

$$\text{MvdCpLX} = \text{MvdCpLX} \ll (\text{MvShift} + 2)$$

Ngoài ra, Mvd hoặc MvdCp có thể là giá trị được báo hiệu bởi mvd_coding.

Dựa vào Fig.35, khi CuPredMode là MODE_IBC, thì amvr_flag, mà là thông tin về độ phân giải của chênh lệch vector chuyển động, có thể không tồn tại. Ngoài ra, khi CuPredMode là MODE_INTER, thì amvr_flag, mà là thông tin về độ phân giải của chênh lệch vector chuyển động, có thể tồn tại, và trong trường hợp này, có thể phân tích amvr_flag nếu điều kiện nhất định được thỏa mãn.

Theo một phương án của sáng chế, có thể xác định được xem có phân tích các phần tử cú pháp liên quan đến AMVR trên cơ sở giá trị báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không. Ví dụ, khi giá trị báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không là 1, thì có thể phân tích các phần tử cú pháp liên quan đến AMVR. Ngoài ra, khi giá trị báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không là 0, thì các phần tử cú pháp liên quan đến AMVR có thể không

được phân tích. Dựa vào Fig.35, khi CuPredMode là MODE_IBC và sps_amvr_enabled_flag là 0, thì amvr_precision_flag (cờ cho phép AMVR), mà là thông tin về độ phân giải của chênh lệch vector chuyển động, có thể không được phân tích. Ngoài ra, khi CuPredMode là MODE_IBC và sps_amvr_enabled_flag (cờ cho phép AMVR) là 1, thì amvr_precision_flag, mà là thông tin về độ phân giải của chênh lệch vector chuyển động, có thể được phân tích. Trong trường hợp này, các điều kiện khác để phân tích có thể được xem xét.

Ví dụ, có thể phân tích amvr_precision_flag khi ít nhất một giá trị khác không tồn tại trong số các giá trị MvdLX (các chênh lệch vector chuyển động). MvdLX có thể là giá trị Mvd cho danh mục tham chiếu LX. Ngoài ra, Mvd (chênh lệch vector chuyển động) có thể được báo hiệu thông qua mvd_coding. LX có thể bao gồm L0 (danh mục ảnh tham chiếu thứ không) và L1 (danh mục ảnh tham chiếu thứ nhất). Ngoài ra, trong MvdLX, thành phần tương ứng với mỗi trong số trục x và trục y có thể tồn tại. Ví dụ, trục x có thể tương ứng với trục nằm ngang của ảnh, và trục y có thể tương ứng với trục thẳng đứng của ảnh. Dựa vào Fig.35, có thể biểu thị rằng [0] và [1] trong MvdLX[x0][y0][0] và MvdLX[x0][y0] là cho các thành phần trục x và trục y, một cách tương ứng. Ngoài ra, khi CuPredMode là MODE_IBC, thì có thể sử dụng chỉ L0. Dựa vào Fig.35, khi 1) sps_amvr_enabled_flag là 1 và 2) MvdL0[x0][y0][0] hoặc MvdL0[x0][y0][1] không phải là 0, thì amvr_precision_flag có thể được phân tích. Ngoài ra, khi 1) sps_amvr_enabled_flag là 0 hoặc 2) MvdL0[x0][y0][0] và MvdL0[x0][y0][1] đều bằng 0, thì amvr_precision_flag có thể không được phân tích.

Ngoài ra, có thể có các trường hợp trong đó CuPredMode không phải là MODE_IBC. Trong trường hợp này, tham chiếu đến Fig.35, khi sps_amvr_enabled_flag là 1, inter_affine_flag (cờ afin liên ảnh) là 0 và ít nhất một giá trị khác không tồn tại trong số các giá trị MvdLX (các chênh lệch vector chuyển động), thì amvr_flag có thể được phân tích. Ở đây, amvr_flag có thể là thông tin về độ phân giải của chênh lệch vector chuyển động. Ngoài ra, như đã được mô tả, sps_amvr_enabled_flag (cờ cho phép AMVR) là 1 có thể biểu thị có sử dụng độ phân giải vi sai vector chuyển động thích ứng. Ngoài ra, inter_affine_flag (cờ afin liên ảnh) là 0 có thể biểu thị rằng phép bù chuyển động afin không được sử dụng cho khối hiện thời. Ngoài ra, như được mô tả với tham chiếu đến Fig.34, các chênh lệch vector chuyển động cho khối hiện thời có thể được hiệu chỉnh trên cơ sở thông tin về độ phân giải của chênh lệch vector chuyển động chẳng hạn như amvr_flag. Điều kiện này có thể được

gọi là điều kiện A.

Ngoài ra, khi `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) là 1, thì `inter_affine_flag` (cờ afin liên ảnh) là 1 và ít nhất một giá trị khác không tồn tại trong số các giá trị `MvdCpLX` (các chênh lệch vector chuyển động điều khiển), thì `amvr_flag` có thể được phân tích. Ở đây, `amvr_flag` có thể là thông tin về độ phân giải của chênh lệch vector chuyển động. Ngoài ra, như đã được mô tả, `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) là 1 có thể biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng có thể được sử dụng cho phép bù chuyển động afin. Ngoài ra, `inter_affine_flag` (cờ afin liên ảnh) là 1 có thể biểu thị có sử dụng phép bù chuyển động afin cho khối hiện thời. Ngoài ra, như được mô tả với tham chiếu đến Fig.34, các chênh lệch vector chuyển động điểm điều khiển đối với khối hiện thời có thể được hiệu chỉnh dựa trên thông tin về độ phân giải của chênh lệch vector chuyển động chẳng hạn như `amvr_flag`. Điều kiện này có thể được gọi là điều kiện B.

Ngoài ra, nếu điều kiện A hoặc điều kiện B được thỏa mãn, thì `amvr_flag` có thể được phân tích. Ngoài ra, nếu cả hai điều kiện A và điều kiện B không được thỏa mãn, thì `amvr_flag`, mà là thông tin về độ phân giải của chênh lệch vector chuyển động, có thể không được phân tích. Nghĩa là, khi 1) `sps_amvr_enabled_flag` (cờ cho phép AMVR) là 0, `inter_affine_flag` (cờ afin liên ảnh) là 1 hoặc `MvdLX` (các chênh lệch vector chuyển động) tất cả bằng 0, và 2) `sps_affine_amvr_enabled_flag` là 0, `inter_affine_flag` (cờ afin liên ảnh) là 0, hoặc các giá trị `MvdCpLX` (các chênh lệch vector chuyển động điểm điều khiển) tất cả bằng 0, thì `amvr_flag` có thể không được phân tích.

Cũng có thể xác định xem có phân tích `amvr_precision_flag` hay không dựa trên giá trị `amvr_flag`. Ví dụ, khi giá trị `amvr_flag` là 1, thì `amvr_precision_flag` có thể được phân tích. Ngoài ra, khi giá trị `amvr_flag` là 0, thì `amvr_precision_flag` có thể không được phân tích.

Ngoài ra, `MvdCpLX` (các chênh lệch vector chuyển động điều khiển) có thể có nghĩa là chênh lệch đối với vector chuyển động điểm điều khiển. Ngoài ra, `MvdCpLX` (các chênh lệch vector chuyển động điều khiển) có thể được báo hiệu thông qua `mvd_coding`. `LX` có thể bao gồm `L0` (danh mục ảnh tham chiếu thứ không) và `L1` (danh mục ảnh tham chiếu thứ nhất). Ngoài ra, trong `MvdCpLX`, các thành phần tương ứng với các vector chuyển động điểm điều khiển 0, 1, 2, v.v. có thể tồn tại. Ví dụ, các vector chuyển động điểm điều khiển 0, 1, 2, v.v. có thể là các vector chuyển động điểm điều khiển tương ứng với các vị trí được

thiết lập trước dựa trên khối hiện thời.

Dựa vào Fig.35, thì có thể biểu thị rằng [0], [1] và [2] trong $MvdCpLX[x0][y0][0][[]]$, $MvdCpLX[x0][y0][1][[]]$ và $MvdCpLX[x0][y0][2][[]]$ lần lượt tương ứng với các vector chuyển động điểm điều khiển 0, 1 và 2. Ngoài ra, giá trị $MvdCpLX$ (các chênh lệch vector chuyển động điểm điều khiển) tương ứng với vector chuyển động điểm điều khiển 0 có thể được sử dụng cho các vector chuyển động điểm điều khiển khác. Ví dụ, vector chuyển động điểm điều khiển 0 có thể được sử dụng như được sử dụng trên Fig.47. Ngoài ra, trong $MvdCpLX$ (các chênh lệch vector chuyển động điểm điều khiển), các thành phần mà lần lượt tương ứng với trục x và trục y có thể tồn tại. Ví dụ, trục x có thể tương ứng với trục nằm ngang của ảnh, và trục y có thể tương ứng với trục thẳng đứng của ảnh. Dựa vào Fig.35, thì có thể biểu thị rằng [0] và [1] trong $MvdCpLX[x0][y0][[]][0]$ và $MvdCpLX[x0][y0][[]][1]$ là cho các thành phần trục x và trục y, một cách tương ứng.

Fig.36 là sơ đồ minh họa cấu trúc báo hiệu mức cao hơn theo một phương án của sáng chế.

Báo hiệu mức cao hơn như được mô tả với tham chiếu đến các hình vẽ từ Fig.34 đến Fig.35 có thể tồn tại. Ví dụ, `sps_affine_enabled_flag` (cờ cho phép afin), `sps_affine_amvr_enabled_flag`, `sps_amvr_enabled_flag` (cờ cho phép AMVR), `sps_affine_type_flag`, v.v. có thể tồn tại.

Theo một phương án của sáng chế, báo hiệu mức cao hơn được mô tả ở trên có thể có sự phụ thuộc phân tích. Ví dụ, có thể xác định được xem có phân tích báo hiệu mức cao hơn khác hay không dựa trên giá trị báo hiệu mức cao hơn.

Theo một phương án của sáng chế, có thể xác định xem AMVR afin có thể được sử dụng hay không dựa trên việc MC afin (phép bù chuyển động afin) có thể được sử dụng hay không. Ví dụ, có thể xác định được xem AMVR afin có thể được sử dụng hay không trên cơ sở báo hiệu mức cao hơn biểu thị xem MC afin có thể được sử dụng hay không. Cụ thể hơn là, có thể xác định xem có phân tích báo hiệu mức cao hơn hay không biểu thị xem AMVR afin có thể được sử dụng hay không trên cơ sở báo hiệu mức cao hơn biểu thị xem MC afin có thể được sử dụng hay không.

Theo một phương án, có thể sử dụng AMVR afin khi MC afin có thể được sử dụng. Ngoài ra, khi MC afin không thể được sử dụng, thì AMVR afin có thể không sử dụng được.

Cụ thể hơn là, nếu báo hiệu mức cao hơn biểu thị xem MC afin có thể được sử dụng

hay không là 1, thì AMVR afin có thể sử dụng được. Trong trường hợp này, có thể là báo hiệu khác tồn tại. Ngoài ra, khi báo hiệu mức cao hơn biểu thị xem MC afin có thể được sử dụng hay không là 0, thì AMVR afin có thể không sử dụng được. Ví dụ, khi báo hiệu mức cao hơn biểu thị xem MC afin có thể được sử dụng hay không là 1, thì báo hiệu mức cao hơn biểu thị xem AMVR afin có thể được sử dụng hay không có thể được phân tích. Ngoài ra, khi báo hiệu mức cao hơn biểu thị xem MC afin có thể được sử dụng hay không là 0, thì báo hiệu mức cao hơn biểu thị xem AMVR afin có thể được sử dụng hay không có thể không được phân tích. Ngoài ra, khi báo hiệu mức cao hơn biểu thị xem AMVR afin có thể được sử dụng hay không không tồn tại, thì giá trị của nó có thể được suy ra. Ví dụ, giá trị này có thể được suy ra là bằng 0. Là một ví dụ khác, giá trị này có thể được suy ra trên cơ sở báo hiệu mức cao hơn biểu thị xem MC afin có thể được sử dụng hay không. Là một ví dụ khác, giá trị này có thể được suy ra trên cơ sở báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không.

Theo một phương án, AMVR afin có thể là AMVR được sử dụng cho MC afin được mô tả với tham chiếu đến các hình vẽ từ Fig.34 đến Fig.35. Ví dụ, báo hiệu mức cao hơn biểu thị xem MC afin có thể được sử dụng hay không có thể là `sps_affine_enabled_flag` (cờ cho phép afin). Ngoài ra, báo hiệu mức cao hơn biểu thị xem AMVR afin có thể được sử dụng hay không có thể là `sps_affine_amvr_enabled_flag`.

Dựa vào Fig.36, khi `sps_affine_enabled_flag` (cờ cho phép afin) là 1, thì `sps_affine_amvr_enabled_flag` có thể được phân tích. Ngoài ra, khi `sps_affine_enabled_flag` (cờ cho phép afin) là 0, thì `sps_affine_amvr_enabled_flag` có thể không được phân tích. Ngoài ra, khi `sps_affine_amvr_enabled_flag` không tồn tại, thì giá trị của nó có thể được suy ra là bằng 0.

Điều này có thể là vì, theo phương án của sáng chế, AMVR afin có thể có nghĩa khi sử dụng MC afin.

Fig.37 là sơ đồ minh họa cấu trúc báo hiệu mức cao hơn theo một phương án của sáng chế.

Báo hiệu mức cao hơn như được mô tả với tham chiếu đến các hình vẽ từ Fig.34 đến Fig.35 có thể tồn tại. Ví dụ, `sps_affine_enabled_flag` (cờ cho phép afin), `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin), `sps_amvr_enabled_flag` (cờ cho phép AMVR) và `sps_affine_type_flag`, v.v. có thể tồn tại.

Theo một phương án của sáng chế, các báo hiệu mức cao hơn có thể có sự phụ thuộc phân tích. Ví dụ, có thể xác định được xem có phân tích báo hiệu mức cao hơn khác hay không dựa trên giá trị báo hiệu mức cao hơn nhất định.

Theo một phương án của sáng chế, có thể xác định xem AMVR affin có thể được sử dụng hay không trên cơ sở việc AMVR có thể được sử dụng hay không. Ví dụ, có thể xác định được xem AMVR affin có thể được sử dụng hay không trên cơ sở báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không. Cụ thể hơn là, có thể xác định xem có phân tích báo hiệu mức cao hơn hay không biểu thị xem AMVR affin có thể được sử dụng hay không trên cơ sở báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không.

Theo một phương án, có thể sử dụng AMVR affin khi AMVR có thể được sử dụng. Ngoài ra, khi AMVR không thể được sử dụng, thì AMVR affin có thể không sử dụng được.

Cụ thể hơn là, khi báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không là 1, thì AMVR affin có thể sử dụng được. Trong trường hợp này, có thể là báo hiệu khác tồn tại. Ngoài ra, khi báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không là 0, thì AMVR affin có thể không sử dụng được. Ví dụ, khi báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không là 1, thì báo hiệu mức cao hơn biểu thị xem AMVR affin có thể được sử dụng hay không có thể được phân tích. Ngoài ra, khi báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không là 0, thì báo hiệu mức cao hơn biểu thị xem AMVR affin có thể được sử dụng hay không có thể không được phân tích. Ngoài ra, nếu không có báo hiệu mức cao hơn nào biểu thị xem AMVR affin có thể được sử dụng hay không, thì giá trị của nó có thể được suy ra. Ví dụ, giá trị này có thể được suy ra là bằng 0. Là một ví dụ khác, giá trị này có thể được suy ra trên cơ sở báo hiệu mức cao hơn biểu thị xem MC affin có thể được sử dụng hay không. Là một ví dụ khác, giá trị này có thể được suy ra trên cơ sở báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không.

Theo một phương án, AMVR affin có thể là AMVR được sử dụng cho MC affin (phép bù chuyển động affin) được mô tả với tham chiếu đến các hình vẽ từ Fig.34 đến Fig.35. Ví dụ, báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không có thể là `sps_amvr_enabled_flag` (cờ cho phép AMVR). Ngoài ra, báo hiệu mức cao hơn biểu thị xem AMVR affin có thể được sử dụng hay không có thể là `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR affin).

Dựa vào Fig.37(a), khi `sps_amvr_enabled_flag` (cờ cho phép AMVR) là 1, thì `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) có thể được phân tích. Ngoài ra, khi `sps_amvr_enabled_flag` (cờ cho phép AMVR) là 0, thì `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) có thể không được phân tích. Ngoài ra, khi `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) không tồn tại, thì giá trị của nó có thể được suy ra là bằng 0.

Điều này có thể là vì, theo phương án của sáng chế, việc độ phân giải thích ứng có hữu hiệu hay không có thể thay đổi theo chuỗi.

Ngoài ra, việc AMVR afin có thể được sử dụng hay không có thể được xác định bằng cách xem xét cả hai việc MC afin có thể được sử dụng hay không và việc AMVR có thể được sử dụng hay không. Ví dụ, có thể xác định xem có phân tích báo hiệu mức cao hơn hay không biểu thị xem AMVR afin có thể được sử dụng hay không trên cơ sở báo hiệu mức cao hơn biểu thị xem MC afin có thể được sử dụng hay không và báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không. Theo một phương án, khi cả hai báo hiệu mức cao hơn biểu thị xem MC afin có thể được sử dụng hay không và báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không là 1, thì báo hiệu mức cao hơn biểu thị xem AMVR afin có thể được sử dụng hay không có thể được phân tích. Ngoài ra, khi báo hiệu mức cao hơn biểu thị xem MC afin có thể được sử dụng hay không hoặc báo hiệu mức cao hơn biểu thị xem AMVR có thể được sử dụng hay không là 0, thì báo hiệu mức cao hơn biểu thị xem AMVR afin có thể được sử dụng hay không có thể không được phân tích. Ngoài ra, nếu báo hiệu mức cao hơn biểu thị xem AMVR afin có thể được sử dụng hay không không tồn tại, thì giá trị của nó có thể được suy ra.

Dựa vào Fig.37(b), khi cả hai `sps_affine_enabled_flag` (cờ cho phép afin) và `sps_amvr_enabled_flag` (cờ cho phép AMVR) là 1, thì `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) có thể được phân tích. Ngoài ra, khi ít nhất một trong số `sps_affine_enabled_flag` (cờ cho phép afin) và `sps_amvr_enabled_flag` (cờ cho phép AMVR) là 0, thì `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) có thể không được phân tích. Ngoài ra, khi `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) không tồn tại, thì giá trị có thể được suy ra là bằng 0.

Cụ thể hơn là, tham chiếu đến Fig.37(b), có thể xác định được xem phép bù chuyển động afin có thể được sử dụng hay không dựa trên `sps_affine_enabled_flag` (cờ cho phép afin), ở dòng 3701. Như đã được mô tả với tham chiếu đến các hình vẽ từ Fig.34 đến Fig.35,

khi `sps_affine_enabled_flag` (cờ cho phép afin) là 1, thì có thể có nghĩa là phép bù chuyển động afin có thể được sử dụng. Ngoài ra, khi `sps_affine_enabled_flag` (cờ cho phép afin) là 0, thì có thể có nghĩa là phép bù chuyển động afin không thể được sử dụng.

Khi xác định được rằng phép bù chuyển động afin được sử dụng ở dòng 3701 của Fig.37(b), thì có thể xác định được xem độ phân giải vi sai vector chuyển động thích ứng được sử dụng hay không dựa trên `sps_amvr_enabled_flag` (cờ cho phép AMVR), ở dòng 3702. Như đã được mô tả với tham chiếu đến các hình vẽ từ Fig.34 đến Fig.35, khi `sps_amvr_enabled_flag` (cờ cho phép AMVR) là 1, thì có thể có nghĩa là độ phân giải vi sai vector chuyển động thích ứng được sử dụng. Khi `sps_amvr_enabled_flag` (cờ cho phép AMVR) là 0, thì có thể có nghĩa là độ phân giải vi sai vector chuyển động thích ứng không được sử dụng.

Khi xác định được rằng phép bù chuyển động afin không được sử dụng ở dòng 3701, thì có thể không được xác định xem độ phân giải vi sai vector chuyển động thích ứng được sử dụng hay không, trên cơ sở `sps_amvr_enabled_flag` (cờ cho phép AMVR). Nghĩa là, dòng 3702 của Fig.37(b) có thể không được thực hiện. Cụ thể là, khi phép bù chuyển động afin không được sử dụng, thì `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) có thể không được truyền từ bộ mã hóa đến bộ giải mã. Nghĩa là, bộ giải mã có thể không nhận `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) và `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) có thể không được phân tích bởi bộ giải mã. Trong trường hợp này, vì `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) không tồn tại, nên giá trị này có thể được suy ra là bằng 0. Như đã được mô tả, khi `sps_affine_amvr_enabled_flag` (AMVR afin enablex cờ) là 0, thì có thể biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng không thể được sử dụng cho phép bù chuyển động afin.

Khi xác định được rằng độ phân giải vi sai vector chuyển động thích ứng được sử dụng ở dòng 3702 của Fig.37(b), thì `sps_affine_amvr_enabled_flag` biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có thể được sử dụng hay không cho phép bù chuyển động afin có thể được phân tích từ dòng bit, ở dòng 3703.

Khi xác định được rằng độ phân giải vi sai vector chuyển động thích ứng không được sử dụng ở dòng 3702, thì `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) có thể không được phân tích từ dòng bit. Nghĩa là, dòng 3703 của Fig.37(b) có thể không được thực hiện. Cụ thể hơn là, khi phép bù chuyển động afin được sử dụng và độ phân giải vi sai

vector chuyển động thích ứng không được sử dụng, thì `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) có thể không được truyền từ bộ mã hóa đến bộ giải mã. Nghĩa là, bộ giải mã có thể không nhận `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin). Bộ giải mã có thể không phân tích `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) từ dòng bit. Trong trường hợp này, vì `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) không tồn tại, nên giá trị này có thể được suy ra là bằng 0. Như đã được mô tả, khi `sps_affine_amvr_enabled_flag` (cờ cho phép AMVR afin) là 0, thì có thể biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng không thể được sử dụng cho phép bù chuyển động afin.

Bằng cách kiểm tra `sps_affine_enabled_flag` (cờ cho phép afin) đầu tiên và `sps_amvr_enabled_flag` (cờ cho phép AMVR) được kiểm tra tiếp theo như được minh họa trên Fig.37(b), quy trình không cần thiết có thể được giảm bớt và hiệu quả có thể được tăng lên. Ví dụ, khi `sps_amvr_enabled_flag` (cờ cho phép AMVR) được kiểm tra đầu tiên và `sps_affine_enabled_flag` (cờ cho phép afin) được kiểm tra tiếp theo, thì `sps_affine_enabled_flag` (cờ cho phép afin) có thể cần được kiểm tra một lần nữa để suy ra `sps_affine_type_flag` ở dòng thứ 7. Tuy nhiên, bằng cách kiểm tra `sps_affine_enabled_flag` (cờ cho phép afin) đầu tiên và `sps_amvr_enabled_flag` (cờ cho phép AMVR) tiếp theo, quy trình không cần thiết này có thể được giảm bớt.

Fig.38 là sơ đồ minh họa cấu trúc cú pháp đơn vị mã hóa theo một phương án của sáng chế.

Như được mô tả với tham chiếu đến Fig.35, có thể xác định xem có phân tích cú pháp liên quan đến AMVR hay không dựa trên việc ít nhất một giá trị khác không có tồn tại trong số `MvdLX` hoặc `MvdCpLX` hay không. Tuy nhiên, `MvdLX` (chênh lệch vector chuyển động) hoặc `MvdCpLX` (chênh lệch vector chuyển động điểm điều khiển) có thể khác tùy thuộc vào danh mục tham chiếu (danh mục ảnh tham chiếu) nào được sử dụng, bao nhiêu thông số mà mô hình afin sử dụng, v.v.. Nếu giá trị ban đầu của `MvdLX` hoặc `MvdCpLX` không phải là 0, vì `MvdLX` hoặc `MvdCpLX` không được sử dụng trong khối hiện thời không phải là 0, nên phần tử cú pháp liên quan đến AMVR không cần thiết được báo hiệu, và sự không phù hợp giữa bộ mã hóa và bộ giải mã có thể được tạo ra. Các hình vẽ từ Fig.38 đến Fig.39 có thể mô tả phương pháp không tạo ra sự không phù hợp giữa bộ mã hóa và bộ giải mã.

Theo một phương án, `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) có thể

biểu thị danh mục tham chiếu nào được sử dụng hoặc hướng của phép dự đoán là hướng nào. Ví dụ, `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) có thể là giá trị của `PRED_L0`, `PRED_L1` hoặc `PRED_BI`. Nếu `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) là `PRED_L0`, thì chỉ danh mục tham chiếu 0 (danh mục ảnh tham chiếu thứ không) có thể được sử dụng. Ngoài ra, khi `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) là `PRED_L1`, thì chỉ danh mục tham chiếu 1 (danh mục ảnh tham chiếu thứ nhất) có thể được sử dụng. Ngoài ra, khi `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) là `PRED_BI`, thì cả hai danh mục tham chiếu 0 (danh mục ảnh tham chiếu thứ không) và danh mục tham chiếu 1 (danh mục ảnh tham chiếu thứ nhất) có thể được sử dụng. Khi `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) là `PRED_L0` hoặc `PRED_L1`, thì nó có thể là phép dự đoán theo một hướng. Ngoài ra, khi `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) là `PRED_BI`, thì nó có thể là phép dự đoán theo hai hướng.

Cũng có thể xác định mô hình afin cần được sử dụng trên cơ sở giá trị `MotionModelIdc`. Cũng có thể xác định xem có sử dụng MC afin hay không trên cơ sở giá trị `MotionModelIdc`. Ví dụ, `MotionModelIdc` có thể biểu thị chuyển động tịnh tiến, chuyển động afin 4 thông số hoặc chuyển động afin 6 thông số. Ví dụ, khi giá trị `MotionModelIdc` là 0, 1 và 2, thì chuyển động tịnh tiến, chuyển động afin 4 thông số và chuyển động afin 6 thông số có thể lần lượt được biểu thị. Ngoài ra, theo một phương án, `MotionModelIdc` có thể được xác định trên cơ sở `inter_affine_flag` (cờ afin liên ảnh) và `cu_affine_type_flag`. Ví dụ, khi `merge_flag` là 0 (không phải chế độ hợp nhất), thì `MotionModelIdc` có thể được xác định trên cơ sở `inter_affine_flag` (cờ afin liên ảnh) và `cu_affine_type_flag`. Ví dụ, `MotionModelIdc` có thể là $(inter_affine_flag + cu_affine_type_flag)$. Theo một phương án khác, `MotionModelIdc` có thể được xác định bởi `merge_subblock_flag`. Ví dụ, khi `merge_flag` là 1 (chế độ hợp nhất), thì `MotionModelIdc` có thể được xác định bởi `merge_subblock_flag`. Ví dụ, giá trị `MotionModelIdc` có thể được thiết lập thành giá trị `merge_subblock_flag`.

Ví dụ, khi `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) là `PRED_L0` hoặc `PRED_BI`, thì có thể sử dụng giá trị tương ứng với L0 trong `MvdLX` (chênh lệch vector chuyển động) hoặc `MvdCpLX` (chênh lệch vector chuyển động điểm điều khiển). Do đó, khi phân tích cú pháp liên quan đến AMVR, thì `MvdL0` hoặc `MvdCpL0` có thể được xem xét chỉ khi `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) là `PRED_L0` hoặc `PRED_BI`. Nghĩa là, khi `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) là

PRED_L1, thì không thể xem xét MvdL0 (chênh lệch vector chuyển động đối với danh mục ảnh tham chiếu thứ không) hoặc MvdCpL0 (chênh lệch vector chuyển động điểm điều khiển đối với danh mục ảnh tham chiếu thứ không).

Ngoài ra, khi inter_pred_idc (thông tin về danh mục ảnh tham chiếu) là PRED_L1 hoặc PRED_BI, thì có thể sử dụng giá trị tương ứng với L1 trong MvdLX hoặc MvdCpLX. Do đó, khi phân tích cú pháp liên quan đến AMVR, thì MvdL1 (chênh lệch vector chuyển động đối với danh mục ảnh tham chiếu thứ nhất) hoặc MvdCpL1 (chênh lệch vector chuyển động điểm điều khiển đối với danh mục ảnh tham chiếu thứ nhất) có thể được xem xét chỉ khi inter_pred_idc (thông tin về danh mục ảnh tham chiếu) là PRED_L1 hoặc PRED_BI. Nghĩa là, khi inter_pred_idc (thông tin về danh mục ảnh tham chiếu) là PRED_L0, thì không thể xem xét MvdL1 hoặc MvdCpL1.

Dựa vào Fig.38, có thể xác định, đối với MvdL0 và MvdCpL0, xem có phân tích cú pháp liên quan đến AMVR hay không theo việc các giá trị của chúng có không phải là 0 hay không chỉ khi inter_pred_idc không phải là PRED_L1. Nghĩa là, khi inter_pred_idc là PRED_L1, thậm chí nếu giá trị khác không tồn tại trong MvdL0 hoặc giá trị khác không tồn tại trong MvdCpL0, thì cú pháp liên quan đến AMVR có thể không được phân tích.

Ngoài ra, khi MotionModelIdc là 1, thì có thể xem xét chỉ MvdCpLX[x0][y0][0][], MvdCpLX[x0][y0][1][] trong số MvdCpLX[x0][y0][0][] và MvdCpLX[x0][y0][1][], MvdCpLX[x0][y0][2][]. Nghĩa là, khi MotionModelIdc là 1, thì không thể xem xét MvdCpLX[x0][y0][2][]. Ví dụ, khi MotionModelIdc là 1, thì việc giá trị khác không có tồn tại trong MvdCpLX[x0][y0][2][] hay không có thể không ảnh hưởng đến việc có phân tích cú pháp liên quan đến AMVR hay không.

Ngoài ra, khi MotionModelIdc là 2, thì có thể xem xét tất cả trong số MvdCpLX[x0][y0][0][], MvdCpLX[x0][y0][1][] và MvdCpLX[x0][y0][2][]. Nghĩa là, khi MotionModelIdc là 2, thì có thể xem xét MvdCpLX[x0][y0][2][].

Ngoài ra, theo phương án ở trên, giá trị được biểu diễn dưới dạng MotionModelIdc 1 hoặc 2 có thể được biểu diễn dưới dạng cu_affine_type_flag là 0 hoặc 1. Điều này có thể là vì nó có thể xác định xem có sử dụng MC afin hay không. Ví dụ, việc có sử dụng MC afin hay không có thể được xác định thông qua inter_affine_flag (cờ afin liên ảnh).

Dựa vào Fig.38, việc giá trị khác không có tồn tại trong MvdCpLX[x0][y0][2][] hay không có thể được xem xét chỉ khi MotionModelIdc là 2. Khi MvdCpLX[x0][y0][0][] và

MvdCpLX[x0][y0][1][] tất cả bằng 0 và giá trị khác không tồn tại trong MvdCpLX[x0][y0][2] (theo phương án ở trên, chỉ một trong số L0 và L1 có thể được xem xét bằng cách phân chia L0 và L1 ở đây), nếu MotionModelIdc không phải là 2, thì cú pháp liên quan đến AMVR có thể không được phân tích.

Fig.39 là sơ đồ minh họa thiết lập giá trị mặc định MVD theo một phương án của sáng chế.

Như được mô tả ở trên, MvdLX (chênh lệch vector chuyển động) hoặc MvdCpLX (chênh lệch vector chuyển động điểm điều khiển) có thể được báo hiệu thông qua mvd_coding. Ngoài ra, giá trị lMvd có thể được báo hiệu thông qua mvd_coding, và MvdLX hoặc MvdCpLX có thể được thiết lập là giá trị lMvd. Dựa vào Fig.39, khi MotionModelIdc là 0, thì MvdLX có thể được thiết lập thông qua giá trị lMvd. Ngoài ra, khi MotionModelIdc không phải là 0, thì MvdCpLX có thể được thiết lập thông qua giá trị lMvd. Ngoài ra, tùy thuộc vào giá trị refList, có thể xác định hoạt động nào tương ứng với bất kỳ trong số LX cần được thực hiện.

Ngoài ra, mvd_coding như được mô tả với tham chiếu đến Fig.10 hoặc Fig.33, Fig.34 và Fig.35 có thể tồn tại. Ngoài ra, mvd_coding có thể bao gồm bước phân tích hoặc xác định abs_mvd_greater0_flag, abs_mvd_greater1_flag, abs_mvd_minus2, mvd_sign_flag, v.v.. Ngoài ra, lMvd có thể được xác định thông qua abs_mvd_greater0_flag, abs_mvd_greater1_flag, abs_mvd_minus2, mvd_sign_flag, v.v..

Dựa vào Fig.39, lMvd có thể được thiết lập như sau.

$$lMvd = abs_mvd_greater0_flag * (abs_mvd_minus2 + 2) * (1 - 2 * mvd_sign_flag)$$

Có thể thiết lập giá trị mặc định của MvdLX (chênh lệch vector chuyển động) hoặc MvdCpLX (chênh lệch vector chuyển động điểm điều khiển) thành giá trị được thiết lập trước. Theo một phương án của sáng chế, có thể thiết lập giá trị mặc định của MvdLX hoặc MvdCpLX thành 0. Theo cách khác, có thể thiết lập giá trị mặc định của lMvd thành giá trị được thiết lập trước. Theo cách khác, có thể thiết lập giá trị mặc định của phần tử cú pháp có liên quan sao cho giá trị của lMvd, MvdLX hoặc MvdCpLX trở thành giá trị được thiết lập trước. Giá trị mặc định của phần tử cú pháp có thể có nghĩa là giá trị cần được suy ra khi phần tử cú pháp không tồn tại. Giá trị được thiết lập trước có thể là 0.

Theo một phương án của sáng chế, abs_mvd_greater0_flag có thể biểu thị xem giá trị tuyệt đối của MVD có lớn hơn 0 hay không. Ngoài ra, theo một phương án của sáng chế,

khi `abs_mvd_greater0_flag` không tồn tại, thì giá trị của nó có thể được suy ra là bằng 0. Trong trường hợp này, giá trị `IMvd` có thể được thiết lập thành 0. Ngoài ra, trong trường hợp này, giá trị `MvdLX` hoặc `MvdCpLX` có thể được thiết lập thành 0.

Theo cách khác, theo một phương án của sáng chế, khi giá trị `IMvd`, `MvdLX` hoặc `MvdCpLX` không được thiết lập, thì giá trị của nó có thể được thiết lập là giá trị được thiết lập trước. Ví dụ, giá trị này có thể được thiết lập thành 0.

Ngoài ra, theo một phương án của sáng chế, khi `abs_mvd_greater0_flag` không tồn tại, thì giá trị `IMvd`, `MvdLX` hoặc `MvdCpLX` tương ứng có thể được thiết lập thành 0.

Ngoài ra, `abs_mvd_greater1_flag` có thể biểu thị xem giá trị tuyệt đối của `MVD` có lớn hơn 1 hay không. Ngoài ra, khi `abs_mvd_greater1_flag` không tồn tại, thì giá trị của nó có thể được suy ra là bằng 0.

Ngoài ra, $(\text{abs_mvd_minus2} + 2)$ có thể biểu thị giá trị tuyệt đối của `MVD`. Ngoài ra, khi giá trị của `abs_mvd_minus2` không tồn tại, thì giá trị này có thể được suy ra là bằng -1.

Ngoài ra, `mvd_sign_flag` có thể biểu thị dấu của `MVD`. Khi `mvd_sign_flag` là 0 và 1, thì có thể biểu thị rằng các `MVD` tương ứng lần lượt có các giá trị dương và âm. Nếu `mvd_sign_flag` không tồn tại, thì giá trị này có thể được suy ra là bằng 0.

Fig.40 là sơ đồ minh họa thiết lập giá trị mặc định `MVD` theo một phương án của sáng chế.

Bằng cách thiết lập giá trị ban đầu của `MvdLX` hoặc `MvdCpLX` thành 0, giá trị `MvdLX` (các chênh lệch vector chuyển động) hoặc `MvdCpLX` (các chênh lệch vector chuyển động điểm điều khiển) có thể được khởi tạo như được mô tả với tham chiếu đến Fig.39 để không tạo ra sự không phù hợp giữa bộ mã hóa và bộ giải mã. Ngoài ra, trong trường hợp này, giá trị được khởi tạo có thể là 0. Phương án của Fig.40 mô tả điều này một cách chi tiết hơn.

Theo một phương án của sáng chế, các giá trị `MvdLX` (các chênh lệch vector chuyển động) hoặc `MvdCpLX` (các chênh lệch vector chuyển động điểm điều khiển) có thể được khởi tạo thành giá trị được thiết lập trước. Ngoài ra, vị trí khởi tạo có thể đứng trước các vị trí mà các phần tử cú pháp liên quan đến `AMVR` được phân tích. Các phần tử cú pháp liên quan đến `AMVR` có thể bao gồm `amvr_flag`, `amvr_precision_flag`, v.v. của Fig.40. `amvr_flag` hoặc `amvr_precision_flag` có thể là thông tin về độ phân giải của chênh lệch

vector chuyển động.

Ngoài ra, độ phân giải của MVD (chênh lệch vector chuyển động) hoặc MV (vector chuyển động) hoặc độ phân giải, mà được báo hiệu, của MVD hoặc MV có thể được xác định bởi các phần tử cú pháp liên quan đến AMVR. Ngoài ra, giá trị được thiết lập trước được sử dụng để khởi tạo theo các phương án của sáng chế có thể là 0.

Vì các giá trị MvdLX và MvdCpLX giữa bộ mã hóa và bộ giải mã có thể giống nhau khi phân tích các phần tử cú pháp liên quan đến AMVR bằng cách thực hiện việc khởi tạo, sự không phù hợp giữa bộ mã hóa và bộ giải mã có thể không được tạo ra. Ngoài ra, bằng cách thực hiện việc khởi tạo để được thiết lập thành giá trị 0, các phần tử cú pháp liên quan đến AMVR có thể không được bao gồm trong dòng bit một cách không cần thiết.

Theo một phương án của sáng chế, MvdLX có thể được xác định cho danh mục tham chiếu (L0, L1, v.v.), thành phần x hoặc thành phần y, v.v.. Ngoài ra, MvdCpLX có thể được xác định cho danh mục tham chiếu (L0, L1, v.v.), thành phần x hoặc thành phần y, điểm điều khiển 0, 1 hoặc 2, v.v..

Theo một phương án của sáng chế, có thể khởi tạo cả hai giá trị MvdLX hoặc MvdCpLX. Ngoài ra, vị trí khởi tạo có thể đứng trước vị trí mà mvd_coding cho MvdLX hoặc MvdCpLX tương ứng được thực hiện. Ví dụ, có thể khởi tạo MvdLX hoặc MvdCpLX tương ứng với L0 thậm chí trong khối dự đoán nhờ chỉ sử dụng L0. Kiểm tra điều kiện có thể cần thiết để khởi tạo chỉ các giá trị mà việc khởi tạo là cần thiết, nhưng bằng cách thực hiện việc khởi tạo không phân biệt theo cách này, gánh nặng của kiểm tra điều kiện có thể được giảm bớt.

Theo một phương án khác của sáng chế, có thể khởi tạo giá trị tương ứng với giá trị MvdLX hoặc MvdCpLX, tùy theo giá trị nào không được sử dụng. Ở đây, giá trị đang không được sử dụng có thể có nghĩa là giá trị không được sử dụng trong khối hiện thời. Ví dụ, giá trị tương ứng với danh mục tham chiếu mà đang không được sử dụng trong số MvdLX hoặc MvdCpLX có thể được khởi tạo. Ví dụ, khi L0 không được sử dụng, thì giá trị tương ứng với MvdL0 hoặc MvdCpL0 có thể được khởi tạo. L0 có thể không được sử dụng khi inter_pred_idc là PRED_L1. Ngoài ra, khi L1 không được sử dụng, thì giá trị tương ứng với MvdL1 hoặc MvdCpL1 có thể được khởi tạo. L1 có thể không được sử dụng khi inter_pred_idc là PRED_L0. L0 có thể được sử dụng khi inter_pred_idc là PRED_L0 hoặc PRED_BI và L1 có thể được sử dụng khi inter_pred_idc là PRED_L1 hoặc PRED_BI. Dựa vào Fig.40, khi inter_pred_idc là PRED_L1, thì MvdL0[x0][y0][0],

MvdL0[x0][y0][1], MvdCpL0[x0][y0][0][0], MvdCpL0[x0][y0][0][1],
MvdCpL0[x0][y0][1][0], MvdCpL0[x0][y0][1][1], MvdCpL0[x0][y0][2][0] và
MvdCpL0[x0][y0][2][1] có thể được khởi tạo. Ngoài ra, giá trị được khởi tạo có thể là 0.
Ngoài ra, khi inter_pred_idc là PRED_L0, thì MvdL1[x0][y0][0], MvdL1[x0][y0][1],
MvdCpL1[x0][y0][0][0], MvdCpL1[x0][y0][0][1], MvdCpL1[x0][y0][1][0],
MvdCpL1[x0][y0][1][1], MvdCpL1[x0][y0][2][0] và MvdCpL1[x0][y0][2][1] có thể được
khởi tạo. Ngoài ra, giá trị được khởi tạo có thể là 0.

MvdLX[x][y][compIdx] có thể là chênh lệch vector chuyển động đối với vị trí (x, y) đối với danh mục tham chiếu LX và chỉ số thành phần compIdx. MvdCpLX[x][y][cpIdx][compIdx] có thể là chênh lệch vector chuyển động đối với danh mục tham chiếu LX. Ngoài ra, MvdCpLX[x][y][cpIdx][compIdx] có thể là chênh lệch vector chuyển động đối với vị trí (x, y), chỉ số vector chuyển động điểm điều khiển cpIdx và chỉ số thành phần compIdx. Ở đây, thành phần có thể biểu thị thành phần x hoặc thành phần y.

Ngoài ra, theo một phương án của sáng chế, MvdLX (chênh lệch vector chuyển động) hoặc MvdCpLX (chênh lệch vector chuyển động điểm điều khiển), tùy theo giá trị nào không được sử dụng, có thể phụ thuộc vào việc phép bù chuyển động afin có được sử dụng hay không. Ví dụ, MvdLX có thể được khởi tạo khi phép bù chuyển động afin được sử dụng. Ngoài ra, MvdCpLX có thể được khởi tạo khi phép bù chuyển động afin không được sử dụng. Ví dụ, báo hiệu biểu thị xem phép bù chuyển động afin có được sử dụng hay không có thể tồn tại. Dựa vào Fig.40, inter_affine_flag (cờ afin liên ảnh) có thể là báo hiệu biểu thị xem phép bù chuyển động afin có được sử dụng hay không. Ví dụ, khi inter_affine_flag (cờ afin liên ảnh) là 1, thì phép bù chuyển động afin có thể được sử dụng. Theo cách khác, MotionModelIdc có thể là báo hiệu biểu thị xem phép bù chuyển động afin có được sử dụng hay không. Ví dụ, khi MotionModelIdc không phải là 0, thì phép bù chuyển động afin có thể được sử dụng.

Ngoài ra, theo một phương án của sáng chế, MvdLX hoặc MvdCpLX, tùy theo giá trị nào không được sử dụng, có thể liên quan đến mô hình chuyển động afin nào được sử dụng. Ví dụ, MvdLX hoặc MvdCpLX mà không được sử dụng có thể khác tùy thuộc vào việc mô hình afin 4 thông số hay mô hình afin 6 thông số được sử dụng. Ví dụ, tùy thuộc vào mô hình chuyển động afin nào được sử dụng, MvdCpLX mà không được sử dụng cho cpIdx của MvdCpLX[x][y][cpIdx][compIdx] có thể khác. Ví dụ, khi sử dụng mô hình afin

4 thông số, thì chỉ một phần của $MvdCpLX[x][y][cpIdx][compIdx]$ có thể được sử dụng. Theo cách khác, khi mô hình afin 6 thông số không được sử dụng, thì chỉ một phần của $MvdCpLX[x][y][cpIdx][compIdx]$ có thể được sử dụng. Do đó, $MvdCpLX$ mà không được sử dụng có thể được khởi tạo thành giá trị được thiết lập trước. Trong trường hợp này, $MvdCpLX$ mà không được sử dụng có thể tương ứng với $cpIdx$ được sử dụng trong mô hình afin 6 thông số và không được sử dụng trong mô hình afin 4 thông số. Ví dụ, khi mô hình afin 4 thông số được sử dụng, thì giá trị trong đó $cpIdx$ của $MvdCpLX[x][y][cpIdx][compIdx]$ là 2 có thể không được sử dụng, và giá trị này có thể được khởi tạo thành giá trị được thiết lập trước. Ngoài ra, như được mô tả ở trên, báo hiệu hoặc thông số biểu thị xem mô hình afin 4 thông số hay mô hình afin 6 thông số được sử dụng có thể tồn tại. Ví dụ, có thể biết liệu mô hình afin 4 thông số được sử dụng hay mô hình afin 6 thông số được sử dụng bởi $MotionModelIdc$ hoặc $cu_affine_type_flag$. Các giá trị $MotionModelIdc$ là 1 và 2 có thể lần lượt biểu thị rằng mô hình afin 4 thông số và mô hình afin 6 thông số được sử dụng. Dựa vào Fig.40, khi $MotionModelIdc$ là 1, thì có thể khởi tạo $MvdCpL0[x0][y0][2][0]$, $MvdCpL0[x0][y0][2][1]$, $MvdCpL1[x0][y0][2][0]$ và $MvdCpL1[x0][y0][2][1]$ thành các giá trị được thiết lập trước. Ngoài ra, trong trường hợp này, có thể sử dụng điều kiện khi $MotionModelIdc$ không phải là 2 thay vì khi $MotionModelIdc$ là 1. Giá trị được thiết lập trước có thể là 0.

Ngoài ra, theo một phương án của sáng chế, $MvdLX$ hoặc $MvdCpLX$, tùy theo giá trị nào không được sử dụng, có thể dựa trên giá trị của $mvd_11_zero_flag$ (cờ chênh lệch vector chuyển động bằng không đối với danh mục ảnh tham chiếu thứ nhất). Ví dụ, khi $mvd_11_zero_flag$ là 1, thì có thể khởi tạo $MvdL1$ và $MvdCpL1$ thành các giá trị được thiết lập trước. Ngoài ra, trong trường hợp này, các điều kiện khác có thể được xem xét. Ví dụ, $MvdLX$ hoặc $MvdCpLX$, tùy theo giá trị nào không được sử dụng, có thể được xác định trên cơ sở $mvd_11_zero_flag$ và $inter_pred_idc$ (thông tin về danh mục ảnh tham chiếu). Ví dụ, khi $mvd_11_zero_flag$ là 1 và $inter_pred_idc$ là $PRED_BI$, thì có thể khởi tạo $MvdL1$ và $MvdCpL1$ thành các giá trị được thiết lập trước. Ví dụ, $mvd_11_zero_flag$ (cờ chênh lệch vector chuyển động bằng không đối với danh mục ảnh tham chiếu thứ nhất) có thể là báo hiệu mức cao hơn mà có thể biểu thị rằng các giá trị MVD (ví dụ, $MvdLX$ hoặc $MvdCpLX$) đối với danh mục tham chiếu L1 là 0. Báo hiệu có thể có nghĩa là tín hiệu được truyền từ bộ mã hóa đến bộ giải mã thông qua dòng bit. Bộ giải mã có thể phân tích $mvd_11_zero_flag$ (cờ chênh lệch vector chuyển động bằng không đối với danh mục ảnh tham chiếu thứ nhất) từ dòng bit.

Theo một phương án khác của sáng chế, có thể khởi tạo tất cả Mvd và MvdCp thành các giá trị được thiết lập trước trước khi thực hiện mvd_coding đối với khối nhất định. Trong trường hợp này, mvd_coding có thể có nghĩa là tất cả mvd_codings đối với CU nhất định. Do đó, việc phân tích mvd_coding cú pháp và khởi tạo giá trị Mvd hoặc MvdCp được xác định có thể được ngăn không cho được tạo ra, và vấn đề được mô tả ở trên có thể được giải quyết bằng cách khởi tạo tất cả các giá trị Mvd và MvdCp.

Fig.41 là sơ đồ minh họa cấu trúc cú pháp liên quan đến AMVR theo một phương án của sáng chế.

Phương án của Fig.41 có thể dựa trên phương án của Fig.38.

Như được mô tả với tham chiếu đến Fig.38, có thể kiểm tra được xem ít nhất một giá trị khác không có tồn tại trong số MvdLX (chênh lệch vector chuyển động) và MvdCpLX (chênh lệch vector chuyển động điểm điều khiển) hay không. Trong trường hợp này, MvdLX hoặc MvdCpLX cần được kiểm tra có thể được xác định trên cơ sở mvd_11_zero_flag (cờ chênh lệch vector chuyển động bằng không đối với danh mục ảnh tham chiếu thứ nhất). Do đó, có thể xác định xem có phân tích cú pháp liên quan đến AMVR hay không trên cơ sở mvd_11_zero_flag. Như được mô tả ở trên, khi mvd_11_zero_flag là 1, thì có thể biểu thị rằng các MVD (chênh lệch vector chuyển động) đối với danh mục tham chiếu L1 là 0 và do đó, trong trường hợp này, MvdL1 (chênh lệch vector chuyển động đối với danh mục ảnh tham chiếu thứ nhất) hoặc MvdCpL1 (chênh lệch vector chuyển động điểm điều khiển đối với danh mục ảnh tham chiếu thứ nhất) có thể không được xem xét. Ví dụ, khi mvd_11_zero_flag là 1, bất kể việc MvdL1 hoặc MvdCpL1 có bằng 0 hay không, cú pháp liên quan đến AMVR có thể được phân tích dựa trên việc ít nhất một giá trị bằng 0 có tồn tại trong số MvdL0 (chênh lệch vector chuyển động đối với danh mục ảnh tham chiếu thứ không) hoặc MvdCpL0 (chênh lệch vector chuyển động điểm điều khiển đối với danh mục ảnh tham chiếu thứ không) hay không. Theo một phương án khác, mvd_11_zero_flag biểu thị rằng các MVD đối với danh mục tham chiếu L1 là 0 có thể chỉ là cho khối mà là phép dự đoán theo hai hướng. Do đó, cú pháp liên quan đến AMVR có thể được phân tích trên cơ sở mvd_11_zero_flag và inter_pred_idc (thông tin về danh mục ảnh tham chiếu). Ví dụ, MvdLX hoặc MvdCpLX để đánh giá xem ít nhất một giá trị khác không có tồn tại hay không trên cơ sở mvd_11_zero_flag và inter_pred_idc (thông tin về danh mục ảnh tham chiếu) có thể được xác định. Ví dụ, MvdL1 hoặc MvdCpL1 có thể không được xem xét trên cơ sở mvd_11_zero_flag và inter_pred_idc. Ví dụ, khi mvd_11_zero_flag là 1 và

inter_pred_idc là PRED_BI, thì MvdL1 hoặc MvdCpL1 có thể không được xem xét. Ví dụ, khi mvd_11_zero_flag là 1 và inter_pred_idc là PRED_BI, thì cú pháp liên quan đến AMVR có thể được phân tích trên cơ sở việc ít nhất một giá trị 0 có tồn tại trong số MvdL0 hoặc MvdCpL0 hay không bất kể việc MvdL1 hoặc MvdCpL1 có bằng 0 hay không.

Dựa vào Fig.41, khi mvd_11_zero_flag là 1 và inter_pred_idc[x0][y0] == PRED_BI, thì hoạt động bất kể việc giá trị khác không có tồn tại trong số MvdL1[x0][y0][0], MvdL1[x0][y0][1], MvdCpL1[x0][y0][0][0], MvdCpL1[x0][y0][0][1], MvdCpL1[x0][y0][1][0], MvdCpL1[x0][y0][1][1], MvdCpL1[x0][y0][2][0] và MvdCpL1[x0][y0][2][1] hay không có thể được thực hiện. Ví dụ, khi mvd_11_zero_flag là 1 và inter_pred_idc[x0][y0] == PRED_BI, nếu giá trị khác không không tồn tại trong số MvdL0 hoặc MvdCpL0, thì cú pháp liên quan đến AMVR có thể không được phân tích thậm chí nếu giá trị khác không tồn tại trong số MvdL1[x0][y0][0], MvdL1[x0][y0][1], MvdCpL1[x0][y0][0][0], MvdCpL1[x0][y0][0][1], MvdCpL1[x0][y0][1][0], MvdCpL1[x0][y0][1][1], MvdCpL1[x0][y0][2][0] và MvdCpL1[x0][y0][2][1].

Ngoài ra, khi mvd_11_zero_flag là 0 hoặc inter_pred_idc[x0][y0] != PRED_BI, thì việc giá trị khác không có tồn tại trong số MvdL1[x0][y0][0], MvdL1[x0][y0][1], MvdCpL1[x0][y0][0][0], MvdCpL1[x0][y0][0][1], MvdCpL1[x0][y0][1][0], MvdCpL1[x0][y0][1][1], MvdCpL1[x0][y0][2][0] và MvdCpL1[x0][y0][2][1] hay không có thể được xem xét. Ví dụ, khi mvd_11_zero_flag là 0 hoặc inter_pred_idc[x0][y0] != PRED_BI, nếu giá trị khác không tồn tại trong số MvdL1[x0][y0][0], MvdL1[x0][y0][1], MvdCpL1[x0][y0][0][0], MvdCpL1[x0][y0][0][1], MvdCpL1[x0][y0][1][0], MvdCpL1[x0][y0][1][1], MvdCpL1[x0][y0][2][0] và MvdCpL1[x0][y0][2][1], thì cú pháp liên quan đến AMVR có thể được phân tích. Ví dụ, khi mvd_11_zero_flag là 0 hoặc inter_pred_idc[x0][y0] != PRED_BI, nếu giá trị khác không tồn tại trong số MvdL1[x0][y0][0], MvdL1[x0][y0][1], MvdCpL1[x0][y0][0][0], MvdCpL1[x0][y0][0][1], MvdCpL1[x0][y0][1][0], MvdCpL1[x0][y0][1][1], MvdCpL1[x0][y0][2][0] và MvdCpL1[x0][y0][2][1], thì cú pháp liên quan đến AMVR có thể được phân tích thậm chí nếu cả hai MvdL0 và MvdCpL0 là 0.

Fig.42 là sơ đồ minh họa cấu trúc cú pháp liên quan đến phép dự đoán liên ảnh theo một phương án của sáng chế.

Theo một phương án của sáng chế, mvd_11_zero_flag (cờ chênh lệch vector chuyển động bằng không đối với danh mục ảnh tham chiếu thứ nhất) có thể là báo hiệu biểu thị

rằng các giá trị Mvd (chênh lệch vector chuyển động) đối với danh mục tham chiếu L1 (danh mục ảnh tham chiếu thứ nhất) là 0. Ngoài ra, báo hiệu này có thể được báo hiệu ở mức cao hơn so với khối hiện thời. Do đó, các giá trị Mvd đối với danh mục tham chiếu L1 trong các khối có thể là 0 trên cơ sở giá trị `mvd_11_zero_flag`. Ví dụ, khi giá trị `mvd_11_zero_flag` là 1, thì các giá trị Mvd đối với danh mục tham chiếu L1 có thể là 0. Theo cách khác, các giá trị Mvd đối với danh mục tham chiếu L1 có thể là 0 trên cơ sở `mvd_11_zero_flag` và `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu). Ví dụ, khi `mvd_11_zero_flag` là 1 và `inter_pred_idc` là `PRED_BI`, thì các giá trị Mvd đối với danh mục tham chiếu L1 có thể là 0. Trong trường hợp này, các giá trị Mvd có thể là `MvdL1[x][y][compIdx]`. Ngoài ra, trong trường hợp này, các giá trị Mvd có thể không có nghĩa là các chênh lệch vector chuyển động điểm điều khiển. Nghĩa là, trong trường hợp này, các giá trị Mvd có thể không có nghĩa là các giá trị MvdCp.

Theo một phương án khác của sáng chế, `mvd_11_zero_flag` có thể là báo hiệu biểu thị rằng các giá trị Mvd và MvdCp đối với danh mục tham chiếu L1 là 0. Ngoài ra, báo hiệu này có thể được báo hiệu ở mức cao hơn so với khối hiện thời. Do đó, các giá trị Mvd và MvdCp đối với danh mục tham chiếu L1 trong các khối có thể là 0. Ví dụ, khi giá trị `mvd_11_zero_flag` là 1, thì các giá trị Mvd và MvdCp đối với danh mục tham chiếu L1 có thể là 0 trên cơ sở giá trị `mvd_11_zero_flag`. Theo cách khác, các giá trị Mvd và MvdCp đối với danh mục tham chiếu L1 có thể là 0 trên cơ sở giá trị `mvd_11_zero_flag`, `mvd_11_zero_flag` và `inter_pred_idc`. Ví dụ, khi `mvd_11_zero_flag` là 1 và `inter_pred_idc` là `PRED_BI`, thì các giá trị Mvd và MvdCp đối với danh mục tham chiếu L1 có thể là 0. Trong trường hợp này, các giá trị Mvd có thể là `MvdL1[x][y][compIdx]`. Ngoài ra, các giá trị MvdCp có thể là `MvdCpL1[x][y][cpIdx][compIdx]`.

Theo cách khác, các giá trị Mvd hoặc MvdCp là 0 có thể có nghĩa là cấu trúc cú pháp `mvd_coding` tương ứng không được phân tích. Nghĩa là, ví dụ, khi giá trị của `mvd_11_zero_flag` là 1, thì cấu trúc cú pháp `mvd_coding` tương ứng với MvdL1 hoặc MvdCpL1 có thể không được phân tích. Ngoài ra, khi giá trị `mvd_11_zero_flag` là 0, thì cấu trúc cú pháp `mvd_coding` tương ứng với MvdL1 hoặc MvdCpL1 có thể được phân tích.

Theo một phương án của sáng chế, khi các giá trị Mvd hoặc MvdCp là 0 trên cơ sở `mvd_11_zero_flag`, thì báo hiệu biểu thị MVP có thể không được phân tích. Báo hiệu biểu thị MVP có thể bao gồm `mvp_11_flag`. Ngoài ra, theo phần mô tả của `mvd_11_zero_flag` được mô tả ở trên, báo hiệu của `mvd_11_zero_flag` có thể có nghĩa là biểu thị rằng nó là 0

cho cả hai Mvd và MvdCp. Ví dụ, khi điều kiện biểu thị rằng các giá trị Mvd hoặc MvdCp đối với danh mục tham chiếu L1 là 0 được thỏa mãn, thì báo hiệu biểu thị MVP có thể không được phân tích. Trong trường hợp này, báo hiệu biểu thị MVP có thể được suy ra là giá trị được thiết lập trước. Ví dụ, khi báo hiệu biểu thị MVP không tồn tại, thì giá trị của nó có thể được suy ra là bằng 0. Ngoài ra, khi điều kiện biểu thị rằng các giá trị Mvd hoặc MvdCp là 0 trên cơ sở mvd_11_zero_flag không được thỏa mãn, thì báo hiệu biểu thị MVP có thể được phân tích. Tuy nhiên, theo phương án này, khi các giá trị Mvd hoặc MvdCp là 0, thì độ tự do khi chọn MVP có thể bị mất, và do đó hiệu quả mã hóa có thể bị giảm.

Cụ thể hơn là, khi điều kiện biểu thị rằng các giá trị Mvd hoặc MvdCp đối với danh mục tham chiếu L1 là 0 được thỏa mãn và MC affin được sử dụng, thì báo hiệu biểu thị MVP có thể không được phân tích. Trong trường hợp này, báo hiệu biểu thị MVP có thể được suy ra là giá trị được thiết lập trước.

Dựa vào Fig.42, khi mvd_11_zero_flag là 1 và giá trị inter_pred_idc là PRED_BI, thì mvp_11_flag có thể không được phân tích. Ngoài ra, trong trường hợp này, giá trị của mvp_11_flag có thể được suy ra là bằng 0. Theo cách khác, khi mvd_11_zero_flag là 0 hoặc giá trị inter_pred_idc không phải là PRED_BI, thì mvp_11_flag có thể được phân tích.

Theo phương án này, việc xác định xem có phân tích báo hiệu biểu thị MVP hay không trên cơ sở mvd_11_zero_flag có thể xảy ra khi điều kiện cụ thể được thỏa mãn. Ví dụ, điều kiện cụ thể có thể bao gồm điều kiện trong đó general_merge_flag là 0. Ví dụ, general_merge_flag có thể có cùng ý nghĩa với merge_flag được mô tả ở trên. Ngoài ra, điều kiện cụ thể có thể bao gồm điều kiện dựa trên CuPredMode. Cụ thể hơn là, điều kiện cụ thể có thể bao gồm điều kiện trong đó CuPredMode không phải là MODE_IBC. Theo cách khác, điều kiện cụ thể có thể bao gồm điều kiện trong đó CuPredMode là MODE_INTER. Khi CuPredMode là MODE_IBC, thì có thể sử dụng phép dự đoán với ảnh hiện thời làm tham chiếu. Ngoài ra, khi CuPredMode là MODE_IBC, thì vector khối hoặc vector chuyển động tương ứng với khối có thể tồn tại. Khi CuPredMode là MODE_INTER, thì có thể sử dụng phép dự đoán với ảnh khác với ảnh hiện thời làm tham chiếu. Khi CuPredMode là MODE_INTER, thì vector chuyển động tương ứng với khối có thể tồn tại.

Do đó, theo phương án của sáng chế, khi general_merge_flag là 0, CuPredMode không phải là MODE_IBC, mvd_11_zero_flag là 1 và inter_pred_idc là PRED_BI, thì mvp_11_flag có thể không được phân tích. Ngoài ra, khi mvp_11_flag không tồn tại, thì giá

trị của nó có thể được suy ra là bằng 0.

Cụ thể hơn là, khi `general_merge_flag` là 0, `CuPredMode` không phải là `MODE_IBC`, `mvd_11_zero_flag` là 1, `inter_pred_idc` là `PRED_BI` và MC afin được sử dụng, thì `mvp_11_flag` có thể không được phân tích. Ngoài ra, khi `mvp_11_flag` không tồn tại, thì giá trị của nó có thể được suy ra là bằng 0.

Dựa vào Fig.42, `sym_mvd_flag` có thể là báo hiệu biểu thị MVD đối xứng. Trong trường hợp MVD đối xứng, MVD khác có thể được xác định trên cơ sở MVD nhất định. Trong trường hợp MVD đối xứng, MVD khác có thể được xác định trên cơ sở MVD được báo hiệu rõ ràng. Ví dụ, trong trường hợp MVD đối xứng, MVD đối với danh mục tham chiếu khác có thể được xác định trên cơ sở MVD đối với một danh mục tham chiếu. Ví dụ, trong trường hợp MVD đối xứng, MVD đối với danh mục tham chiếu L1 có thể được xác định trên cơ sở MVD đối với danh mục tham chiếu L0. Khi xác định MVD khác dựa trên MVD nhất định, có thể xác định giá trị được thu bằng cách đảo dấu của MVD nhất định với MVD khác.

Fig.43 là sơ đồ cấu trúc cú pháp liên quan đến phép dự đoán liên ảnh theo một phương án của sáng chế.

Phương án của Fig.43 có thể là một phương án để giải quyết vấn đề được mô tả với tham chiếu đến Fig.42.

Theo một phương án của sáng chế, khi các giá trị `Mvd` (chênh lệch vector chuyển động) hoặc `MvdCp` (chênh lệch vector chuyển động điểm điều khiển) là 0 trên cơ sở `mvd_11_zero_flag` (cờ chênh lệch vector chuyển động bằng không đối với danh mục ảnh tham chiếu thứ nhất), thì có thể phân tích báo hiệu biểu thị MVP (bộ dự đoán vector chuyển động). Báo hiệu biểu thị MVP có thể bao gồm `mvp_11_flag` (chỉ số dự đoán vector chuyển động cho danh mục ảnh tham chiếu thứ nhất). Ngoài ra, theo phần mô tả của `mvd_11_zero_flag` được mô tả ở trên, báo hiệu `mvd_11_zero_flag` có thể có nghĩa là biểu thị rằng nó là 0 cho cả hai `Mvd` và `MvdCp`. Ví dụ, khi điều kiện biểu thị rằng các giá trị `Mvd` hoặc `MvdCp` đối với danh mục tham chiếu L1 (danh mục ảnh tham chiếu thứ nhất) là 0 được thỏa mãn, thì báo hiệu biểu thị MVP có thể được phân tích. Do đó, báo hiệu biểu thị MVP có thể không được suy ra. Do đó, thậm chí khi giá trị `Mvd` hoặc `MvdCp` là 0 trên cơ sở `mvd_11_zero_flag`, có thể có độ tự do khi chọn MVP. Do đó, hiệu quả mã hóa có thể được cải thiện. Ngoài ra, thậm chí khi điều kiện biểu thị rằng các giá trị `Mvd` hoặc `MvdCp` là 0 không được thỏa mãn trên cơ sở `mvd_11_zero_flag`, thì báo hiệu biểu thị MVP có thể

được phân tích.

Cụ thể hơn là, khi điều kiện biểu thị rằng các giá trị Mvd hoặc MvdCp đối với danh mục tham chiếu L1 là 0 được thỏa mãn và MC affin được sử dụng, thì báo hiệu biểu thị MVP có thể được phân tích.

Dựa vào dòng 4301 của Fig.43, thông tin `inter_pred_idc` về danh mục ảnh tham chiếu cho khôi hiện thời có thể được thu. Dựa vào dòng 4302, khi thông tin `inter_pred_idc` về danh mục ảnh tham chiếu biểu thị rằng chỉ danh mục ảnh tham chiếu thứ không list 0 không được sử dụng, thì chỉ số dự đoán vector chuyển động `mvp_11_flag` của danh mục ảnh tham chiếu thứ nhất list 1 có thể được phân tích từ dòng bit, ở dòng 4303.

`mvd_11_zero_flag` (cờ chênh lệch vector chuyển động bằng không) có thể được thu từ dòng bit. `mvd_11_zero_flag` (cờ chênh lệch vector chuyển động bằng không) có thể biểu thị xem MvdLX (chênh lệch vector chuyển động) và MvdCpLX (các chênh lệch vector chuyển động điểm điều khiển) có được thiết lập thành 0 cho danh mục ảnh tham chiếu thứ nhất hay không. Báo hiệu có thể có nghĩa là tín hiệu được truyền từ bộ mã hóa đến bộ giải mã thông qua dòng bit. Bộ giải mã có thể phân tích `mvd_11_zero_flag` (cờ chênh lệch vector chuyển động bằng không) từ dòng bit.

Khi `mvd_11_zero_flag` (cờ chênh lệch vector chuyển động bằng không) là 1 và `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) là `PRED_BI`, thì `mvp_11_flag` (chỉ số dự đoán vector chuyển động) có thể được phân tích. Ở đây, `PRED_BI` có thể biểu thị rằng cả hai danh mục 0 (danh mục ảnh tham chiếu thứ không) và danh mục 1 (danh mục ảnh tham chiếu thứ nhất) được sử dụng. Theo cách khác, khi `mvd_11_zero_flag` (cờ chênh lệch vector chuyển động bằng không) là 0 hoặc `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) không phải là `PRED_BI`, thì `mvp_11_flag` (chỉ số dự đoán vector chuyển động) có thể được phân tích. Nghĩa là, khi `mvd_11_zero_flag` (cờ chênh lệch vector chuyển động bằng không) là 1, bất kể việc `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) biểu thị rằng cả hai danh mục ảnh tham chiếu thứ không và danh mục ảnh tham chiếu thứ nhất có được sử dụng hay không, thì `mvp_11_flag` (chỉ số dự đoán vector chuyển động) có thể được phân tích.

Theo phương án này, việc xác định Mvd và MvdCp trên cơ sở `mvd_11_zero_flag` (cờ chênh lệch vector chuyển động bằng không của danh mục ảnh tham chiếu thứ nhất) và phân tích báo hiệu biểu thị MVP có thể xảy ra khi điều kiện cụ thể được thỏa mãn. Ví dụ, điều kiện cụ thể có thể bao gồm điều kiện trong đó `general_merge_flag` là 0. Ví dụ,

`general_merge_flag` có thể có cùng ý nghĩa với `merge_flag` được mô tả ở trên. Ngoài ra, điều kiện cụ thể có thể bao gồm điều kiện dựa trên `CuPredMode`. Cụ thể hơn là, điều kiện cụ thể có thể bao gồm điều kiện trong đó `CuPredMode` không phải là `MODE_IBC`. Theo cách khác, điều kiện cụ thể có thể bao gồm điều kiện trong đó `CuPredMode` là `MODE_INTER`. Khi `CuPredMode` là `MODE_IBC`, thì có thể sử dụng phép dự đoán với ảnh hiện thời làm tham chiếu. Ngoài ra, khi `CuPredMode` là `MODE_IBC`, thì vector khối hoặc vector chuyển động tương ứng với khối có thể tồn tại. Nếu `CuPredMode` là `MODE_INTER`, thì có thể sử dụng phép dự đoán với ảnh khác với ảnh hiện thời làm tham chiếu. Khi `CuPredMode` là `MODE_INTER`, thì vector chuyển động tương ứng với khối có thể tồn tại.

Do đó, theo phương án của sáng chế, khi `general_merge_flag` là 0, `CuPredMode` không phải là `MODE_IBC`, `mvd_11_zero_flag` là 1, và `inter_pred_idc` là `PRED_BI`, `mvp_11_flag` có thể được phân tích. Do đó, `mvp_11_flag` (chỉ số dự đoán vector chuyển động cho danh mục ảnh tham chiếu thứ nhất) tồn tại, và giá trị của nó có thể không được suy ra.

Cụ thể hơn là, khi `general_merge_flag` là 0, `CuPredMode` không phải là `MODE_IBC`, `mvd_11_zero_flag` (cờ chênh lệch vector chuyển động bằng không đối với danh mục ảnh tham chiếu thứ nhất) là 1, `inter_pred_idc` (thông tin về danh mục ảnh tham chiếu) là `PRED_BI` và MC afin được sử dụng, thì `mvp_11_flag` có thể được phân tích. Ngoài ra, `mvp_11_flag` (chỉ số dự đoán vector chuyển động cho danh mục ảnh tham chiếu thứ nhất) tồn tại, và giá trị của nó có thể không được suy ra.

Cũng có thể thực hiện phương án của Fig.43 và Fig.40 cùng nhau. Ví dụ, có thể phân tích `mvp_11_flag` sau khi khởi tạo `Mvd` hoặc `MvdCp`. Trong trường hợp này, việc khởi tạo `Mvd` hoặc `MvdCp` có thể là khởi tạo được mô tả với tham chiếu đến Fig.40. Ngoài ra, việc phân tích `mvp_11_flag` có thể theo phần mô tả của Fig.43. Ví dụ, khi các giá trị `Mvd` và `MvdCp` đối với danh mục tham chiếu L1 không phải là 0 dựa trên `mvd_11_zero_flag`, nếu giá trị `MotionModelIdc` là 1, thì các giá trị `MvdCpL1` của chỉ số điểm điều khiển 2 có thể được khởi tạo và `mvp_11_flag` có thể được phân tích.

Fig.44 là sơ đồ minh họa cấu trúc cú pháp liên quan đến phép dự đoán liên ảnh theo một phương án của sáng chế.

Phương án của Fig.44 có thể là một phương án để tăng hiệu quả mã hóa bằng cách không loại bỏ độ tự do khi chọn MVP. Ngoài ra, phương án của Fig.44 có thể là một phương

án được thu bằng cách mô tả phương án được mô tả với tham chiếu đến Fig.43 theo cách khác. Do đó, phần mô tả dư thừa với phương án của Fig.43 có thể được lược bỏ.

Theo phương án của Fig.44, khi biểu thị rằng giá trị Mvd hoặc MvdCp là 0 dựa trên `mvd_11_zero_flag`, thì báo hiệu biểu thị MVP có thể được phân tích. Báo hiệu biểu thị MVP có thể bao gồm `mvp_11_flag`.

Fig.45 là sơ đồ minh họa cú pháp liên quan đến phép dự đoán liên ảnh theo một phương án của sáng chế.

Theo một phương án của sáng chế, phép dự đoán liên ảnh phương pháp có thể bao gồm chế độ bỏ qua, chế độ hợp nhất, chế độ liên ảnh, v.v.. Theo một phương án, tín hiệu dư có thể không được truyền ở chế độ bỏ qua. Ngoài ra, phương pháp xác định MV chẳng hạn như chế độ hợp nhất có thể được sử dụng ở chế độ bỏ qua. Việc có sử dụng chế độ bỏ qua hay không có thể được xác định theo cờ bỏ qua. Dựa vào Fig.33, việc có sử dụng chế độ bỏ qua hay không có thể được xác định theo giá trị của `cu_skip_flag`.

Theo một phương án, chênh lệch vector chuyển động có thể không được sử dụng ở chế độ hợp nhất. Vector chuyển động có thể được xác định trên cơ sở chỉ số ứng viên chuyển động. Việc có sử dụng chế độ hợp nhất hay không có thể được xác định theo cờ hợp nhất. Dựa vào Fig.33, việc có sử dụng chế độ hợp nhất hay không có thể được xác định theo giá trị `merge_flag`. Ngoài ra, có thể sử dụng chế độ hợp nhất khi chế độ bỏ qua không được sử dụng.

Ở chế độ bỏ qua hoặc chế độ hợp nhất, có thể sử dụng chọn lọc một hoặc nhiều kiểu danh mục ứng viên. Ví dụ, có thể sử dụng ứng viên hợp nhất hoặc ứng viên hợp nhất khối con. Ngoài ra, ứng viên hợp nhất có thể bao gồm ứng viên lân cận không gian, ứng viên thời gian, v.v.. Ngoài ra, ứng viên hợp nhất có thể bao gồm ứng viên sử dụng vector chuyển động cho toàn bộ khối hiện thời (CU; đơn vị mã hóa). Nghĩa là, vector chuyển động của mỗi khối con thuộc khối hiện thời có thể bao gồm cùng ứng viên. Ngoài ra, ứng viên hợp nhất khối con có thể bao gồm MV thời gian dựa trên khối con, ứng viên hợp nhất afin, v.v.. Ngoài ra, ứng viên hợp nhất khối con có thể bao gồm ứng viên mà vector chuyển động khác đối với mỗi khối con của khối hiện thời (CU) có thể được sử dụng. Ứng viên hợp nhất afin có thể là phương pháp được xây dựng bằng phương pháp xác định vector chuyển động điểm điều khiển của phép dự đoán chuyển động afin mà không sử dụng chênh lệch vector chuyển động khi xác định vector chuyển động điểm điều khiển. Ngoài ra, ứng viên hợp nhất khối con có thể bao gồm các phương pháp xác định vector chuyển động theo các đơn vị của các

khối con trong khối hiện thời. Ví dụ, ứng viên hợp nhất khối con có thể bao gồm MV mặt phẳng, MV dựa trên hồi quy, STMVP, v.v. ngoài MV thời gian dựa trên khối con và ứng viên hợp nhất afin được nêu ở trên.

Theo một phương án, chênh lệch vector chuyển động có thể được sử dụng ở chế độ liên ảnh. Bộ dự đoán vector chuyển động có thể được xác định trên cơ sở chỉ số ứng viên chuyển động, và vector chuyển động có thể được xác định trên cơ sở chênh lệch giữa bộ dự đoán vector chuyển động và chênh lệch vector chuyển động. Việc có sử dụng chế độ liên ảnh hay không có thể được xác định theo việc các chế độ khác có được sử dụng hay không. Theo một phương án khác, việc có sử dụng chế độ liên ảnh hay không có thể được xác định bởi cờ. Fig.45 minh họa ví dụ sử dụng chế độ liên ảnh khi chế độ bỏ qua và chế độ hợp nhất, mà là các chế độ khác, không được sử dụng.

Chế độ liên ảnh có thể bao gồm chế độ AMVP, chế độ liên ảnh afin, v.v.. Chế độ liên ảnh có thể là chế độ để xác định vector chuyển động trên cơ sở bộ dự đoán vector chuyển động và chênh lệch vector chuyển động. Chế độ liên ảnh afin có thể là phương pháp sử dụng chênh lệch vector chuyển động khi xác định vector chuyển động điểm điều khiển của phép dự đoán chuyển động afin.

Dựa vào Fig.45, có thể xác định xem có sử dụng ứng viên hợp nhất khối con hoặc ứng viên hợp nhất sau khi chế độ bỏ qua hoặc chế độ hợp nhất được xác định. Ví dụ, khi điều kiện cụ thể được thỏa mãn, thì `merge_subblock_flag` biểu thị xem ứng viên hợp nhất khối con có được sử dụng hay không có thể được phân tích. Ngoài ra, điều kiện cụ thể có thể là điều kiện liên quan đến kích thước khối. Ví dụ, nó có thể là điều kiện liên quan đến chiều rộng, chiều cao, diện tích, v.v., và tổ hợp các yếu tố này có thể được sử dụng. Dựa vào Fig.45, ví dụ, nó có thể là điều kiện khi chiều rộng và chiều cao của khối hiện thời (CU) lớn hơn hoặc bằng giá trị cụ thể. Khi phân tích `merge_subblock_flag`, thì giá trị của nó có thể được suy ra là bằng 0. Nếu `merge_subblock_flag` là 1, thì ứng viên hợp nhất khối con có thể được sử dụng, và nếu `merge_subblock_flag` là 0, thì ứng viên hợp nhất có thể được sử dụng. Khi ứng viên hợp nhất khối con được sử dụng, thì `merge_subblock_idx`, mà là chỉ số ứng viên, có thể được phân tích, và khi ứng viên hợp nhất được sử dụng, thì `merge_idx`, mà là chỉ số ứng viên, có thể được phân tích. Trong trường hợp này, khi số lượng danh sách ứng viên tối đa là 1, thì việc phân tích có thể không được thực hiện. Khi `merge_subblock_idx` hoặc `merge_idx` không được phân tích, thì giá trị này có thể được suy ra là bằng 0.

Fig.45 minh họa hàm `coding_unit`, trong đó nội dung liên quan đến phép dự đoán nội ảnh có thể được lược bỏ, và Fig.45 có thể minh họa trường hợp phép dự đoán liên ảnh được xác định.

Fig.46 là sơ đồ minh họa chế độ phân chia hình tam giác theo một phương án của sáng chế.

Chế độ phân chia hình tam giác (triangle partitioning mode, TPM) được nêu trong sáng chế có thể được gọi bằng các tên khác nhau chẳng hạn như chế độ phân chia hình tam giác, phép dự đoán hình tam giác, phép dự đoán dựa trên hình tam giác, phép bù chuyển động hình tam giác, phép dự đoán tam giác, phép dự đoán liên ảnh hình tam giác, chế độ hợp nhất tam giác và chế độ hợp nhất hình tam giác. Ngoài ra, TPM có thể được bao gồm trong chế độ phân chia hình học (geometric partitioning mode, GPM).

Như được minh họa trên Fig.46, TPM có thể là phương pháp phân chia khối hình chữ nhật thành hai hình tam giác. Tuy nhiên, GPM có thể phân chia một khối thành hai khối theo các cách khác nhau. Ví dụ, GPM có thể phân chia một khối hình chữ nhật thành hai khối hình tam giác như được minh họa trên Fig.46. Ngoài ra, GPM có thể phân chia một khối hình chữ nhật thành một khối hình ngũ giác và một khối hình tam giác. Ngoài ra, GPM có thể phân chia một khối hình chữ nhật thành hai khối tứ giác. Ở đây, hình chữ nhật có thể bao gồm hình vuông. Sau đây, nhằm thuận tiện cho phần mô tả, phần mô tả được đưa ra dựa trên TPM, mà là phiên bản đơn giản của GPM, nhưng nên được hiểu là bao gồm GPM.

Theo một phương án của sáng chế, phép dự đoán theo một hướng có thể tồn tại dưới dạng phương pháp dự đoán. Phép dự đoán theo một hướng có thể là phương pháp dự đoán sử dụng một danh mục tham chiếu. Các danh mục tham chiếu có thể tồn tại, và theo một phương án, hai danh mục tham chiếu L0 và L1 có thể tồn tại. Khi sử dụng phép dự đoán theo một hướng, có thể sử dụng một danh mục tham chiếu trong một khối. Ngoài ra, khi sử dụng phép dự đoán theo một hướng, có thể sử dụng một thông tin chuyển động để dự đoán một điểm ảnh. Theo sáng chế, khối có thể có nghĩa là đơn vị mã hóa (coding unit, CU) hoặc đơn vị dự đoán (prediction unit, PU). Ngoài ra, theo sáng chế, khối có thể có nghĩa là đơn vị biến đổi (transformation unit, TU).

Theo một phương án khác của sáng chế, phép dự đoán theo hai hướng có thể tồn tại dưới dạng phương pháp dự đoán. Phép dự đoán theo hai hướng có thể là phương pháp dự đoán sử dụng các danh mục tham chiếu. Theo một phương án, phép dự đoán theo hai hướng có thể là phương pháp dự đoán sử dụng hai danh mục tham chiếu. Ví dụ, phép dự đoán theo

hai hướng có thể sử dụng các danh mục tham chiếu L0 và L1. Khi sử dụng phép dự đoán theo hai hướng, có thể sử dụng nhiều danh mục tham chiếu trong một khối. Ví dụ, khi sử dụng phép dự đoán theo hai hướng, có thể sử dụng hai danh mục tham chiếu trong một khối. Ngoài ra, khi sử dụng phép dự đoán theo hai hướng, có thể sử dụng nhiều thông tin chuyển động để dự đoán một điểm ảnh.

Thông tin chuyển động có thể bao gồm vector chuyển động, chỉ số tham chiếu và cờ sử dụng danh mục dự đoán.

Danh mục tham chiếu có thể là danh mục ảnh tham chiếu.

Theo sáng chế, thông tin chuyển động tương ứng với phép dự đoán theo một hướng hoặc phép dự đoán theo hai hướng có thể được xác định là một tập thông tin chuyển động.

Theo một phương án của sáng chế, có thể sử dụng các tập thông tin chuyển động khi sử dụng TPM. Ví dụ, khi sử dụng TPM, có thể sử dụng hai tập thông tin chuyển động. Ví dụ, khi sử dụng TPM, có thể sử dụng nhiều nhất hai tập thông tin chuyển động. Ngoài ra, phương pháp trong đó hai tập thông tin chuyển động được áp dụng trong khối sử dụng TPM có thể dựa trên vị trí. Ví dụ, trong khối mà TPM được sử dụng, có thể sử dụng một tập thông tin chuyển động cho vị trí được thiết lập trước và sử dụng tập thông tin chuyển động khác cho vị trí được thiết lập trước khác. Ngoài ra, đối với vị trí được thiết lập trước khác, có thể sử dụng hai tập thông tin chuyển động cùng nhau. Ví dụ, đối với vị trí được thiết lập trước khác, phép dự đoán 3 dựa trên phép dự đoán 1, mà là dựa trên một tập thông tin chuyển động, và phép dự đoán 2, mà là dựa trên tập thông tin chuyển động khác, có thể được sử dụng để dự đoán. Ví dụ, phép dự đoán 3 có thể là tổng được xác định trọng số của phép dự đoán 1 và phép dự đoán 2.

Dựa vào Fig.46, phần chia 1 và phần chia 2 có thể biểu diễn giản lược vị trí được thiết lập trước và vị trí được thiết lập trước khác. Khi sử dụng TPM, một trong số hai phương pháp phân chia có thể được sử dụng như được minh họa trên Fig.46. Hai phương pháp phân chia có thể bao gồm phân chia theo đường chéo và phân chia không theo đường chéo. Khối cũng có thể được phân chia thành hai phần chia có dạng hình tam giác bằng cách phân chia. Như đã được mô tả, TPM có thể được bao gồm trong GPM. Vì GPM đã được mô tả, nên phần mô tả dư thừa sẽ được lược bỏ.

Theo một phương án của sáng chế, khi sử dụng TPM, có thể sử dụng chỉ phép dự đoán theo một hướng cho mỗi phần chia. Nghĩa là, có thể sử dụng một thông tin chuyển

động cho mỗi phần chia. Điều này có thể là để làm giảm truy cập bộ nhớ và độ phức tạp chẳng hạn như độ phức tạp tính toán. Do đó, có thể sử dụng chỉ hai thông tin chuyển động cho mỗi CU.

Cũng có thể xác định mỗi thông tin chuyển động từ danh mục ứng viên. Theo một phương án, danh mục ứng viên được sử dụng cho TPM có thể dựa trên danh mục ứng viên hợp nhất. Theo một phương án khác, danh mục ứng viên được sử dụng cho TPM có thể dựa trên danh mục ứng viên AMVP. Do đó, có thể báo hiệu chỉ số ứng viên để sử dụng TPM. Ngoài ra, đối với khối sử dụng TPM, có thể mã hóa, giải mã và phân tích chỉ số ứng viên bằng số lượng phần chia hoặc nhiều nhất bằng số lượng phần chia trong TPM.

Ngoài ra, thậm chí nếu khối được dự đoán trên cơ sở nhiều thông tin chuyển động bởi TPM, có thể thực hiện việc biến đổi và lượng tử hóa đối với toàn bộ khối.

Fig.47 là sơ đồ minh họa cú pháp dữ liệu hợp nhất theo một phương án của sáng chế.

Theo một phương án của sáng chế, cú pháp dữ liệu hợp nhất có thể bao gồm báo hiệu liên quan đến các chế độ khác nhau. Các chế độ khác nhau có thể bao gồm chế độ hợp nhất thông thường, hợp nhất với MVD (merge with MVD, MMVD), chế độ hợp nhất khối con, phép dự đoán nội ảnh và liên ảnh kết hợp (combined intra-and inter-prediction, CIIP), TPM, v.v.. Chế độ hợp nhất thông thường có thể là chế độ giống với chế độ hợp nhất trong HEVC. Ngoài ra, báo hiệu biểu thị xem các chế độ khác nhau có được sử dụng trong khối hay không có thể tồn tại. Ngoài ra, các báo hiệu này có thể được phân tích dưới dạng các phần tử cú pháp hoặc có thể được báo hiệu ngầm. Dựa vào Fig.47, báo hiệu biểu thị xem chế độ hợp nhất thông thường, MMVD, chế độ hợp nhất khối con, CIIP và TPM có được sử dụng hay không có thể lần lượt là `regular_merge_flag`, `mmvd_merge_flag` (hoặc `mmvd_flag`), `merge_subblock_flag`, `ciip_flag` (hoặc `mh_intra_flag`), `MergeTriangleFlag` (hoặc `merge_triangle_flag`).

Theo một phương án của sáng chế, khi chế độ hợp nhất được sử dụng, nếu được báo hiệu rằng tất cả các chế độ ngoại trừ chế độ nhất định trong số các chế độ khác nhau không được sử dụng, có thể xác định được là sử dụng chế độ nhất định. Ngoài ra, khi chế độ hợp nhất được sử dụng, nếu được báo hiệu rằng ít nhất một trong số các chế độ ngoại trừ chế độ nhất định trong số các chế độ khác nhau được sử dụng, thì có thể xác định được rằng chế độ nhất định không được sử dụng. Ngoài ra, báo hiệu mức cao hơn biểu thị xem chế độ này có thể được sử dụng hay không có thể tồn tại. Mức cao hơn có thể là đơn vị bao gồm khối. Mức cao hơn có thể là chuỗi, ảnh, phần chia, nhóm tám, tám, CTU, v.v.. Nếu báo hiệu

mức cao hơn biểu thị xem chế độ này có thể được sử dụng hay không biểu thị rằng nó có thể được sử dụng, thì báo hiệu khác biểu thị xem chế độ này có được sử dụng hay không có thể tồn tại, và chế độ này có thể hoặc có thể không được sử dụng. Nếu báo hiệu mức cao hơn biểu thị xem chế độ này có thể được sử dụng hay không biểu thị rằng nó không thể được sử dụng, thì chế độ này có thể không được sử dụng. Ví dụ, khi sử dụng chế độ hợp nhất, nếu được báo hiệu rằng tất cả các chế độ trong số chế độ hợp nhất thông thường, MMVD, chế độ hợp nhất khối con và CIIP không được sử dụng, thì có thể xác định được là sử dụng TPM. Ngoài ra, khi sử dụng chế độ hợp nhất, nếu được báo hiệu rằng ít nhất một trong số chế độ hợp nhất thông thường, MMVD, chế độ hợp nhất khối con và CIIP được sử dụng, thì có thể xác định được là không sử dụng TPM. Ngoài ra, báo hiệu biểu thị xem chế độ hợp nhất có được sử dụng hay không có thể tồn tại. Ví dụ, báo hiệu biểu thị xem có sử dụng chế độ hợp nhất hay không có thể là `general_merge_flag` hoặc `merge_flag`. Nếu chế độ hợp nhất được sử dụng, thì cú pháp dữ liệu hợp nhất như được minh họa trên Fig.47 có thể được phân tích.

Ngoài ra, kích thước khối mà có thể được sử dụng cho TPM có thể được giới hạn. Ví dụ, có thể sử dụng TPM khi cả hai chiều rộng và chiều cao lớn hơn hoặc bằng 8.

Nếu TPM được sử dụng, thì các phần tử cú pháp liên quan đến TPM có thể được phân tích. Các phần tử cú pháp liên quan đến TPM có thể bao gồm báo hiệu biểu thị phương pháp phân chia và báo hiệu biểu thị chỉ số ứng viên. Phương pháp phân chia có thể có nghĩa là hướng phân chia. Nhiều báo hiệu (ví dụ, hai báo hiệu) biểu thị chỉ số ứng viên có thể tồn tại cho khối sử dụng TPM. Dựa vào Fig.47, báo hiệu biểu thị phương pháp phân chia có thể là `merge_triangle_split_dir`. Ngoài ra, báo hiệu biểu thị chỉ số ứng viên có thể là `merge_triangle_idx0` và `merge_triangle_idx1`.

Theo sáng chế, các chỉ số ứng viên cho TPM có thể là m và n. Ví dụ, các chỉ số ứng viên cho phần chia 1 và phần chia 2 của Fig.46 có thể lần lượt là m và n. Theo một phương án, m và n có thể được xác định dựa trên báo hiệu biểu thị chỉ số ứng viên được mô tả với tham chiếu đến Fig.47. Theo một phương án của sáng chế, một trong số m và n có thể được xác định trên cơ sở một trong số `merge_triangle_idx0` và `merge_triangle_idx1`, và một trong số m và n còn lại có thể được xác định trên cơ sở cả hai `merge_triangle_idx0` và `merge_triangle_idx1`.

Theo cách khác, một trong số m và n có thể được xác định trên cơ sở một trong số `merge_triangle_idx0` và `merge_triangle_idx1`, và một trong số m và n còn lại có thể được

xác định trên cơ sở một trong số `merge_triangle_idx0` và `merge_triangle_idx1` còn lại.

Cụ thể hơn là, có thể là `m` được xác định trên cơ sở `merge_triangle_idx0` và `n` được xác định trên cơ sở `merge_triangle_idx0` (hoặc `m`) và `merge_triangle_idx1`. Ví dụ, `m` và `n` có thể được xác định như sau.

$$m = \text{merge_triangle_idx0}$$

$$n = \text{merge_triangle_idx1} + (\text{merge_triangle_idx1} \geq m) ? 1 : 0$$

Theo một phương án của sáng chế, `m` và `n` có thể không giống nhau. Đó là do, trong TPM, khi hai chỉ số ứng viên giống nhau, nghĩa là, khi hai thông tin chuyển động giống nhau, thì hiệu quả phân chia có thể không được thu. Do đó, phương pháp báo hiệu ở trên có thể là để làm giảm số lượng bit báo hiệu trong trường hợp $n > m$ khi báo hiệu `n`. Vì `m` sẽ không phải là `n` trong số tất cả các ứng viên, nên nó có thể bị loại khỏi báo hiệu.

Nếu danh mục ứng viên được sử dụng trong TPM là `mergeCandList`, thì `mergeCandList[m]` và `mergeCandList[n]` có thể được sử dụng làm thông tin chuyển động trong TPM.

Fig.48 là sơ đồ minh họa báo hiệu mức cao hơn theo một phương án của sáng chế.

Theo một phương án của sáng chế, nhiều báo hiệu mức cao hơn có thể tồn tại. Báo hiệu mức cao hơn có thể là báo hiệu được truyền trong đơn vị mức cao hơn. Đơn vị mức cao hơn có thể bao gồm một hoặc nhiều đơn vị mức thấp hơn. Báo hiệu mức cao hơn có thể là báo hiệu được áp dụng vào một hoặc nhiều đơn vị mức thấp hơn. Ví dụ, phân chia hoặc chuỗi có thể là đơn vị mức cao hơn cho CU, PU, TU hoặc dạng tương tự. Ngược lại, CU, PU hoặc TU có thể là đơn vị mức thấp cho phân chia hoặc chuỗi.

Theo một phương án của sáng chế, báo hiệu mức cao hơn có thể bao gồm báo hiệu biểu thị số lượng ứng viên tối đa. Ví dụ, báo hiệu mức cao hơn có thể bao gồm báo hiệu biểu thị số lượng ứng viên hợp nhất tối đa. Ví dụ, báo hiệu mức cao hơn có thể bao gồm báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM. Báo hiệu biểu thị số lượng ứng viên hợp nhất tối đa hoặc báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM có thể được báo hiệu và được phân tích khi phép dự đoán liên ảnh được cho phép. Việc phép dự đoán liên ảnh có được cho phép hay không có thể được xác định bởi loại phân chia. Đối với các loại phân chia, I, P, B, v.v. có thể tồn tại. Ví dụ, khi loại phân chia là I, phép dự đoán liên ảnh có thể không được cho phép. Ví dụ, khi loại phân chia là I, chỉ phép dự đoán nội ảnh hoặc sao chép khối nội ảnh IBC có thể được sử dụng. Ngoài ra,

khi loại phần chia là P hoặc B, thì phép dự đoán liên ảnh có thể được cho phép. Ngoài ra, khi loại phần chia là P hoặc B, thì phép dự đoán nội ảnh, IBC, v.v. có thể được cho phép. Ngoài ra, khi loại phần chia là P, thì có thể sử dụng nhiều nhất một danh mục tham chiếu để dự đoán các điểm ảnh. Ngoài ra, khi loại phần chia là B, thì có thể sử dụng nhiều danh mục tham chiếu để dự đoán các điểm ảnh. Ví dụ, nếu loại phần chia là B, có thể sử dụng hai danh mục tham chiếu để dự đoán các điểm ảnh.

Theo một phương án của sáng chế, khi báo hiệu số lớn nhất, có thể báo hiệu trên cơ sở giá trị tham chiếu. Ví dụ, có thể báo hiệu (giá trị tham chiếu - số lớn nhất). Do đó, có thể suy ra số lớn nhất trên cơ sở giá trị được thu bằng cách phân tích bởi bộ giải mã và giá trị tham chiếu. Ví dụ, (giá trị tham chiếu - giá trị được thu bằng cách phân tích) có thể được xác định là số lớn nhất.

Theo một phương án, giá trị tham chiếu trong báo hiệu biểu thị số lượng ứng viên hợp nhất tối đa có thể là 6.

Theo một phương án, giá trị tham chiếu trong báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM có thể là số lượng ứng viên hợp nhất tối đa.

Dựa vào Fig.48, báo hiệu biểu thị số lượng ứng viên hợp nhất tối đa có thể là `six_minus_max_num_merge_cand`. Ở đây, ứng viên hợp nhất có thể có nghĩa là ứng viên dùng cho phép dự đoán vector chuyển động hợp nhất. Sau đây, nhằm thuận tiện cho phần mô tả, `six_minus_max_num_merge_cand` còn được gọi là thông tin thứ nhất. Dựa vào Fig.2 và Fig.7, báo hiệu có thể có nghĩa là tín hiệu được truyền từ bộ mã hóa đến bộ giải mã thông qua dòng bit. `six_minus_max_num_merge_cand` (thông tin thứ nhất) có thể được báo hiệu theo các đơn vị của các chuỗi. Bộ giải mã có thể phân tích `six_minus_max_num_merge_cand` (thông tin thứ nhất) từ dòng bit.

Ngoài ra, báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM có thể là `max_num_merge_cand_minus_max_num_triangle_cand`. Dựa vào Fig.2 và Fig.7, báo hiệu có thể có nghĩa là tín hiệu được truyền từ bộ mã hóa đến bộ giải mã thông qua dòng bit. Bộ giải mã có thể phân tích `max_num_merge_cand_minus_max_num_triangle_cand` (thông tin thứ ba) từ dòng bit. `max_num_merge_cand_minus_max_num_triangle_cand` (thông tin thứ ba) có thể là thông tin liên quan đến số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia.

Ngoài ra, số lượng ứng viên hợp nhất tối đa có thể là `MaxNumMergeCand` (số lượng

ứng viên hợp nhất tối đa), và giá trị này có thể dựa trên `six_minus_max_num_merge_cand` (thông tin thứ nhất). Ngoài ra, số lượng ứng viên tối đa được sử dụng trong TPM có thể là `MaxNumTriangleMergeCand`, và giá trị này có thể dựa trên `max_num_merge_cand_minus_max_num_triangle_cand`. `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) có thể được sử dụng cho chế độ hợp nhất, và là thông tin mà có thể được sử dụng khi khối được phân chia hoặc không được phân chia cho phép bù chuyển động. Ở trên, phần mô tả được đưa ra dựa trên TPM, nhưng GPM cũng có thể được mô tả theo cùng cách.

Theo một phương án của sáng chế, báo hiệu mức cao hơn biểu thị xem chế độ TPM có thể được sử dụng hay không có thể tồn tại. Dựa vào Fig.48, báo hiệu mức cao hơn biểu thị xem chế độ TPM có thể được sử dụng hay không có thể là `sps_triangle_enabled_flag` (thông tin thứ hai). Thông tin biểu thị xem chế độ TPM có thể được sử dụng hay không có thể giống với thông tin biểu thị xem khối có thể được phân chia hay không như được minh họa trên Fig.46 đối với phép dự đoán liên ảnh. Vì GPM bao gồm TPM, thông tin biểu thị xem khối có thể được phân chia hay không có thể giống với thông tin biểu thị xem có sử dụng chế độ GPM hay không. Việc thực hiện phép dự đoán liên ảnh có thể biểu thị việc thực hiện phép bù chuyển động. Nghĩa là, `sps_triangle_enabled_flag` (thông tin thứ hai) có thể là thông tin biểu thị xem khối có thể được phân chia hay không đối với phép dự đoán liên ảnh. Khi thông tin thứ hai biểu thị xem khối có thể được phân chia hay không là 1, thì có thể biểu thị rằng TPM hoặc GPM có thể được sử dụng. Ngoài ra, khi thông tin thứ hai là 0, thì có thể biểu thị rằng TPM hoặc GPM không thể được sử dụng. Tuy nhiên, sáng chế không bị giới hạn ở đó, và khi thông tin thứ hai là 0, thì có thể biểu thị rằng TPM hoặc GPM có thể được sử dụng. Ngoài ra, khi thông tin thứ hai là 1, thì có thể biểu thị rằng TPM hoặc GPM không thể được sử dụng.

Dựa vào Fig.2 và Fig.7, báo hiệu có thể có nghĩa là tín hiệu được truyền từ bộ mã hóa đến bộ giải mã thông qua dòng bit. Bộ giải mã có thể phân tích `sps_triangle_enabled_flag` (thông tin thứ hai) từ dòng bit.

Theo một phương án của sáng chế, có thể sử dụng TPM chỉ khi các ứng viên được sử dụng trong TPM lớn hơn hoặc bằng số lượng phân chia của TPM có thể tồn tại. Ví dụ, khi TPM được phân chia thành hai phần chia, có thể sử dụng TPM chỉ khi hai hoặc nhiều hơn hai ứng viên được sử dụng trong TPM có thể tồn tại. Theo một phương án, ứng viên được sử dụng trong TPM có thể dựa trên ứng viên hợp nhất. Do đó, theo một phương án

của sáng chế, có thể sử dụng TPM khi số lượng ứng viên hợp nhất tối đa lớn hơn hoặc bằng 2. Do đó, khi số lượng ứng viên hợp nhất tối đa lớn hơn hoặc bằng 2, có thể phân tích báo hiệu liên quan đến TPM. Báo hiệu liên quan đến TPM có thể là báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM.

Dựa vào Fig.48, khi `sps_triangle_enabled_flag` là 1 và `MaxNumMergeCand` lớn hơn hoặc bằng 2, có thể phân tích `max_num_merge_cand_minus_max_num_triangle_cand`. Ngoài ra, khi `sps_triangle_enabled_flag` là 0 hoặc `MaxNumMergeCand` nhỏ hơn 2, thì không thể phân tích `max_num_merge_cand_minus_max_num_triangle_cand`.

Fig.49 là sơ đồ minh họa số lượng ứng viên tối đa được sử dụng trong TPM theo một phương án của sáng chế.

Dựa vào Fig.49, số lượng ứng viên tối đa được sử dụng trong TPM có thể là `MaxNumTriangleMergeCand`. Ngoài ra, báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM có thể là `max_num_merge_cand_minus_max_num_triangle_cand`. Ngoài ra, các nội dung được mô tả với tham chiếu đến Fig.48 có thể được lược bỏ.

Theo một phương án của sáng chế, số lượng ứng viên tối đa được sử dụng trong TPM có thể tồn tại trong khoảng từ “số lượng phần chia của TPM” đến “giá trị tham chiếu trong báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM”. Do đó, khi số lượng phần chia của TPM là 2 và giá trị tham chiếu là số lượng ứng viên hợp nhất tối đa, thì `MaxNumTriangleMergeCand` có thể tồn tại trong khoảng từ 2 đến `MaxNumMergeCand`, như được minh họa trên Fig.49.

Theo một phương án của sáng chế, khi báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM không tồn tại, thì có thể suy ra báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM hoặc suy ra số lượng ứng viên tối đa được sử dụng trong TPM. Ví dụ, khi báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM không tồn tại, thì số lượng ứng viên tối đa được sử dụng trong TPM có thể được suy ra là bằng 0. Theo cách khác, khi báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM không tồn tại, thì báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM có thể được suy ra là giá trị tham chiếu.

Ngoài ra, khi báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM không tồn tại, thì không thể sử dụng TPM. Theo cách khác, khi số lượng ứng viên tối đa được sử dụng trong TPM nhỏ hơn số lượng phần chia trong TPM, thì không thể sử dụng

TPM. Theo cách khác, khi số lượng ứng viên tối đa được sử dụng trong TPM là 0, thì không thể sử dụng TPM.

Tuy nhiên, theo phương án của các hình vẽ từ Fig.48 đến Fig.49, khi số lượng phần chia của TPM và “giá trị tham chiếu trong báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM” là giống nhau, có thể có chỉ một giá trị khả thi là số lượng ứng viên tối đa được sử dụng trong TPM. Tuy nhiên, theo phương án của các hình vẽ từ Fig.48 đến Fig.49, thậm chí trong trường hợp này, báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM có thể được phân tích, mà có thể là không cần thiết. Khi MaxNumMergeCand là 2, tham chiếu đến Fig.49, giá trị khả thi là MaxNumTriangleMergeCand có thể chỉ là 2. Tuy nhiên, tham chiếu đến Fig.48, thậm chí trong trường hợp này, `max_num_merge_cand_minus_max_num_triangle_cand` có thể được phân tích.

Dựa vào Fig.49, MaxNumTriangleMergeCand có thể được xác định là $(\text{MaxNumMergeCand} - \text{max_num_merge_cand_minus_max_num_triangle_cand})$.

Fig.50 là sơ đồ minh họa báo hiệu mức cao hơn liên quan đến TPM theo một phương án của sáng chế.

Theo một phương án của sáng chế, khi số lượng phần chia của TPM và “giá trị tham chiếu trong báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM” giống nhau, báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM có thể không được phân tích. Ngoài ra, theo phương án được mô tả ở trên, số lượng phần chia của TPM có thể là 2. Ngoài ra, “giá trị tham chiếu trong báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM” có thể là số lượng ứng viên hợp nhất tối đa. Do đó, khi số lượng ứng viên hợp nhất tối đa là 2, thì báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM có thể không được phân tích.

Theo cách khác, khi “giá trị tham chiếu trong báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM” nhỏ hơn hoặc bằng số lượng phần chia trong TPM, thì báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM có thể không được phân tích. Do đó, khi số lượng ứng viên hợp nhất tối đa nhỏ hơn hoặc bằng 2, thì báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM có thể không được phân tích.

Dựa vào dòng 5001 của Fig.50, khi MaxNumMergeCand (số lượng ứng viên hợp nhất tối đa) là 2 hoặc MaxNumMergeCand (số lượng ứng viên hợp nhất tối đa) nhỏ hơn

hoặc bằng 2, thì `max_num_merge_cand_minus_max_num_triangle_cand` (thông tin thứ ba) có thể không được phân tích. Ngoài ra, khi `sps_triangle_enabled_flag` (thông tin thứ hai) là 1 và `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) lớn hơn 2, thì `max_num_merge_cand_minus_max_num_triangle_cand` (thông tin thứ ba) có thể được phân tích. Ngoài ra, khi `sps_triangle_enabled_flag` (thông tin thứ hai) là 0, thì `max_num_merge_cand_minus_max_num_triangle_cand` (thông tin thứ ba) có thể không được phân tích. Do đó, khi `sps_triangle_enabled_flag` (thông tin thứ hai) là 0 hoặc `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) nhỏ hơn hoặc bằng 2, thì `max_num_merge_cand_minus_max_num_triangle_cand` (thông tin thứ ba) có thể không được phân tích.

Fig.51 là sơ đồ minh họa số lượng ứng viên tối đa được sử dụng trong TPM theo một phương án của sáng chế.

Phương án của Fig.51 có thể được thực hiện cùng với trường hợp của phương án của Fig.50. Ngoài ra, các phần mô tả về những gì được mô tả ở trên có thể được lược bỏ trên hình vẽ này.

Theo một phương án của sáng chế, khi “giá trị tham chiếu trong báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM” là số lượng phần chia trong TPM, thì số lượng ứng viên tối đa được sử dụng trong TPM có thể được suy ra và được thiết lập thành số lượng phần chia của TPM. Ngoài ra, việc suy ra và thiết lập có thể được thực hiện trong trường hợp trong đó báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM không tồn tại. Theo phương án của Fig.50, khi “giá trị tham chiếu trong báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM” là số lượng phần chia trong TPM, thì báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM có thể không được phân tích, và khi báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM không tồn tại, thì giá trị của số lượng ứng viên tối đa được sử dụng trong TPM có thể được suy ra là số lượng phần chia trong TPM. Cũng có thể thực hiện việc suy ra và thiết lập khi điều kiện khác được thỏa mãn. Điều kiện khác có thể là điều kiện trong đó báo hiệu mức cao hơn biểu thị xem chế độ TPM có thể được sử dụng hay không là 1.

Ngoài ra, theo phương án này, mặc dù việc suy ra và thiết lập số lượng ứng viên tối đa được sử dụng trong TPM đã được mô tả, cũng có thể suy ra và thiết lập báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM sao cho giá trị số lớn nhất của ứng viên được sử dụng trong TPM được mô tả được suy ra, thay vì suy ra và thiết lập số lượng

ứng viên tối đa được sử dụng trong TPM.

Dựa vào Fig.50, khi `sps_triangle_enabled_flag` (thông tin thứ hai) biểu thị 1 và `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) lớn hơn 2, thì `max_num_merge_cand_minus_max_num_triangle_cand` (thông tin thứ ba) có thể được nhận. Trong trường hợp này, tham chiếu đến dòng 5101 của Fig.51, `MaxNumTriangleMergeCand` (số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia) có thể được thu nhờ sử dụng `max_num_merge_cand_minus_max_num_triangle_cand` (thông tin thứ ba) được báo hiệu rõ ràng. Tóm lại, khi `sps_triangle_enabled_flag` (thông tin thứ hai) biểu thị 1 và `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) lớn hơn hoặc bằng 3, thì `MaxNumTriangleMergeCand` (số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia) có thể được thu bằng cách lấy `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) trừ thông tin thứ ba (`max_num_merge_cand_minus_max_num_triangle_cand`).

Dựa vào dòng 5102 của Fig.51, khi `sps_triangle_enabled_flag` (thông tin thứ hai) là 1 và `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) là 2, thì có thể thiết lập `MaxNumTriangleMergeCand` (số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia) thành 2. Cụ thể hơn là, như đã được mô tả, khi `sps_triangle_enabled_flag` (thông tin thứ hai) biểu thị 1 và `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) lớn hơn 2, thì `max_num_merge_cand_minus_max_num_triangle_cand` (thông tin thứ ba) có thể được nhận, và do đó, như ở dòng 5102, khi `sps_triangle_enabled_flag` (thông tin thứ hai) là 1 và `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) là 2, thì `max_num_merge_cand_minus_max_num_triangle_cand` (thông tin thứ ba) có thể không được nhận. Trong trường hợp này, `MaxNumTriangleMergeCand` (số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia) có thể được xác định mà không có `max_num_merge_cand_minus_max_num_triangle_cand` (thông tin thứ ba).

Ngoài ra, tham chiếu đến dòng 5103 của Fig.51, khi `sps_triangle_enabled_flag` (thông tin thứ hai) là 0 hoặc `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) không phải là 2, thì có thể suy ra và thiết lập `MaxNumTriangleMergeCand` (số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia) thành 0. Trong trường hợp này, như đã được mô tả với tham chiếu đến dòng 5101, khi `sps_triangle_enabled_flag` (thông tin thứ hai) biểu thị 1 và `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) lớn hơn hoặc bằng 3, thì `max_num_merge_cand_minus_max_num_triangle_cand` (thông tin thứ ba) có thể được báo

hiệu, và do đó trường hợp trong đó `MaxNumTriangleMergeCand` được suy ra và thiết lập thành 0 có thể là trường hợp trong đó thông tin thứ hai là 0 hoặc số lượng ứng viên hợp nhất tối đa là 1. Tóm lại, khi `sps_triangle_enabled_flag` (thông tin thứ hai) là 0 hoặc `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) là 1, thì `MaxNumTriangleMergeCand` (số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia) có thể được thiết lập thành 0.

`MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) và `MaxNumTriangleMergeCand` (số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia) có thể được sử dụng cho các mục đích khác nhau. Ví dụ, `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) có thể được sử dụng khi khối được phân chia hoặc không được phân chia cho phép bù chuyển động. Tuy nhiên, `MaxNumTriangleMergeCand` (số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia) là thông tin mà có thể được sử dụng khi khối được phân chia. Số lượng ứng viên cho khối được phân chia, mà ở chế độ hợp nhất, không thể vượt quá `MaxNumTriangleMergeCand` (số lượng ứng viên chế độ hợp nhất tối đa cho khối được phân chia).

Cũng có thể sử dụng phương án khác. Theo phương án của Fig.51, trường hợp trong đó `MaxNumMergeCand` không phải là 2 bao gồm trường hợp trong đó `MaxNumMergeCand` lớn hơn 2, và trong trường hợp đó, ý nghĩa của việc suy ra và thiết lập `MaxNumTriangleMergeCand` thành 0 có thể là không rõ ràng, nhưng trong trường hợp này, vì báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM tồn tại, việc suy ra không được thực hiện, và theo đó, không có vấn đề khi hoạt động. Tuy nhiên, theo phương án này, việc suy ra có thể được thực hiện bằng cách sử dụng ý nghĩa.

Nếu `sps_triangle_enabled_flag` (thông tin thứ hai) là 1 và `MaxNumMergeCand` (số lượng ứng viên hợp nhất tối đa) lớn hơn hoặc bằng 2, thì có thể suy ra và thiết lập `MaxNumTriangleMergeCand` thành 2 (hoặc `MaxNumMergeCand`). Theo cách khác (nghĩa là, khi `sps_triangle_enabled_flag` là 0 hoặc `MaxNumMergeCand` nhỏ hơn 2), có thể suy ra và thiết lập `MaxNumTriangleMergeCand` thành 0.

Theo cách khác, nếu `sps_triangle_enabled_flag` là 1 và `MaxNumMergeCand` là 2, có thể suy ra và thiết lập `MaxNumTriangleMergeCand` thành 2. Theo cách khác, nếu `sps_triangle_enabled_flag` là 0, có thể suy ra và thiết lập `MaxNumTriangleMergeCand` thành 0.

Do đó, theo phương án của Fig.50 theo các phương án ở trên, khi

MaxNumMergeCand là 0 hoặc 1 hoặc 2, thì báo hiệu biểu thị số lượng ứng viên tối đa được sử dụng trong TPM có thể không tồn tại, và khi MaxNumMergeCand là 0 hoặc 1, thì số lượng ứng viên tối đa được sử dụng trong TPM có thể được suy ra và thiết lập thành 0. Khi MaxNumMergeCand là 2, thì số lượng ứng viên tối đa được sử dụng trong TPM có thể được suy ra và thiết lập thành 2.

Fig.52 là sơ đồ minh họa các phần tử cú pháp liên quan đến TPM theo một phương án của sáng chế.

Như được mô tả ở trên, số lượng ứng viên tối đa được sử dụng trong TPM có thể tồn tại, và số lượng phần chia của TPM có thể được thiết lập trước. Ngoài ra, các chỉ số ứng viên được sử dụng trong TPM có thể khác nhau.

Theo một phương án của sáng chế, khi số lượng ứng viên tối đa được sử dụng trong TPM giống với số lượng phần chia trong TPM, thì báo hiệu khác với trường hợp trong đó chúng không giống nhau có thể được thực hiện. Ví dụ, khi số lượng ứng viên tối đa được sử dụng trong TPM giống với số lượng phần chia trong TPM, thì báo hiệu khác với trường hợp trong đó chúng không giống nhau có thể được thực hiện. Theo đó, có thể báo hiệu với ít bit hơn. Theo cách khác, khi số lượng ứng viên tối đa được sử dụng trong TPM nhỏ hơn hoặc bằng số lượng phần chia của TPM, báo hiệu khác với trường hợp trong đó số lượng ứng viên tối đa lớn hơn số lượng phần chia có thể được thực hiện (trong số này, khi số lượng phần chia nhỏ hơn số lượng phần chia của TPM, nó có thể là trường hợp mà TPM không thể được sử dụng).

Khi có hai phần chia của TPM, thì hai chỉ số ứng viên có thể được báo hiệu. Nếu số lượng ứng viên tối đa được sử dụng trong TPM là 2, thì có thể chỉ có hai tổ hợp chỉ số ứng viên khả thi. Hai tổ hợp có thể là tổ hợp trong đó m và n lần lượt là 0 và 1 và tổ hợp trong đó m và n lần lượt là 1 và 0. Do đó, hai chỉ số ứng viên có thể được báo hiệu chỉ bởi 1 bit báo hiệu.

Dựa vào Fig.52, khi MaxNumTriangleMergeCand là 2, thì báo hiệu chỉ số ứng viên khác với trường hợp trong đó nó không phải là như vậy (theo cách khác, khi TPM được sử dụng, trường hợp trong đó MaxNumTriangleMergeCand lớn hơn 2) có thể được thực hiện. Theo cách khác, khi MaxNumTriangleMergeCand nhỏ hơn hoặc bằng 2, báo hiệu chỉ số ứng viên khác với trường hợp trong đó nó không phải là như vậy (theo cách khác, khi TPM được sử dụng, trường hợp trong đó MaxNumTriangleMergeCand lớn hơn 2) có thể được thực hiện. Báo hiệu chỉ số ứng viên khác có thể là phân tích merge_triangle_idx_indicator

khi tham chiếu đến Fig.52. Báo hiệu chỉ số ứng viên khác có thể là phương pháp báo hiệu mà không phân tích `merge_triangle_idx0` hoặc `merge_triangle_idx1`. Điều này sẽ còn được mô tả với tham chiếu đến Fig.53.

Fig.53 là sơ đồ minh họa báo hiệu chỉ số ứng viên TPM theo một phương án của sáng chế.

Theo một phương án của sáng chế, khi báo hiệu chỉ số khác với chỉ số được mô tả với tham chiếu đến Fig.52 được sử dụng, thì chỉ số ứng viên có thể được xác định dựa trên `merge_triangle_idx_indicator`. Ngoài ra, khi báo hiệu chỉ số khác được sử dụng thì `merge_triangle_idx0` hoặc `merge_triangle_idx1` có thể không tồn tại.

Theo một phương án của sáng chế, khi số lượng ứng viên tối đa được sử dụng trong TPM giống với số lượng phân chia trong TPM, thì chỉ số ứng viên TPM có thể được xác định trên cơ sở `merge_triangle_idx_indicator`. Ngoài ra, trường hợp này có thể là trường hợp mà khối sử dụng TPM.

Cụ thể hơn là, khi `MaxNumTriangleMergeCand` là 2 (hoặc khi `MaxNumTriangleMergeCand` là 2 và `MergeTriangleFlag` là 1), thì chỉ số ứng viên TPM có thể có thể được xác định trên cơ sở `merge_triangle_idx_indicator`. Trong trường hợp này, nếu `merge_triangle_idx_indicator` là 0, thì `m` và `n`, mà là các chỉ số ứng viên TPM, có thể lần lượt được thiết lập thành 0 và 1, và nếu `merge_triangle_idx_indicator` là 1, thì `m` và `n`, mà là các chỉ số ứng viên TPM, có thể lần lượt được thiết lập thành 1 và 0. Theo cách khác, `merge_triangle_idx0` hoặc `merge_triangle_idx1`, mà là giá trị (phần tử cú pháp) mà có thể được phân tích sao cho `m` và `n` là giống với `m` và `n` được mô tả, có thể được suy ra và được thiết lập.

Dựa vào phương pháp thiết lập `m` và `n` trên cơ sở `merge_triangle_idx0` và `merge_triangle_idx1` được mô tả trên Fig.47 và tham chiếu đến Fig.53, khi `merge_triangle_idx0` không tồn tại, nếu `MaxNumTriangleMergeCand` là 2 và `merge_triangle_idx_indicator` là 1 (hoặc nếu `MaxNumTriangleMergeCand` là 2 và `MergeTriangleFlag` là 1), thì giá trị `merge_triangle_idx0` có thể được suy ra là bằng 1. Ngoài ra, theo cách khác, giá trị của `merge_triangle_idx0` có thể được suy ra là bằng 0. Ngoài ra, khi `merge_triangle_idx1` không tồn tại, thì `merge_triangle_idx1` có thể được suy ra là bằng 0. Do đó, nếu `MaxNumTriangleMergeCand` là 2, khi `merge_triangle_idx_indicator` là 0, `merge_triangle_idx0` và `merge_triangle_idx1` lần lượt là 0 và 0, và theo đó, `m` và `n` có thể lần lượt là 0 và 1. Ngoài ra, nếu `MaxNumTriangleMergeCand` là 2, khi

`merge_triangle_idx_indicator` là 1, `merge_triangle_idx0` và `merge_triangle_idx1` lần lượt là 1 và 0, và theo đó, `m` và `n` có thể lần lượt là 1 và 0.

Fig.54 là sơ đồ minh họa báo hiệu chỉ số ứng viên TPM theo một phương án của sáng chế.

Phương pháp xác định các chỉ số ứng viên TPM đã được mô tả với tham chiếu đến Fig.47, nhưng phương pháp xác định khác và phương pháp báo hiệu được mô tả theo phương án của Fig.54. Phần mô tả mà là dư thừa với những gì được mô tả ở trên có thể được lược bỏ. Ngoài ra, `m` và `n` có thể biểu diễn các chỉ số ứng viên như được mô tả với tham chiếu đến Fig.47.

Theo một phương án của sáng chế, giá trị nhỏ hơn trong số `m` và `n` có thể được báo hiệu cho phần tử cú pháp được thiết lập trước trong số `merge_triangle_idx0` và `merge_triangle_idx1`. Ngoài ra, giá trị dựa trên chênh lệch giữa `m` và `n` có thể được báo hiệu cho phần tử cú pháp còn lại trong số `merge_triangle_idx0` và `merge_triangle_idx1`. Ngoài ra, có thể báo hiệu giá trị biểu thị mối quan hệ độ lớn của `m` và `n`.

Ví dụ, `merge_triangle_idx0` có thể là giá trị nhỏ hơn trong số `m` và `n`. Ngoài ra, `merge_triangle_idx1` có thể là giá trị dựa trên $|m-n|$. `merge_triangle_idx1` có thể là $(|m-n| - 1)$. Đó là do `m` và `n` có thể không giống nhau. Ngoài ra, giá trị biểu diễn mối quan hệ độ lớn giữa `m` và `n` có thể là `merge_triangle_bigger` của Fig.54.

Nhờ sử dụng mối quan hệ này, `m` và `n` có thể được xác định trên cơ sở `merge_triangle_idx0`, `merge_triangle_idx1` và `merge_triangle_bigger`. Dựa vào Fig.54, hoạt động khác có thể được thực hiện trên cơ sở giá trị `merge_triangle_bigger`. Ví dụ, khi `merge_triangle_bigger` là 0, thì `n` có thể lớn hơn `m`. Trong trường hợp này, `m` có thể là `merge_triangle_idx0`. Ngoài ra, `n` có thể là $(\text{merge_triangle_idx1} + m + 1)$. Ngoài ra, khi `merge_triangle_bigger` là 1, thì `m` có thể lớn hơn `n`. Trong trường hợp này, `n` có thể là `merge_triangle_idx0`. Ngoài ra, `m` có thể là $(\text{merge_triangle_idx1} + n + 1)$.

Phương pháp trên Fig.54 có ưu điểm là tổng phí báo hiệu có thể được làm giảm khi giá trị nhỏ hơn trong số `m` và `n` không phải là 0 (hoặc lớn hơn), so với phương pháp trên Fig.47. Ví dụ, khi `m` và `n` lần lượt là 3 và 4, theo phương pháp của Fig.47, `merge_triangle_idx0` và `merge_triangle_idx1` có thể cần được báo hiệu lần lượt là 3 và 3. Tuy nhiên, theo phương pháp của Fig.54, khi `m` và `n` lần lượt là 3 và 4, `merge_triangle_idx0` và `merge_triangle_idx1` có thể cần được báo hiệu lần lượt là 3 và 0 (ngoài ra, báo hiệu biểu

thì mối quan hệ độ lớn có thể cần thiết). Do đó, khi báo hiệu chiều dài khả biến được sử dụng, thì có thể sử dụng ít bit hơn vì kích thước của các giá trị mã hóa và giải mã được giảm bớt.

Fig.55 là sơ đồ minh họa báo hiệu chỉ số ứng viên TPM theo một phương án của sáng chế.

Phương pháp xác định các chỉ số ứng viên TPM đã được mô tả với tham chiếu đến Fig.47, nhưng phương pháp xác định khác và phương pháp báo hiệu sẽ được mô tả theo phương án của Fig.55. Phần mô tả mà là dư thừa với những gì được mô tả ở trên có thể được lược bỏ. Ngoài ra, m và n có thể biểu diễn các chỉ số ứng viên như được mô tả với tham chiếu đến Fig.47.

Theo một phương án của sáng chế, giá trị dựa trên giá trị lớn hơn trong số m và n có thể được báo hiệu cho phần tử cú pháp được thiết lập trước trong số merge_triangle_idx0 và merge_triangle_idx1. Ngoài ra, giá trị dựa trên giá trị nhỏ hơn trong số m và n có thể được báo hiệu cho phần tử cú pháp còn lại trong số merge_triangle_idx0 và merge_triangle_idx1. Ngoài ra, có thể báo hiệu giá trị biểu thị mối quan hệ độ lớn của m và n.

Ví dụ, merge_triangle_idx0 có thể dựa trên giá trị lớn hơn trong số m và n. Theo một phương án, vì m và n không bằng nhau, do đó giá trị lớn hơn trong số m và n sẽ lớn hơn hoặc bằng 1. Do đó, có thể báo hiệu với ít bit hơn xét rằng giá trị lớn hơn trong số m và n, mà là 0, được loại trừ. Ví dụ, merge_triangle_idx0 có thể là ((giá trị lớn hơn trong số m và n) - 1). Trong trường hợp này, giá trị lớn nhất của merge_triangle_idx0 có thể là (MaxNumTriangleMergeCand - 1 - 1) (-1 vì nó là giá trị bắt đầu từ 0, và -1 vì giá trị lớn hơn là 0 có thể được loại trừ). Giá trị lớn nhất có thể được sử dụng khi nhị phân hóa, và khi giá trị lớn nhất được làm giảm, thì có thể có các trường hợp trong đó ít bit hơn được sử dụng. Ngoài ra, merge_triangle_idx1 có thể là giá trị nhỏ hơn trong số m và n. Ngoài ra, giá trị lớn nhất của merge_triangle_idx1 có thể là merge_triangle_idx0. Do đó, có thể có trường hợp trong đó ít bit hơn so với thiết lập giá trị lớn nhất thành MaxNumTriangleMergeCand được sử dụng. Ngoài ra, khi merge_triangle_idx0 là 0, nghĩa là, khi giá trị lớn hơn trong số m và n là 1, thì giá trị nhỏ hơn trong số m và n là 0, và do đó báo hiệu khác có thể không tồn tại. Ví dụ, khi merge_triangle_idx0 là 0, nghĩa là, khi giá trị lớn hơn trong số m và n là 1, thì giá trị nhỏ hơn trong số m và n có thể được xác định là 0. Theo cách khác, khi merge_triangle_idx0 là 0, nghĩa là, khi giá trị lớn hơn trong số m và

n là 1, thì `merge_triangle_idx1` có thể được suy ra là bằng và được xác định là 0. Dựa vào Fig.22, có thể xác định xem có phân tích `merge_triangle_idx1` hay không trên cơ sở `merge_triangle_idx0`. Ví dụ, khi `merge_triangle_idx0` lớn hơn 0, thì `merge_triangle_idx1` có thể được phân tích, và khi `merge_triangle_idx0` là 0, thì `merge_triangle_idx1` có thể không được phân tích.

Ngoài ra, giá trị biểu diễn mối quan hệ độ lớn của m và n có thể là `merge_triangle_bigger` của Fig.55.

Nhờ sử dụng mối quan hệ này, m và n có thể được xác định trên cơ sở `merge_triangle_idx0`, `merge_triangle_idx1` và `merge_triangle_bigger`. Dựa vào Fig.55, hoạt động khác có thể được thực hiện trên cơ sở giá trị của `merge_triangle_bigger`. Ví dụ, khi `merge_triangle_bigger` là 0, thì m có thể lớn hơn n . Trong trường hợp này, m có thể là $(\text{merge_triangle_idx0} + 1)$. Ngoài ra, n có thể là `merge_triangle_idx1`. Ngoài ra, khi `merge_triangle_bigger` là 1, n có thể lớn hơn m . Trong trường hợp này, n có thể là $(\text{merge_triangle_idx0} + 1)$. Ngoài ra, m có thể là `merge_triangle_idx1`. Ngoài ra, khi `merge_triangle_idx1` không tồn tại, thì giá trị của nó có thể được suy ra là bằng 0.

Phương pháp trên Fig.55 có ưu điểm là tổng phí báo hiệu có thể được làm giảm theo các giá trị m và n , so với phương pháp trên Fig.47. Ví dụ, khi m và n lần lượt là 1 và 0, theo phương pháp của Fig.47, `merge_triangle_idx0` và `merge_triangle_idx1` có thể cần được báo hiệu lần lượt là 1 và 0. Tuy nhiên, theo phương pháp của Fig.55, khi m và n lần lượt là 1 và 0, `merge_triangle_idx0` và `merge_triangle_idx1` có thể cần được báo hiệu lần lượt là 0 và 0, nhưng `merge_triangle_idx1` có thể được suy ra mà không mã hóa hoặc phân tích (ngoài ra, báo hiệu biểu thị mối quan hệ độ lớn có thể cần thiết). Do đó, khi báo hiệu chiều dài khả biến được sử dụng, thì có thể sử dụng ít bit hơn vì kích thước của các giá trị mã hóa và giải mã được giảm bớt. Theo cách khác, khi m và n lần lượt là 2 và 1, có thể cần báo hiệu `merge_triangle_idx0` và `merge_triangle_idx1` lần lượt là 2 và 1, theo phương pháp của Fig.47, và `merge_triangle_idx0` và `merge_triangle_idx1` lần lượt là 1 và 1, theo phương pháp của Fig.55. Tuy nhiên, trong trường hợp này, theo phương pháp của Fig.55, vì giá trị lớn nhất của `merge_triangle_idx1` là $(3 - 1 - 1) = 1$, 1 có thể được báo hiệu với ít bit hơn so với khi giá trị lớn nhất là lớn. Ví dụ, phương pháp của Fig.55 có thể là phương pháp có lợi thế khi chênh lệch giữa m và n là nhỏ, ví dụ, khi chênh lệch này là 1.

Cũng được mô tả rằng `merge_triangle_idx0` được tham chiếu đến để xác định xem có phân tích `merge_triangle_idx1` hay không trong cấu trúc cú pháp của Fig.55, nhưng cũng

có thể xác định xem có phân tích `merge_triangle_idx1` hay không trên cơ sở giá trị lớn hơn trong số m và n . Nghĩa là, nó có thể được phân loại thành trường hợp trong đó giá trị lớn hơn trong số m và n lớn hơn hoặc bằng 1 và trường hợp trong đó nó không phải là như vậy. Tuy nhiên, trong trường hợp này, việc phân tích `merge_triangle_bigger` có thể cần xảy ra trước khi xác định xem có phân tích `merge_triangle_idx1` hay không.

Trong phần mô tả ở trên, cấu hình đã được mô tả thông qua các phương án cụ thể, nhưng người có hiểu biết trung bình trong lĩnh vực kỹ thuật có thể thực hiện các sự cải biến và các sự thay đổi mà không lệch khỏi nguyên lý và phạm vi của sáng chế. Do đó, điều có thể được suy ra dễ dàng bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật mà sáng chế thuộc về từ phần mô tả chi tiết và các phương án của sáng chế được hiểu là nằm trong phạm vi bảo hộ của sáng chế.

YÊU CẦU BẢO HỘ

1. Phương pháp xử lý tín hiệu video, trong đó phương pháp này bao gồm các bước:

phân tích, từ dòng bit, phần tử cú pháp thứ nhất biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được cho phép hay không;

phân tích, từ dòng bit, phần tử cú pháp thứ hai biểu thị xem phép bù chuyển động afin có được cho phép hay không; và

khi phần tử cú pháp thứ nhất biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng được cho phép và phần tử cú pháp thứ hai biểu thị rằng phép bù chuyển động afin được cho phép, phân tích, từ dòng bit, phần tử cú pháp thứ ba biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được cho phép đối với phép bù chuyển động afin hay không.

2. Phương pháp theo điểm 1, trong đó ít nhất một trong số phần tử cú pháp thứ nhất, phần tử cú pháp thứ hai và/hoặc phần tử cú pháp thứ ba được báo hiệu dưới dạng đơn vị chuỗi.

3. Phương pháp theo điểm 1, trong đó, khi phần tử cú pháp thứ nhất biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng không được cho phép và phần tử cú pháp thứ hai biểu thị rằng phép bù chuyển động afin được cho phép, thì giá trị của phần tử cú pháp thứ ba được suy ra là giá trị biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng là không được cho phép đối với phép bù chuyển động afin.

4. Phương pháp theo điểm 1, trong đó, khi phần tử cú pháp thứ hai biểu thị rằng phép bù chuyển động afin không được cho phép, thì giá trị của phần tử cú pháp thứ ba được suy ra là giá trị biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng là không được cho phép đối với phép bù chuyển động afin.

5. Phương pháp theo điểm 1, trong đó phương pháp này còn bao gồm các bước:

phân tích, từ dòng bit, phần tử cú pháp thứ tư biểu thị xem phép bù chuyển động afin có được sử dụng cho khối hiện thời hay không;

khi phần tử cú pháp thứ nhất biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng được cho phép, phần tử cú pháp thứ tư biểu thị rằng phép bù chuyển động afin không được sử dụng cho khối hiện thời và ít nhất một trong số các chênh lệch vector chuyển động cho khối hiện thời là khác không, phân tích, từ dòng bit, phần tử cú pháp thứ năm biểu thị độ phân giải của chênh lệch vector chuyển động; và

sửa đổi các chênh lệch vector chuyển động cho khối hiện thời trên cơ sở phần tử cú pháp thứ năm.

6. Phương pháp theo điểm 1, trong đó phương pháp này còn bao gồm các bước:

phân tích, từ dòng bit, phần tử cú pháp thứ tư biểu thị xem phép bù chuyển động afin có được sử dụng cho khối hiện thời hay không;

khi phần tử cú pháp thứ ba biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng được cho phép đối với phép bù chuyển động afin, phần tử cú pháp thứ tư biểu thị rằng phép bù chuyển động afin được sử dụng cho khối hiện thời và ít nhất một trong số các chênh lệch vector chuyển động điểm điều khiển đối với khối hiện thời là khác không, phân tích, từ dòng bit, phần tử cú pháp thứ năm biểu thị độ phân giải của chênh lệch vector chuyển động; và

sửa đổi các chênh lệch vector chuyển động điểm điều khiển đối với khối hiện thời trên cơ sở phần tử cú pháp thứ năm.

7. Phương pháp theo điểm 1, trong đó phương pháp này còn bao gồm các bước:

phân tích, từ dòng bit, phần tử cú pháp thứ sáu liên quan đến danh mục ảnh tham chiếu cho khối hiện thời,

khi phần tử cú pháp thứ sáu biểu thị rằng danh mục ảnh tham chiếu khác với danh mục ảnh tham chiếu 0 là khả dụng, phân tích, từ dòng bit, phần tử cú pháp thứ bảy biểu thị chỉ số dự đoán vector chuyển động của danh mục ảnh tham chiếu 1.

8. Phương pháp theo điểm 7, trong đó phương pháp này còn bao gồm các bước:

phân tích, từ dòng bit, phần tử cú pháp thứ tám biểu thị xem chênh lệch vector chuyển động và các chênh lệch vector chuyển động điểm điều khiển có được thiết lập thành không đối với danh mục ảnh tham chiếu 1 hay không,

trong đó phần tử cú pháp thứ bảy được phân tích bất kể phần tử cú pháp thứ tám.

9. Thiết bị giải mã để xử lý tín hiệu video, thiết bị giải mã này bao gồm:

bộ xử lý,

trong đó bộ xử lý này được tạo cấu hình để:

phân tích, từ dòng bit, phần tử cú pháp thứ nhất biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được cho phép hay không;

phân tích, từ dòng bit, phần tử cú pháp thứ hai biểu thị xem phép bù chuyển động afin có được cho phép hay không; và

khi phần tử cú pháp thứ nhất biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng được cho phép và phần tử cú pháp thứ hai biểu thị rằng phép bù chuyển động afin được cho phép, phân tích, từ dòng bit, phần tử cú pháp thứ ba biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được cho phép đối với phép bù chuyển động afin hay không.

10. Thiết bị giải mã theo điểm 9, trong đó ít nhất một trong số phần tử cú pháp thứ nhất, phần tử cú pháp thứ hai và/hoặc phần tử cú pháp thứ ba được báo hiệu dưới dạng đơn vị chuỗi.

11. Thiết bị giải mã theo điểm 9, trong đó, khi phần tử cú pháp thứ nhất biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng không được cho phép và phần tử cú pháp thứ hai biểu thị rằng phép bù chuyển động afin được cho phép, thì giá trị của phần tử cú pháp thứ ba được suy ra là giá trị biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng không được cho phép đối với phép bù chuyển động afin.

12. Thiết bị giải mã theo điểm 9, trong đó, khi phần tử cú pháp thứ hai biểu thị rằng phép bù chuyển động afin không được cho phép, thì giá trị của phần tử cú pháp thứ ba được suy ra là giá trị biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng không được cho phép đối với phép bù chuyển động afin.

13. Thiết bị giải mã theo điểm 9,

trong đó bộ xử lý được tạo cấu hình để:

phân tích, từ dòng bit, phần tử cú pháp thứ tư biểu thị xem phép bù chuyển động afin có được sử dụng cho khối hiện thời hay không,

khi phần tử cú pháp thứ nhất biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng được cho phép, phần tử cú pháp thứ tư biểu thị rằng phép bù chuyển động afin không được sử dụng cho khối hiện thời, và ít nhất một trong số các chênh lệch vector chuyển động cho khối hiện thời là khác không, phân tích, từ dòng bit, phần tử cú pháp thứ năm biểu thị độ phân giải của chênh lệch vector chuyển động, và

sửa đổi các chênh lệch vector chuyển động cho khối hiện thời trên cơ sở phần tử cú pháp thứ năm.

14. Thiết bị giải mã theo điểm 9,

trong đó bộ xử lý được tạo cấu hình để:

phân tích, từ dòng bit, phần tử cú pháp thứ tư biểu thị xem phép bù chuyển động afin có được sử dụng cho khối hiện thời hay không,

khi phần tử cú pháp thứ ba biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng được cho phép đối với phép bù chuyển động afin, phần tử cú pháp thứ tư biểu thị rằng phép bù chuyển động afin được sử dụng cho khối hiện thời và ít nhất một trong số các chênh lệch vector chuyển động điểm điều khiển đối với khối hiện thời là khác không, phân tích, từ dòng bit, phần tử cú pháp thứ năm biểu thị độ phân giải của chênh lệch vector chuyển động, và

sửa đổi các chênh lệch vector chuyển động điểm điều khiển đối với khối hiện thời trên cơ sở phần tử cú pháp thứ năm.

15. Thiết bị giải mã theo điểm 9,

trong đó bộ xử lý được tạo cấu hình để:

phân tích, từ dòng bit, phần tử cú pháp thứ sáu liên quan đến danh mục ảnh tham chiếu cho khối hiện thời,

khi phần tử cú pháp thứ sáu biểu thị rằng danh mục ảnh tham chiếu khác với danh mục ảnh tham chiếu 0 là khả dụng, phân tích, từ dòng bit, phần tử cú pháp thứ bảy biểu thị chỉ số dự đoán vector chuyển động của danh mục ảnh tham chiếu 1.

16. Thiết bị giải mã theo điểm 15,

trong đó bộ xử lý được tạo cấu hình để:

phân tích, từ dòng bit, phần tử cú pháp thứ tám biểu thị xem chênh lệch vector chuyển động và các chênh lệch vector chuyển động điểm điều khiển có được thiết lập thành không đối với danh mục ảnh tham chiếu 1 hay không,

trong đó phần tử cú pháp thứ bảy được phân tích bất kể phần tử cú pháp thứ tám.

17. Thiết bị mã hóa để xử lý tín hiệu video, thiết bị mã hóa này bao gồm:

bộ xử lý,

trong đó bộ xử lý này được tạo cấu hình để thu dòng bit cần được giải mã bởi bộ giải mã nhờ sử dụng phương pháp giải mã,

trong đó phương pháp giải mã này bao gồm các bước:

phân tích, từ dòng bit, phần tử cú pháp thứ nhất biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được cho phép hay không;

phân tích, từ dòng bit, phần tử cú pháp thứ hai biểu thị xem phép bù chuyển động afin có được cho phép hay không; và

khi phần tử cú pháp thứ nhất biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng được cho phép và phần tử cú pháp thứ hai biểu thị rằng phép bù chuyển động afin được cho phép, phân tích, từ dòng bit, phần tử cú pháp thứ ba biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được cho phép đối với phép bù chuyển động afin hay không.

18. Thiết bị mã hóa theo điểm 17, trong đó ít nhất một trong số phần tử cú pháp thứ nhất, phần tử cú pháp thứ hai và/hoặc phần tử cú pháp thứ ba được báo hiệu dưới dạng đơn vị chuỗi.

19. Thiết bị mã hóa theo điểm 17, trong đó, khi phần tử cú pháp thứ nhất biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng không được cho phép và phần tử cú pháp thứ hai biểu thị rằng phép bù chuyển động afin được cho phép, thì giá trị của phần tử cú pháp thứ ba được suy ra là giá trị biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng không được cho phép đối với phép bù chuyển động afin.

20. Phương pháp tạo ra dòng bit, trong đó phương pháp này bao gồm các bước:

tạo ra phần tử cú pháp thứ nhất biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được cho phép hay không;

tạo ra phần tử cú pháp thứ hai biểu thị xem phép bù chuyển động afin có được cho phép hay không;

khi phần tử cú pháp thứ nhất biểu thị rằng độ phân giải vi sai vector chuyển động thích ứng được cho phép và phần tử cú pháp thứ hai biểu thị rằng phép bù chuyển động afin được cho phép, tạo ra phần tử cú pháp thứ ba biểu thị xem độ phân giải vi sai vector chuyển động thích ứng có được cho phép đối với phép bù chuyển động afin hay không; và

tạo ra dòng bit bao gồm phần tử cú pháp thứ nhất, phần tử cú pháp thứ hai và phần tử cú pháp thứ ba.

1/39

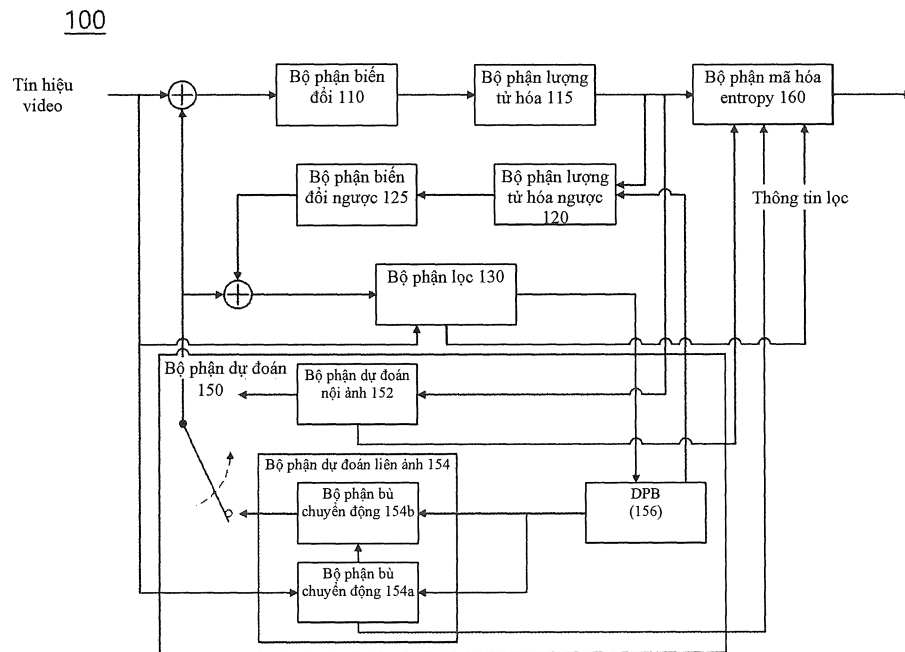


FIG. 1

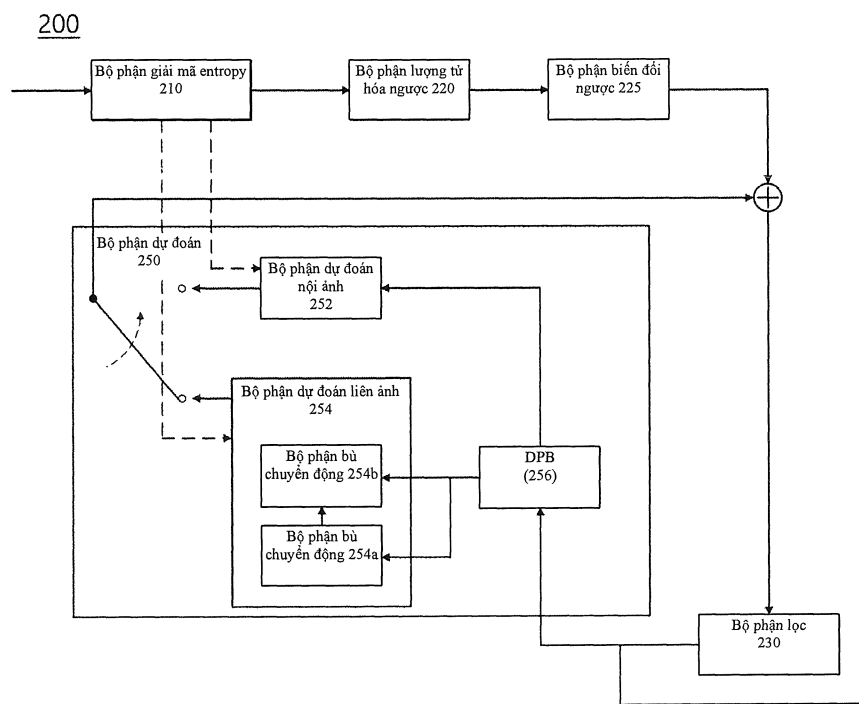
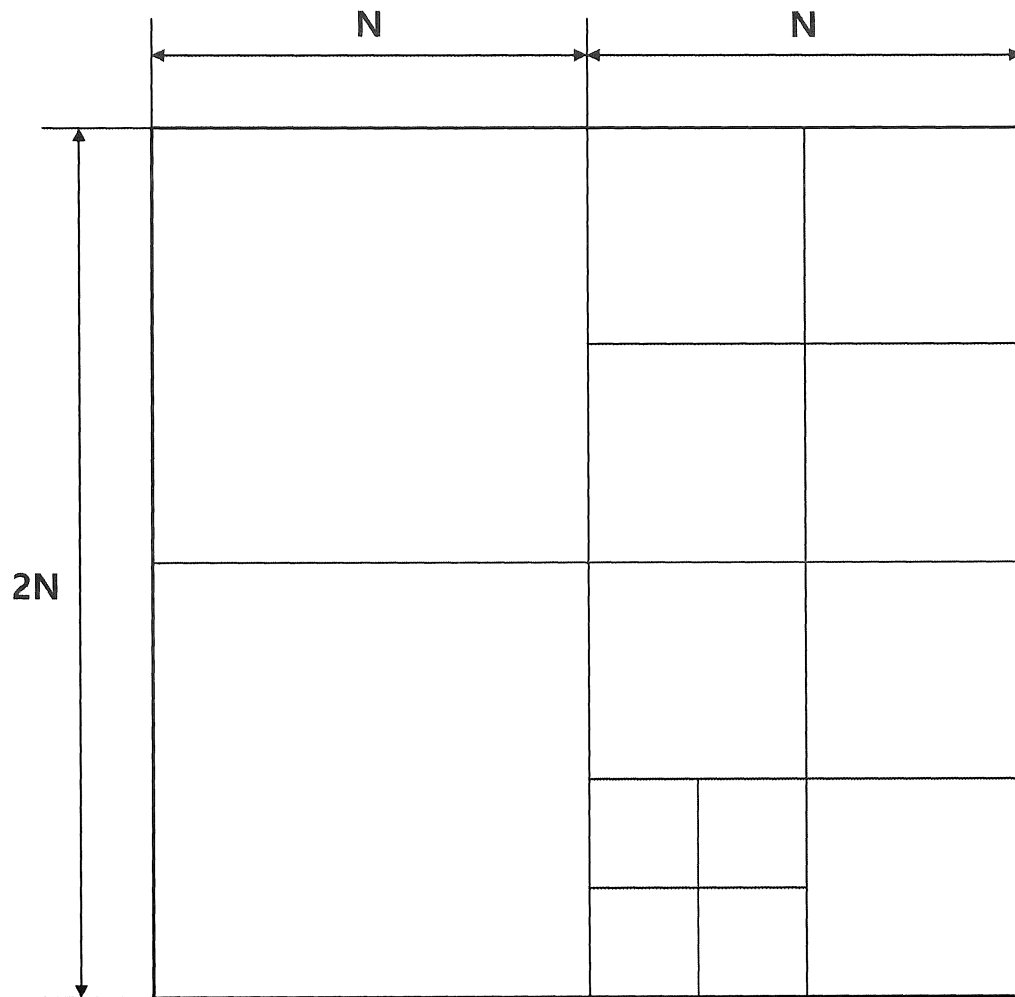
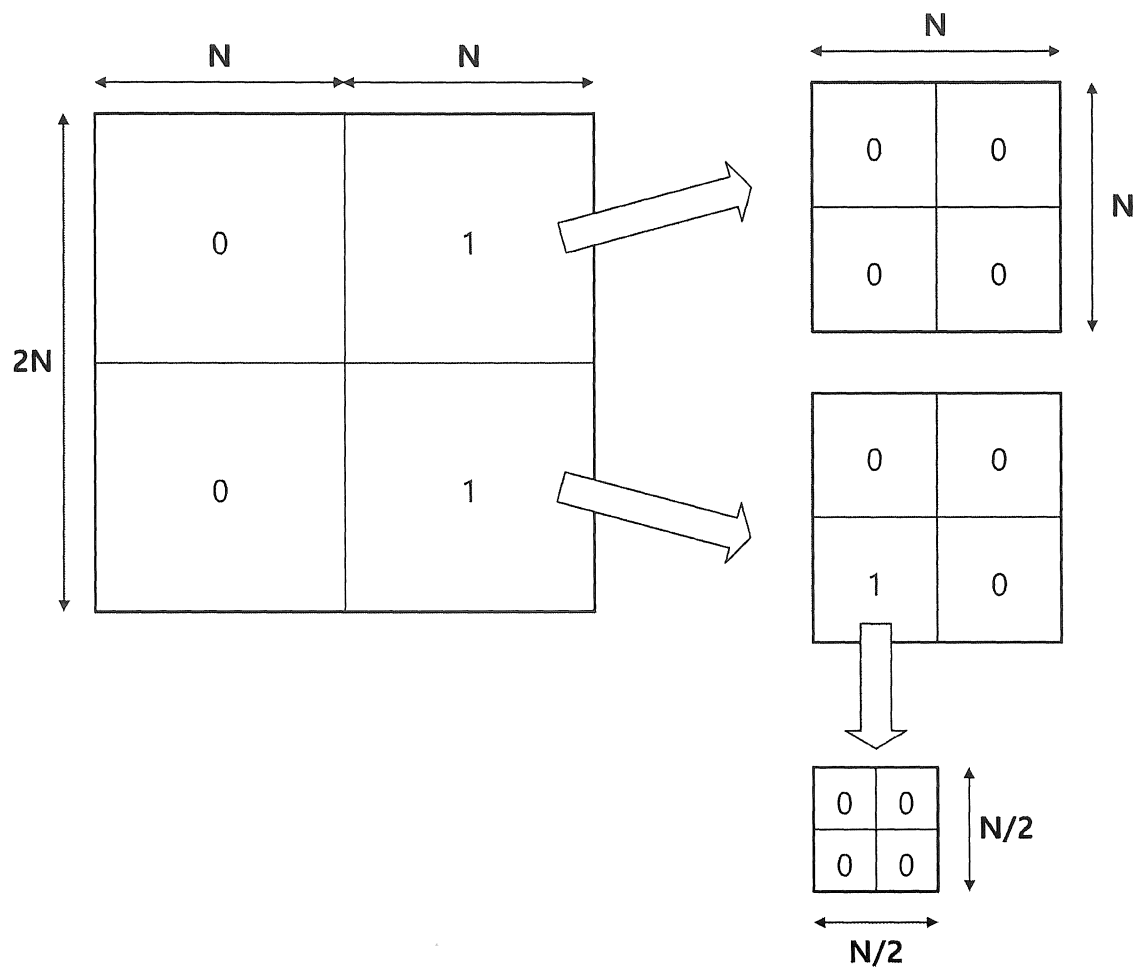


FIG. 2

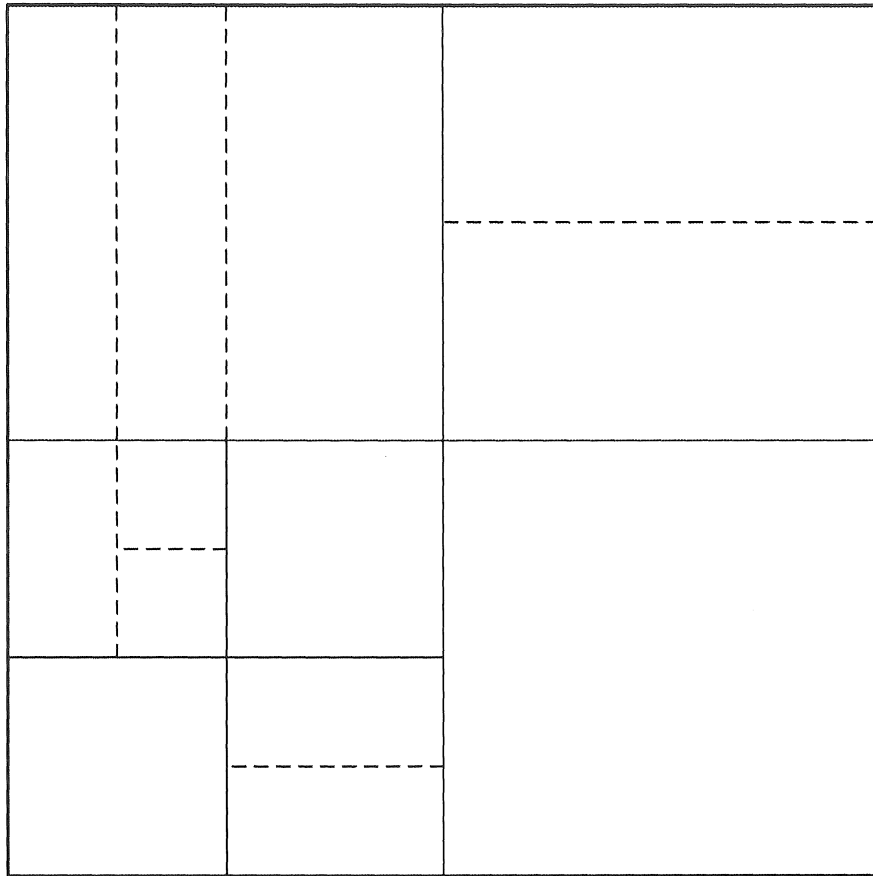
2/39

*FIG. 3*

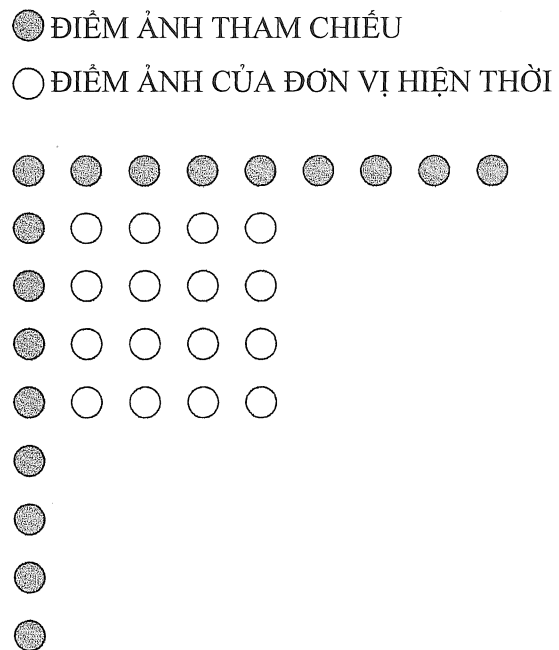
3/39

*FIG. 4*

4/39

*FIG. 5*

5/39

*FIG. 6*

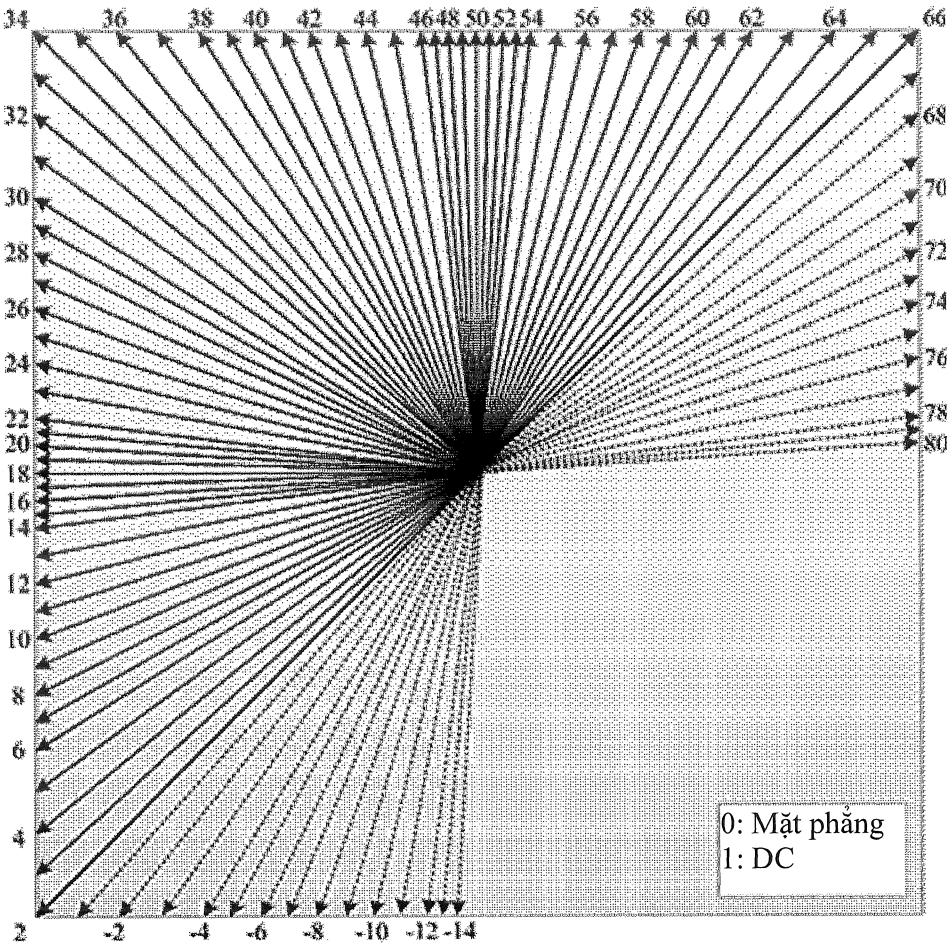
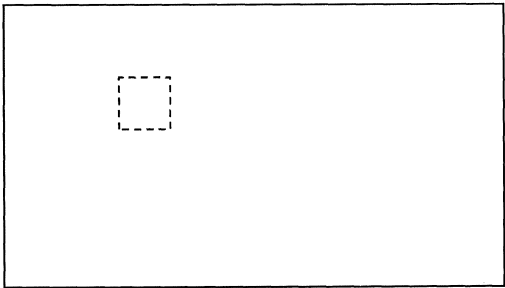


FIG. 7

Ảnh tham chiếu



Ảnh hiện thời

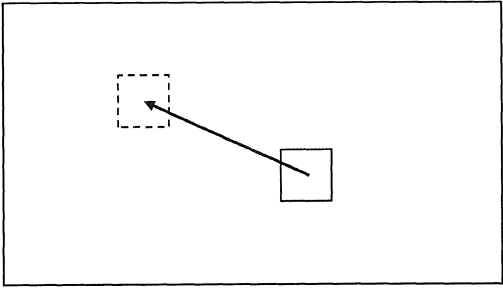


FIG. 8

7/39

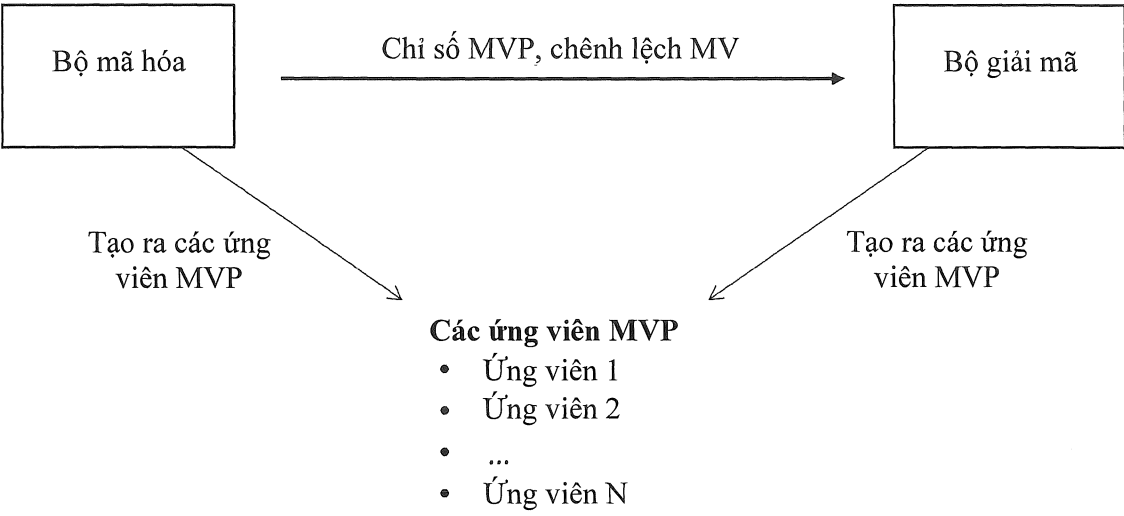
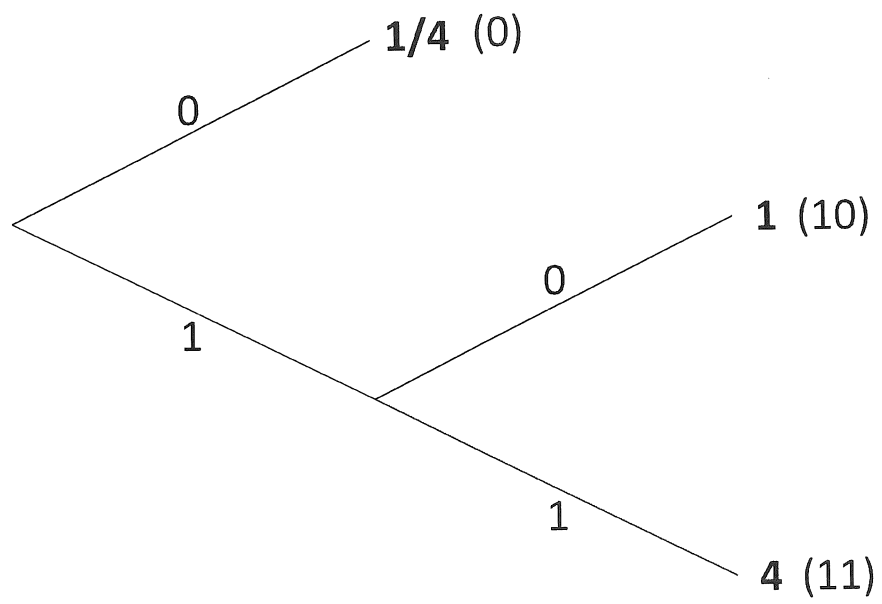


FIG. 9

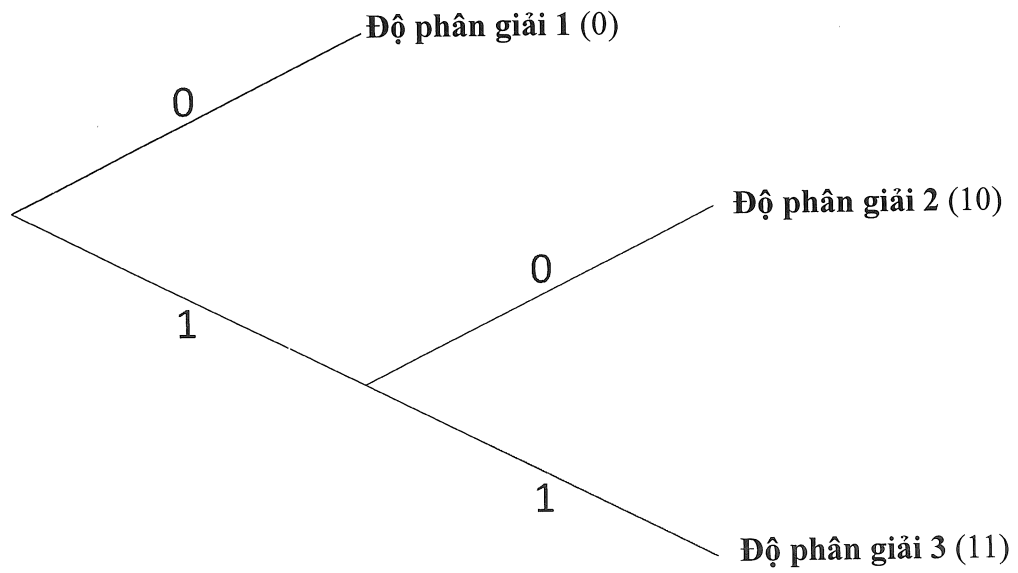
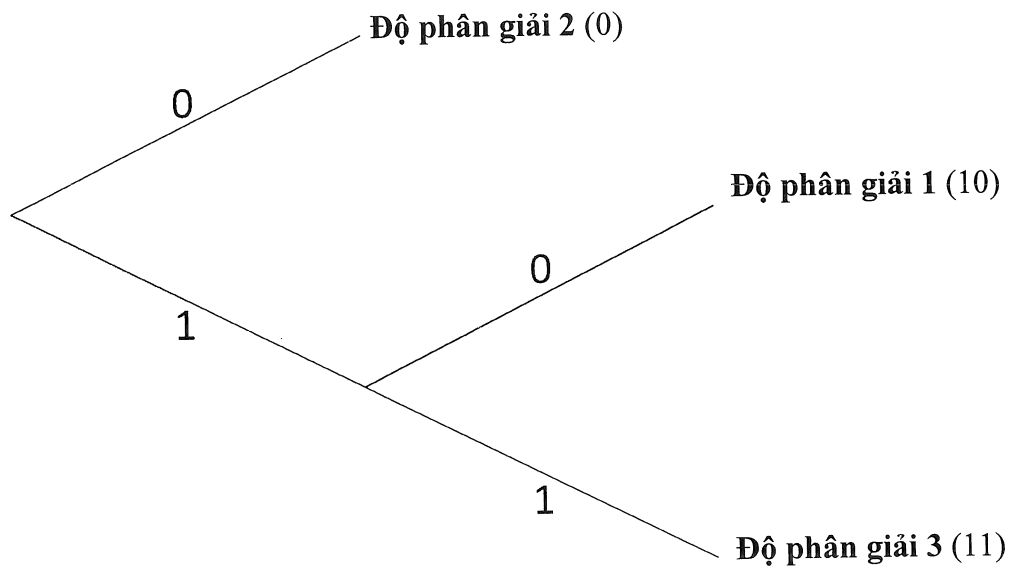
mvd_coding(x0, y0, refList) {	Bộ mô tả
abs_mvd_greater0_flag[0]	ae(v)
abs_mvd_greater0_flag[1]	ae(v)
if(abs_mvd_greater0_flag[0])	
abs_mvd_greater1_flag[0]	ae(v)
if(abs_mvd_greater0_flag[1])	
abs_mvd_greater1_flag[1]	ae(v)
if(abs_mvd_greater0_flag[0]) {	
if(abs_mvd_greater1_flag[0])	
abs_mvd_minus2[0]	ae(v)
mvd_sign_flag[0]	ae(v)
}	
if(abs_mvd_greater0_flag[1]) {	
if(abs_mvd_greater1_flag[1])	
abs_mvd_minus2[1]	ae(v)
mvd_sign_flag[1]	ae(v)
}	
}	

FIG. 10

8/39

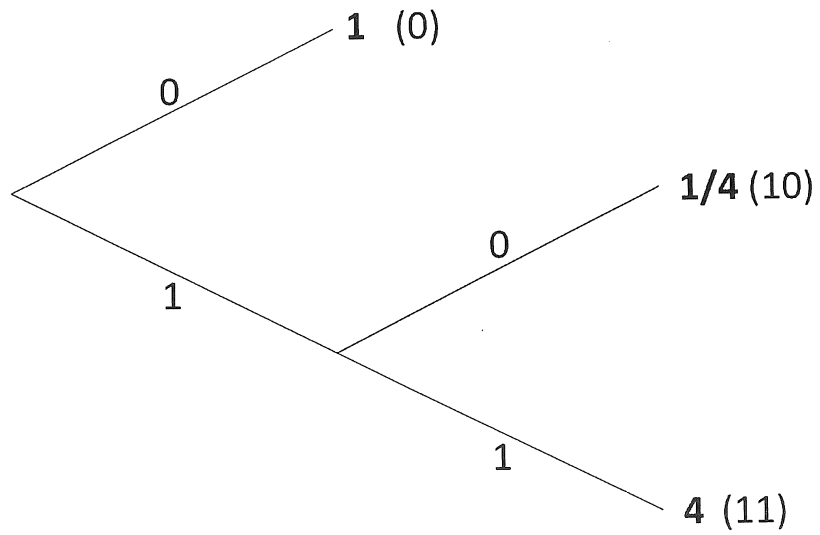
*FIG. 11*

9/39

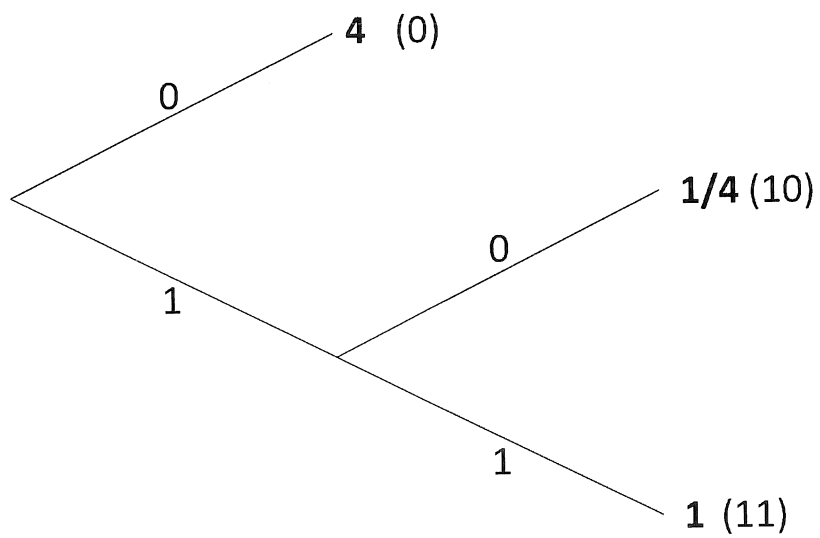
Trường hợp 1*Trường hợp 2***FIG. 12**

10/39

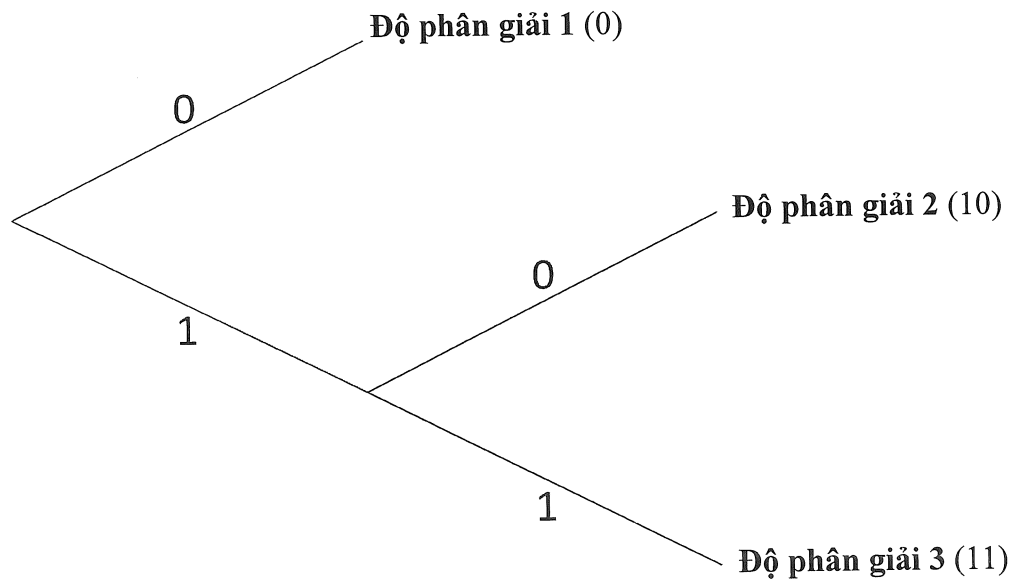
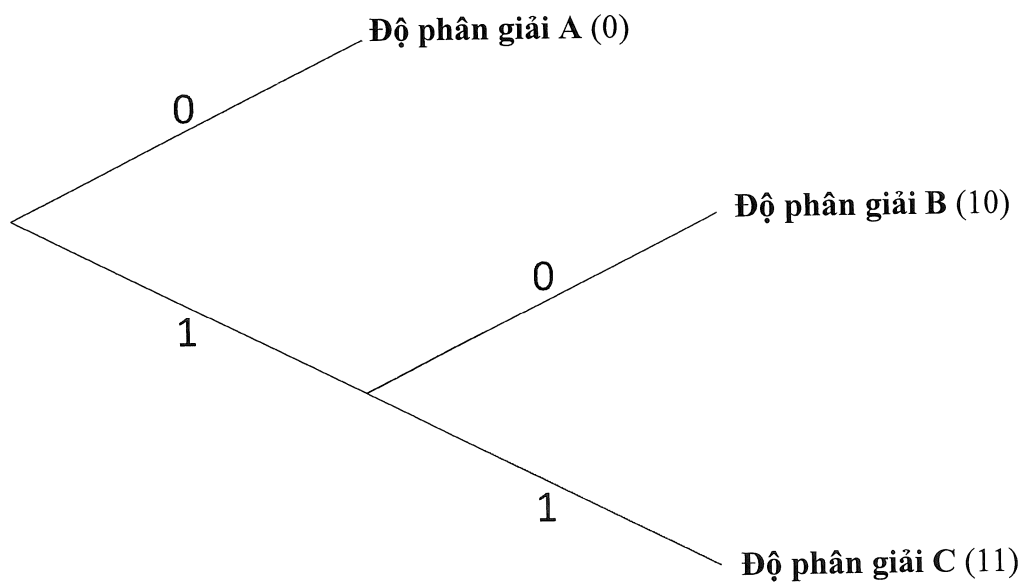
(a)



(b)

**FIG. 13**

11/39

Trường hợp 1*Trường hợp 2***FIG. 14**

12/39

Các ứng viên MVP

- Ứng viên 1
- Ứng viên 2
- ...
- Ứng viên N

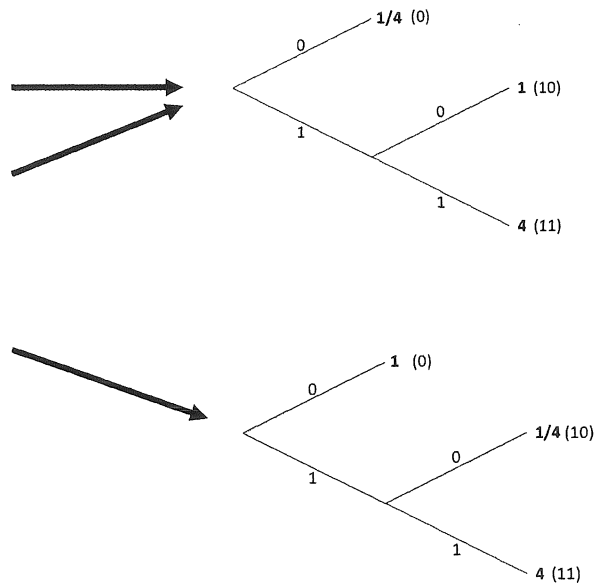


FIG. 15

Các ứng viên MVP

- Ứng viên 1
- Ứng viên 2
- ...
- Ứng viên N
(ứng viên thời gian, vectơ chuyển động bằng không, ứng viên không gian không liên kết, hoặc có hoặc không có sự tinh chỉnh)

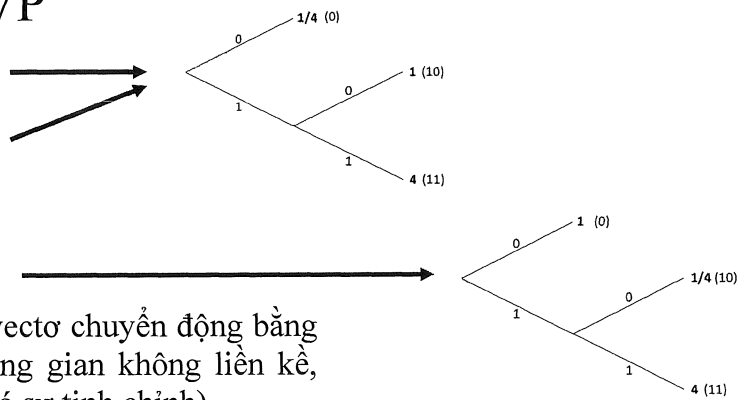
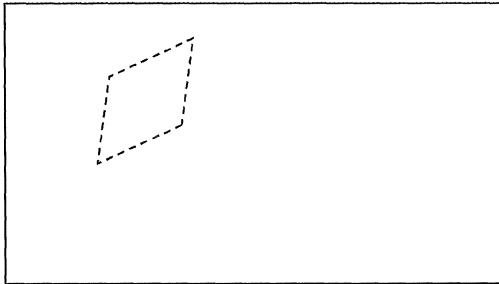


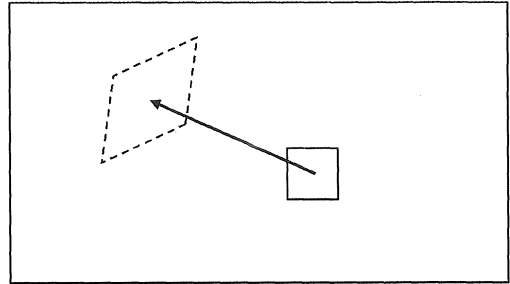
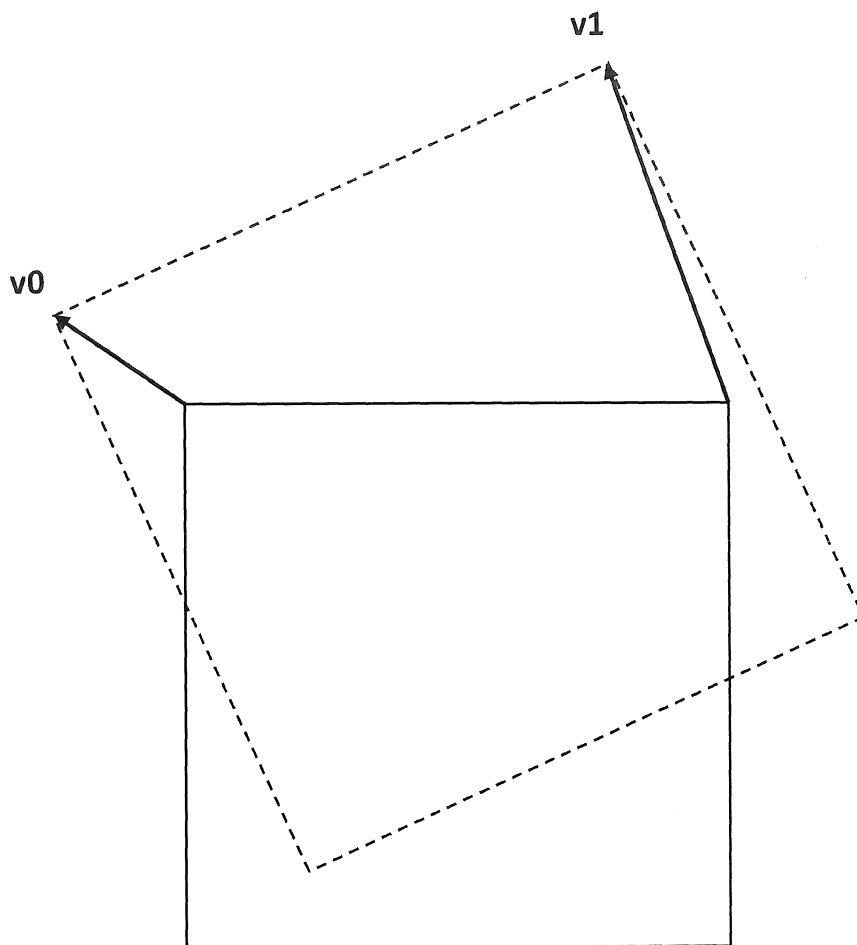
FIG. 16

13/39

Ảnh tham chiếu

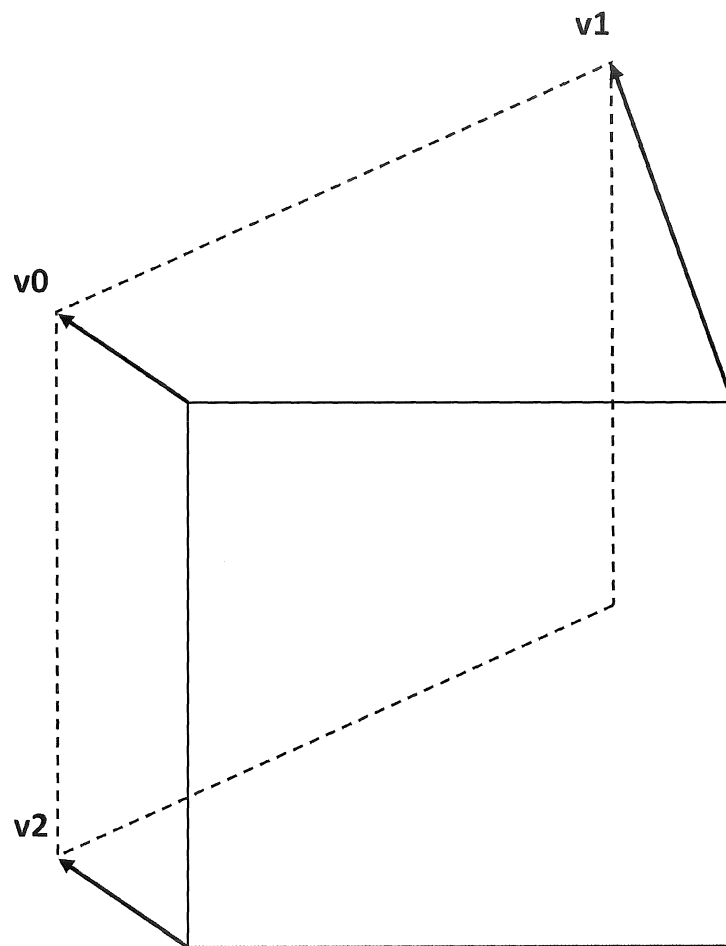


Ảnh hiện thời

*FIG. 17**FIG. 18*

14/39

$$\begin{cases} v_x = \frac{(v_{1x} - v_{0x})}{w} x - \frac{(v_{1y} - v_{0y})}{w} y + v_{0x} \\ v_y = \frac{(v_{1y} - v_{0y})}{w} x + \frac{(v_{1x} - v_{0x})}{w} y + v_{0y} \end{cases}$$

FIG. 19*FIG. 20*

15/39

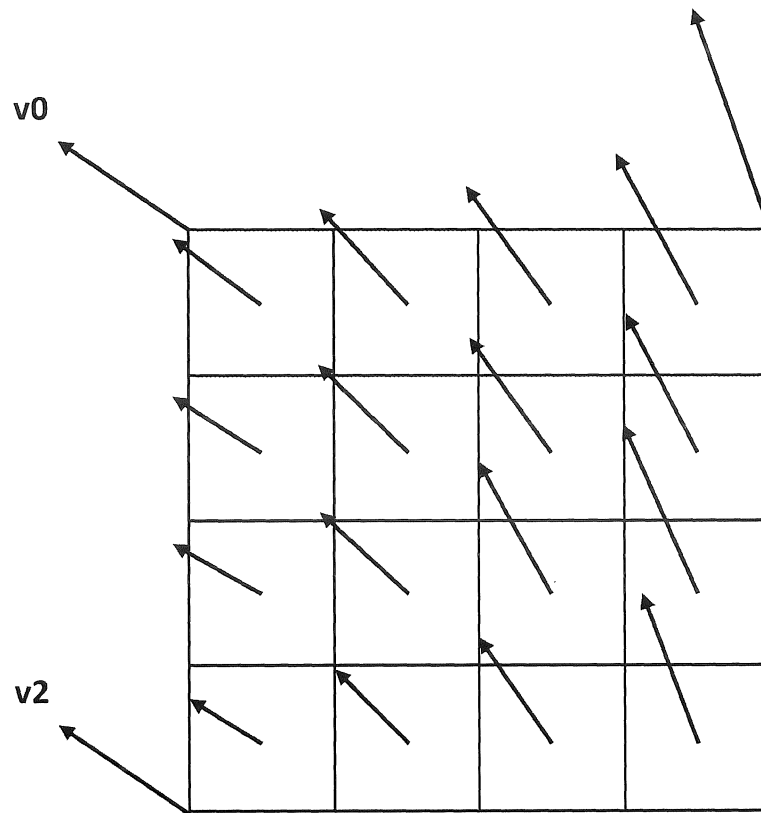
$$mv^x = \frac{(mv_1^x - mv_0^x)}{w}x + \frac{(mv_2^x - mv_0^x)}{h}y + mv_0^x$$

$$mv^y = \frac{(mv_1^y - mv_0^y)}{w}x + \frac{(mv_2^y - mv_0^y)}{h}y + mv_0^y$$

FIG. 21

16/39

(a)



(b)

$$\begin{cases} M = clip3(4, w, \frac{w \times MvPre}{\max(abs(v_{1x} - v_{0x}), abs(v_{1y} - v_{0y}))}) \\ N = clip3(4, h, \frac{h \times MvPre}{\max(abs(v_{2x} - v_{0x}), abs(v_{2y} - v_{0y}))}) \end{cases}$$

FIG. 22

17/39

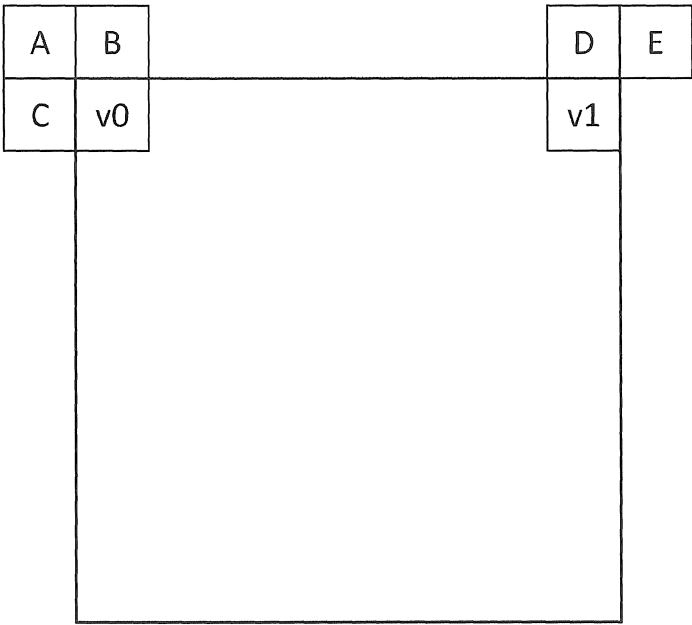
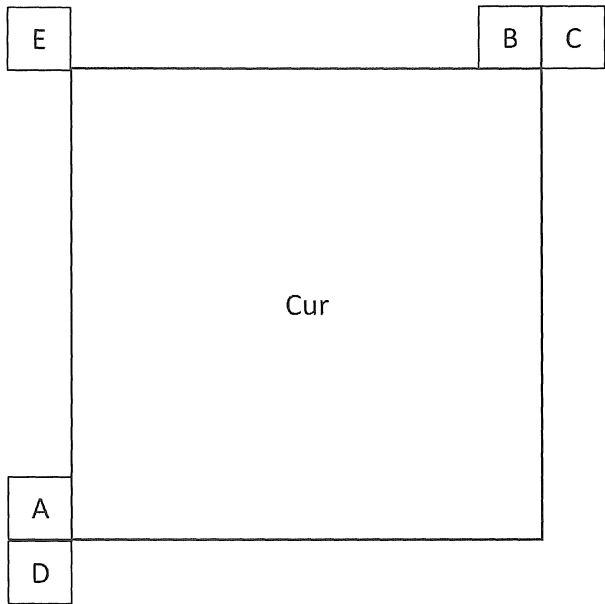


FIG. 23

18/39

(a)



(b)

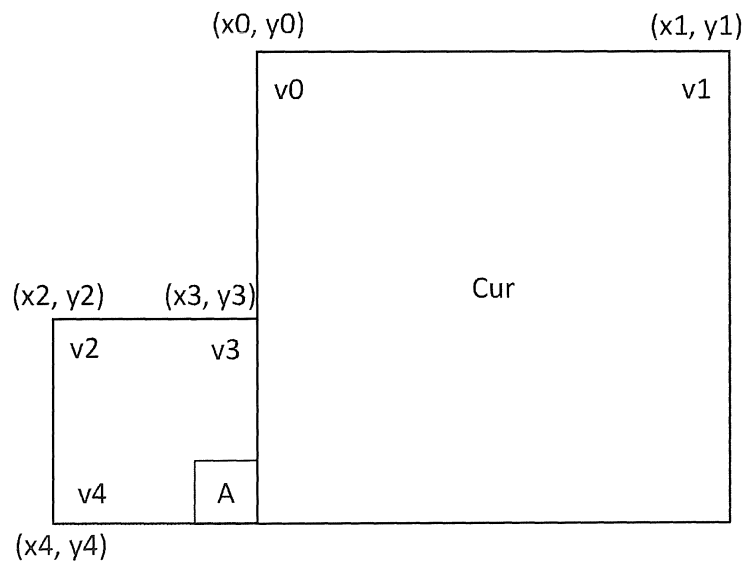
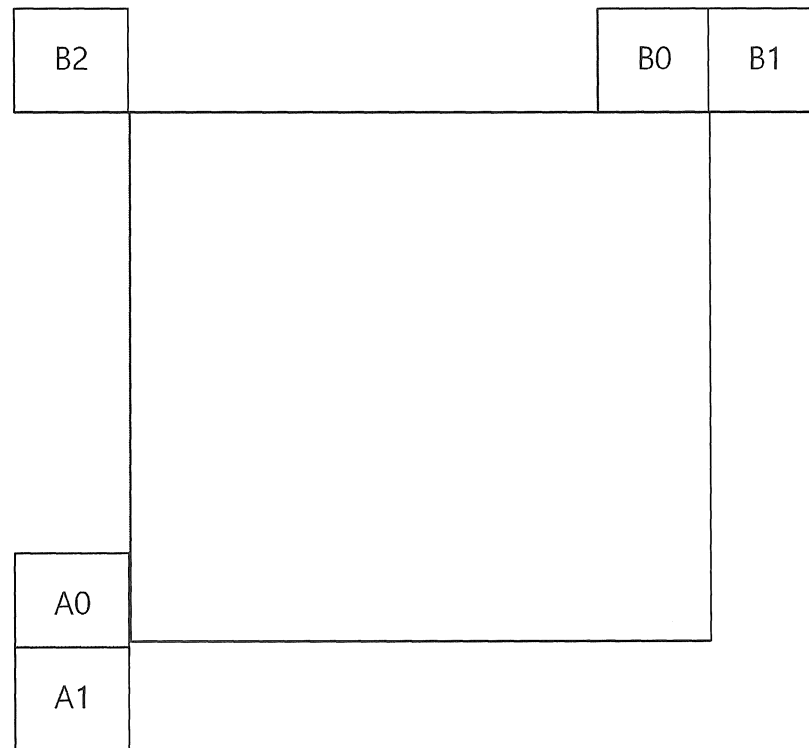


FIG. 24

19/39

*FIG. 25*

20/39

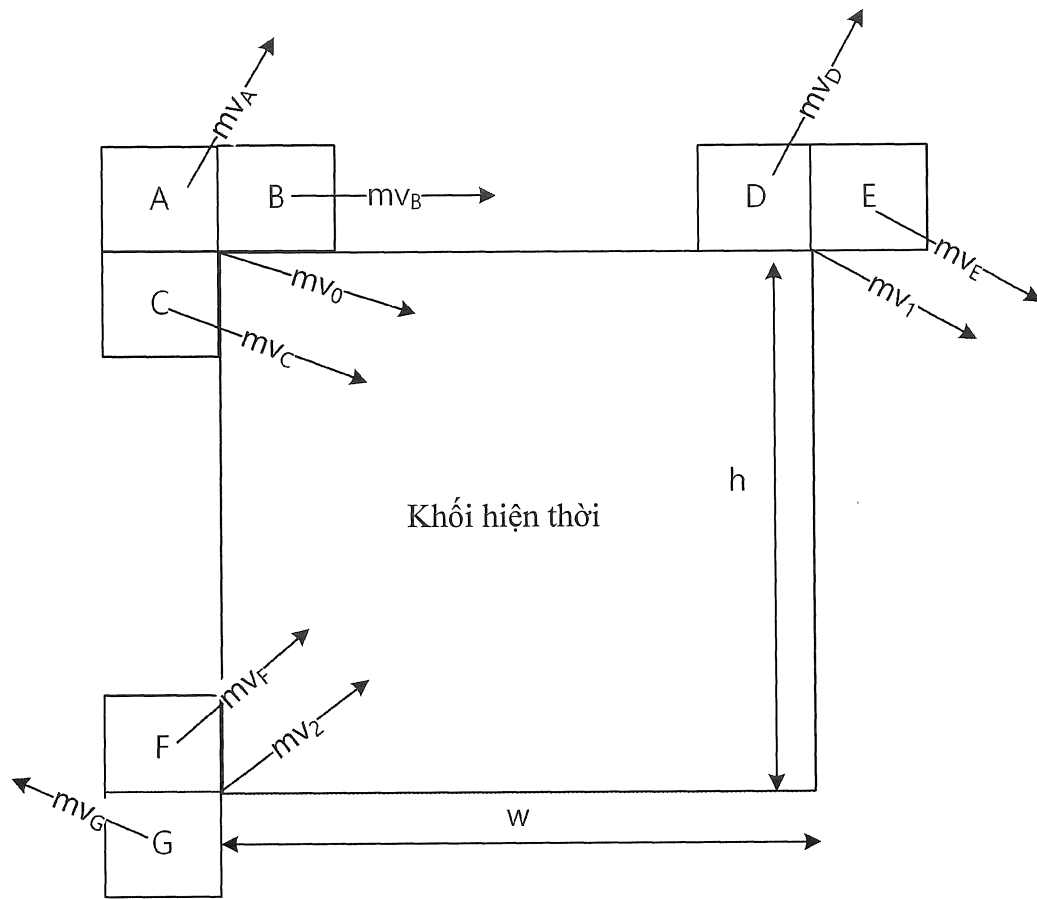


FIG. 26

$$\begin{cases} v_{0x} = \frac{(v_{E1x} - v_{E0x})}{(x_{E1} - x_{E0})} (x_0 - x_{E0}) + \frac{(v_{E2x} - v_{E0x})}{(y_{E2} - y_{E0})} (y_0 - y_{E0}) + v_{E0x} \\ v_{0y} = \frac{(v_{E1y} - v_{E0y})}{(x_{E1} - x_{E0})} (x_0 - x_{E0}) + \frac{(v_{E2y} - v_{E0y})}{(y_{E2} - y_{E0})} (y_0 - y_{E0}) + v_{E0y} \end{cases}$$

$$\begin{cases} v_{1x} = \frac{(v_{E1x} - v_{E0x})}{(x_{E1} - x_{E0})} (x_1 - x_{E0}) + \frac{(v_{E2x} - v_{E0x})}{(y_{E2} - y_{E0})} (y_1 - y_{E0}) + v_{E0x} \\ v_{1y} = \frac{(v_{E1y} - v_{E0y})}{(x_{E1} - x_{E0})} (x_1 - x_{E0}) + \frac{(v_{E2y} - v_{E0y})}{(y_{E2} - y_{E0})} (y_1 - y_{E0}) + v_{E0y} \end{cases}$$

FIG. 27

21/39

- Nếu chỉ $\overline{mv}_0$ và $\overline{mv}_1$ có thể được tìm thấy, $\overline{mv}_2$ được suy ra như sau:

$$\overline{mv}_2^x = \overline{mv}_0^x - h \frac{(\overline{mv}_1^y - \overline{mv}_0^y)}{w}, \overline{mv}_2^y = \overline{mv}_0^y + h \frac{(\overline{mv}_1^x - \overline{mv}_0^x)}{w},$$

trong đó kích thước khối hiện thời là $W \times H$.

- Nếu chỉ $\overline{mv}_0$ và $\overline{mv}_2$ có thể được tìm thấy, $\overline{mv}_1$ được suy ra như sau:

$$\overline{mv}_1^x = \overline{mv}_0^x + h \frac{(\overline{mv}_2^y - \overline{mv}_0^y)}{w}, \overline{mv}_1^y = \overline{mv}_0^y - h \frac{(\overline{mv}_2^x - \overline{mv}_0^x)}{w}.$$

FIG. 28

(a)

$$\begin{aligned} mv_0 &= \overline{mv}_0 + mvd_0 \\ mv_1 &= \overline{mv}_1 + mvd_1 \end{aligned}$$

(b)

$$\begin{aligned} mv_0 &= \overline{mv}_0 + mvd_0 \\ mv_1 &= \overline{mv}_1 + mvd_1 \\ mv_2 &= \overline{mv}_2 + mvd_2 \end{aligned}$$

FIG. 29

$$\begin{aligned} \text{IMvd}[\text{compIdx}] &= \text{abs_mvd_greater0_flag}[\text{compIdx}] * \\ &\quad (\text{abs_mvd_minus2}[\text{compIdx}] + 2) * (1 - 2 * \text{mvd_sign_flag}[\text{compIdx}]) \end{aligned}$$

- Nếu refList bằng 0, thì MvdL0[x0][y0][compIdx] được thiết lập bằng IMvd[compIdx] đối với compIdx = 0..1.
- Theo cách khác (refList bằng 1), MvdL1[x0][y0][compIdx] được thiết lập bằng IMvd[compIdx] đối với compIdx = 0..1.

FIG. 30

22/39

(a)

$$mv_0 = \overline{mv}_0 + mvd_0$$

$$mv_1 = \overline{mv}_1 + mvd_1 + mvd_0$$

(b)

$$mv_0 = \overline{mv}_0 + mvd_0$$

$$mv_1 = \overline{mv}_1 + mvd_1 + mvd_0$$

$$mv_2 = \overline{mv}_2 + mvd_2 + mvd_0$$

AG. 31

$$lMvd[cpIdx][compIdx] = abs_mvd_greater0_flag[cpIdx][compIdx] * \\ (abs_mvd_minus2[cpIdx][compIdx] + 2) * (1 - 2*mvd_sign_flag[cpIdx][compIdx])$$

Nếu cpIdx bằng 0, thì MvdLX[x0][y0][cpIdx][compIdx] được thiết lập bằng lMvd[cpIdx][compIdx]
 Theo cách khác MvdLX[x0][y0][cpIdx][compIdx] được thiết lập bằng lMvd[cpIdx][compIdx] + lMvd[0][compIdx]

AG. 32

23/39

mvd_coding(x0, y0, refList , cpIdx) {	Bộ mô tả
abs_mvd_greater0_flag[cpIdx][0]	ae(v)
abs_mvd_greater0_flag[cpIdx][1]	ae(v)
if(abs_mvd_greater0_flag[cpIdx][0])	
abs_mvd_greater1_flag[cpIdx][0]	ae(v)
if(abs_mvd_greater0_flag[cpIdx][1])	
abs_mvd_greater1_flag[cpIdx][1]	ae(v)
if(abs_mvd_greater0_flag[cpIdx][0]) {	
if(abs_mvd_greater1_flag[cpIdx][0])	
abs_mvd_minus2[cpIdx][0]	ae(v)
mvd_sign_flag[cpIdx][0]	ae(v)
}	
if(abs_mvd_greater0_flag[cpIdx][1]) {	
if(abs_mvd_greater1_flag[cpIdx][1])	
abs_mvd_minus2[cpIdx][1]	ae(v)
mvd_sign_flag[cpIdx][1]	ae(v)
}	
}	

AG. 33

24/39

seq parameter set rbsp() {	Bộ mô tả
// ...	
sps temporal mvp enabled flag	u(1)
if(sps temporal mvp enabled flag)	
sps sbtmvp enabled flag	u(1)
sps amvr enabled flag	u(1)
sps bdof enabled flag	u(1)
sps affine amvr enabled flag	u(1)
sps dmvr enabled flag	u(1)
sps cclm enabled flag	u(1)
if(sps cclm enabled flag && chroma format idc == 1)	
sps cclm colocated chroma flag	u(1)
sps mts enabled flag	u(1)
if(sps mts enabled flag) {	
sps explicit mts intra enabled flag	u(1)
sps explicit mts inter enabled flag	u(1)
}	
sps sbt enabled flag	u(1)
if(sps sbt enabled flag)	
sps sbt max size 64 flag	u(1)
sps affine enabled flag	u(1)
if(sps affine enabled flag)	
sps affine type flag	u(1)
sps gbi enabled flag	u(1)
sps ibc enabled flag	u(1)
sps ciip enabled flag	u(1)
sps fpel mmvd enabled flag	u(1)
sps triangle enabled flag	u(1)
sps lmcs 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)
}	
}	
sps extension flag	u(1)
if(sps extension flag)	
while(more rbsp data())	
sps extension data flag	u(1)
rbsp trailing bits()	
}	

FIG. 34

25/39

} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
if(cu_skip_flag[x0][y0] == 0)	
merge_flag [x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
merge_data(x0, y0, cbWidth, cbHeight)	
} else if(CuPredMode[x0][y0] == MODE_IBC) {	
mvd_coding(x0, y0, 0, 0)	
mvp_l0_flag [x0][y0]	ae(v)
if(sps_amvr_enabled_flag && (MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0)) {	
amvr_precision_flag [x0][y0]	ae(v)
}	
} else {	
if(slice_type == B)	
inter_pred_idc [x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag [x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag [x0][y0]	ae(v)
}	
// sym_mvd_flag[x0][y0], ref_idx_l0[x0][y0], mvd_coding, mvp_l0_flag[x0][y0], ref_idx_l1[x0][y0], mvp_l1_flag[x0][y0]	
if((sps_amvr_enabled_flag && inter_affine_flag == 0 && (MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0 MvdL1[x0][y0][0] != 0 MvdL1[x0][y0][1] != 0)) (sps_affine_amvr_enabled_flag && inter_affine_flag[x0][y0] == 1 && (MvdCpL0[x0][y0][0][0] != 0 MvdCpL0[x0][y0][0][1] != 0 MvdCpL1[x0][y0][0][0] != 0 MvdCpL1[x0][y0][0][1] != 0 MvdCpL0[x0][y0][1][0] != 0 MvdCpL0[x0][y0][1][1] != 0 MvdCpL1[x0][y0][1][0] != 0 MvdCpL1[x0][y0][1][1] != 0 MvdCpL0[x0][y0][2][0] != 0 MvdCpL0[x0][y0][2][1] != 0 MvdCpL1[x0][y0][2][0] != 0 MvdCpL1[x0][y0][2][1] != 0))) {	
amvr_flag [x0][y0]	ae(v)
if(amvr_flag[x0][y0])	
amvr_precision_flag [x0][y0]	ae(v)
}	
if(sps_gbi_enabled_flag && inter_pred_idc[x0][y0] == PRED_BI && luma_weight_l0_flag[ref_idx_l0 [x0][y0]] == 0 && luma_weight_l1_flag[ref_idx_l1 [x0][y0]] == 0 && chroma_weight_l0_flag[ref_idx_l0 [x0][y0]] == 0 && chroma_weight_l1_flag[ref_idx_l1 [x0][y0]] == 0 && cbWidth * cbHeight >= 256)	
gbi_idx [x0][y0]	ae(v)
}	
}	

FIG. 35

26/39

	Bộ mô tả
seq_parameter_set_rbsp() {	
// ...	
sps_temporal_mvp_enabled_flag	u(1)
if(sps_temporal_mvp_enabled_flag)	
sps_sbtmvp_enabled_flag	u(1)
sps_amvr_enabled_flag	u(1)
sps_bdof_enabled_flag	u(1)
sps_dmvr_enabled_flag	u(1)
sps_cclm_enabled_flag	u(1)
if(sps_cclm_enabled_flag && chroma_format_idc == 1)	
sps_cclm_colocated_chroma_flag	u(1)
sps_mts_enabled_flag	u(1)
if(sps_mts_enabled_flag) {	
sps_explicit_mts_intra_enabled_flag	u(1)
sps_explicit_mts_inter_enabled_flag	u(1)
}	
sps_sbt_enabled_flag	u(1)
if(sps_sbt_enabled_flag)	
sps_sbt_max_size_64_flag	u(1)
sps_affine_enabled_flag	u(1)
if(sps_affine_enabled_flag) {	
sps_affine_type_flag	u(1)
sps_affine_amvr_enabled_flag	u(1)
}	
sps_gbi_enabled_flag	u(1)
sps_ibc_enabled_flag	u(1)
sps_ciip_enabled_flag	u(1)
sps_fpel_mmvd_enabled_flag	u(1)
sps_triangle_enabled_flag	u(1)
sps_lmcs_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)
}	
}	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

FIG. 36

27/39

(a)

seq_parameter set rbsp() {	Bộ mô tả
// ...	
sps amvr enabled flag	u(1)
if(sps amvr enabled flag)	
sps affine amvr enabled flag	u(1)
// ...	
sps affine enabled flag	u(1)
if(sps affine enabled flag)	
sps affine type flag	u(1)
// ...	
}	

(b)

	seq_parameter set rbsp() {	Bộ mô tả
	// ...	
	sps amvr enabled flag	u(1)
	// ...	
3701	sps affine enabled flag	u(1)
3702	if(sps affine enabled flag) {	
	sps affine type flag	u(1)
3703	if(sps amvr enabled flag)	
	sps affine amvr enabled flag	u(1)
	}	
	// ...	
	}	

FIG. 37

} else if(treeType != DUAL_TREE_CHROMA) { /* MODE INTER or MODE IBC */	
if(cu_skip_flag[x0][y0] == 0)	
merge_flag [x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
merge_data(x0, y0, cbWidth, cbHeight)	
} else if(CuPredMode[x0][y0] == MODE_IBC) { // ...	
} else {	
if(slice_type == B)	
inter_pred_idc [x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag [x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag [x0][y0]	ae(v)
}	
// sym_mvd_flag[x0][y0], ref_idx_l0[x0][y0], mvd_coding, mvp_l0_flag[x0][y0], ref_idx_l1[x0][y0], mvp_l1_flag[x0][y0]	
if((sps_amvr_enabled_flag && inter_affine_flag == 0 && (((MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0) && inter_pred_idc != PRED_L1) (MvdL1[x0][y0][0] != 0 MvdL1[x0][y0][1] != 0) && inter_pred_idc != PRED_L0)) (sps_affine_amvr_enabled_flag && inter_affine_flag[x0][y0] == 1 && (((MvdCpL0[x0][y0][0][0] != 0 MvdCpL0[x0][y0][0][1] != 0) && inter_pred_idc != PRED_L1) (MvdCpL1[x0][y0][0][0] != 0 MvdCpL1[x0][y0][0][1] != 0) && inter_pred_idc != PRED_L0) (MvdCpL0[x0][y0][1][0] != 0 MvdCpL0[x0][y0][1][1] != 0) && inter_pred_idc != PRED_L1) (MvdCpL1[x0][y0][1][0] != 0 MvdCpL1[x0][y0][1][1] != 0) && inter_pred_idc != PRED_L0) (MvdCpL0[x0][y0][2][0] != 0 MvdCpL0[x0][y0][2][1] != 0) && (inter_pred_idc != PRED_L1 && MotionModelIdc[x0][y0] == 2)) (MvdCpL1[x0][y0][2][0] != 0 MvdCpL1[x0][y0][2][1] != 0) && (inter_pred_idc != PRED_L0 && MotionModelIdc[x0][y0] == 2))) {	
amvr_flag [x0][y0]	ae(v)
if(amvr_flag[x0][y0])	
amvr_precision_flag [x0][y0]	ae(v)
}	
// ...	
}	

FIG. 38

28/39

abs_mvd_greater0_flag[compIdx] xác định xem giá trị tuyệt đối của chênh lệch thành phần vector chuyển động có lớn hơn 0 hay không.

Khi **abs_mvd_greater0_flag[compIdx]** không có mặt, thì giá trị này được suy ra là bằng 0.

abs_mvd_greater1_flag[compIdx] xác định xem giá trị tuyệt đối của chênh lệch thành phần vector chuyển động có lớn hơn 1 hay không.

Khi **abs_mvd_greater1_flag[compIdx]** không có mặt, thì giá trị này được suy ra là bằng 0.

abs_mvd_minus2[compIdx] cộng 2 xác định giá trị tuyệt đối của chênh lệch thành phần vector chuyển động.

Khi **abs_mvd_minus2[compIdx]** không có mặt, thì giá trị này được suy ra là bằng -1.

mvd_sign_flag[compIdx] xác định dấu của của chênh lệch thành phần vector chuyển động như sau:

- Nếu **mvd_sign_flag[compIdx]** bằng 0, thì chênh lệch thành phần vector chuyển động có giá trị dương.
- Theo cách khác (**mvd_sign_flag[compIdx]** bằng 1), thì chênh lệch thành phần vector chuyển động có giá trị âm.

Khi **mvd_sign_flag[compIdx]** không có mặt, thì giá trị này được suy ra là bằng 0.

Chênh lệch vector chuyển động **lMvd[compIdx]** đối với **compIdx = 0..1** được suy ra như sau:

$$lMvd[compIdx] = abs_mvd_greater0_flag[compIdx] * (abs_mvd_minus2[compIdx] + 2) * (1 - 2 * mvd_sign_flag[compIdx]) \quad (7-121)$$

Giá trị của **lMvd[compIdx]** nên nằm trong khoảng từ -2^{15} đến $2^{15} - 1$.

Tùy thuộc vào giá trị của **MotionModelIdx[x][y]**, các chênh lệch vector chuyển động được suy ra như sau:

- Nếu **MotionModelIdx[x][y]** bằng 0, biến **MvdLX[x0][y0][compIdx]**, với **X** bằng 0 hoặc 1, xác định chênh lệch giữa thành phần vector danh mục **X** cần được sử dụng và dự đoán của nó. Các chỉ số mảng **x0, y0** xác định vị trí (**x0, y0**) của mẫu độ sáng trên cùng bên trái của khối mã hóa được xem xét so với mẫu độ sáng trên cùng bên trái của ảnh. Chênh lệch thành phần vector chuyển động ngang được chỉ định **compIdx = 0** và thành phần vector chuyển động dọc được chỉ định **compIdx = 1**.

- Nếu **refList** bằng 0, thì **MvdL0[x0][y0][compIdx]** được thiết lập thành **lMvd[compIdx]** đối với **compIdx = 0..1**.

- Theo cách khác (**refList** bằng 1), thì **MvdL1[x0][y0][compIdx]** được thiết lập thành **lMvd[compIdx]** đối với **compIdx = 0..1**.

- Theo cách khác (**MotionModelIdx[x][y]** không bằng 0), biến **MvdCpLX[x0][y0][cpIdx][compIdx]**, với **X** là 0 hoặc 1, xác định chênh lệch giữa thành phần vector danh mục **X** cần được sử dụng và dự đoán của nó. Các chỉ số mảng **x0, y0** xác định vị trí (**x0, y0**) của mẫu độ sáng trên cùng bên trái của khối mã hóa được xem xét so với mẫu độ sáng trên cùng bên trái của ảnh, chỉ số mảng **cpIdx** xác định chỉ số điểm điều khiển. Chênh lệch thành phần vector chuyển động ngang được chỉ định **compIdx = 0** và thành phần vector chuyển động dọc được chỉ định **compIdx = 1**.

- Nếu **refList** bằng 0, thì **MvdCpL0[x0][y0][cpIdx][compIdx]** được thiết lập thành **lMvd[compIdx]** đối với **compIdx = 0..1**.

- Theo cách khác (**refList** bằng 1), thì **MvdCpL1[x0][y0][cpIdx][compIdx]** được thiết lập thành **lMvd[compIdx]** đối với **compIdx = 0..1**.

29/39

} else {	
if(slice type == B)	
inter_pred_idc[x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag[x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
if(sps_mvmd_enabled_flag && inter_pred_idc[x0][y0] == PRED_BI && !inter_affine_flag[x0][y0] && RefIdxSymL0 > -1 && RefIdxSymL1 > -1)	
sym_mvmd_flag[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvmd_flag[x0][y0])	
ref_idx_l0[x0][y0]	ae(v)
mvmd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvmd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvmd_coding(x0, y0, 0, 2)	
mvmd_flag[x0][y0]	ae(v)
if(MotionModelIdc[x0][y0] == 1) {	
MvdCpL0[x0][y0][2][0] = 0	
MvdCpL0[x0][y0][2][1] = 0	
}	
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
MvdCpL0[x0][y0][0][0] = 0	
MvdCpL0[x0][y0][0][1] = 0	
MvdCpL0[x0][y0][1][0] = 0	
MvdCpL0[x0][y0][1][1] = 0	
MvdCpL0[x0][y0][2][0] = 0	
MvdCpL0[x0][y0][2][1] = 0	
}	

if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvmd_flag[x0][y0])	
ref_idx_l1[x0][y0]	ae(v)
if(mvmd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
} else {	
if(sym_mvmd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = -MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = -MvdL0[x0][y0][1]	
} else {	
mvmd_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvmd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvmd_coding(x0, y0, 1, 2)	
mvmd_flag[x0][y0]	ae(v)
if(MotionModelIdc[x0][y0] == 1) {	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
}	
} else {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
}	
if((sps_mvmd_enabled_flag && inter_affine_flag[x0][y0] == 0 && (MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0 MvdL1[x0][y0][0] != 0 MvdL1[x0][y0][1] != 0)) (sps_affine_mvmd_enabled_flag && inter_affine_flag[x0][y0] == 1 && (MvdCpL0[x0][y0][0][0] != 0 MvdCpL0[x0][y0][0][1] != 0 MvdCpL1[x0][y0][0][0] != 0 MvdCpL1[x0][y0][0][1] != 0 MvdCpL0[x0][y0][1][0] != 0 MvdCpL0[x0][y0][1][1] != 0 MvdCpL1[x0][y0][1][0] != 0 MvdCpL1[x0][y0][1][1] != 0 MvdCpL0[x0][y0][2][0] != 0 MvdCpL0[x0][y0][2][1] != 0 MvdCpL1[x0][y0][2][0] != 0 MvdCpL1[x0][y0][2][1] != 0))) {	
amvr_flag[x0][y0]	ae(v)
if(amvr_flag[x0][y0])	
amvr_precision_flag[x0][y0]	ae(v)
}	

FIG. 40

30/39

} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
if(cu_skip_flag[x0][y0] == 0)	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
merge_data(x0, y0, cbWidth, cbHeight)	
} else if(CuPredMode[x0][y0] == MODE_IBC) { // ...	
} else {	
if(slice_type == B)	
inter_pred_idc[x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag[x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
// sym_mv flag[x0][y0], ref_idx_l0[x0][y0], mvd coding, mvp_l0 flag[x0][y0], ref_idx_l1[x0][y0], mvp_l1 flag[x0][y0]	
if((sps_mv_enabled_flag && inter_affine_flag == 0 &&	
((MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0) && inter_pred_idc != PRED_L1)	
((MvdL1[x0][y0][0] != 0 MvdL1[x0][y0][1] != 0) && inter_pred_idc != PRED_L0 && !(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_	
BD)))	
(sps_affine_mv_enabled_flag && inter_affine_flag[x0][y0] == 1 &&	
((MvdCpL0[x0][y0][0][0] != 0 MvdCpL0[x0][y0][0][1] != 0) && inter_pred_idc != PRED_L1)	
(MvdCpL1[x0][y0][0][0] != 0 MvdCpL1[x0][y0][0][1] != 0) && inter_pred_idc != PRED_L0 && !(mvd_l1_zero_flag && inter_pred_idc[x0]	
[y0] == PRED_BD)	
((MvdCpL0[x0][y0][1][0] != 0 MvdCpL0[x0][y0][1][1] != 0) && inter_pred_idc != PRED_L1)	
(MvdCpL1[x0][y0][1][0] != 0 MvdCpL1[x0][y0][1][1] != 0) && inter_pred_idc != PRED_L0 && !(mvd_l1_zero_flag && inter_pred_idc[x0]	
[y0] == PRED_BD)	
(MvdCpL0[x0][y0][2][0] != 0 MvdCpL0[x0][y0][2][1] != 0) && (inter_pred_idc != PRED_L1 && MotionModelIdc[x0][y0] == 2))	
(MvdCpL1[x0][y0][2][0] != 0 MvdCpL1[x0][y0][2][1] != 0) && (inter_pred_idc != PRED_L0 && MotionModelIdc[x0][y0] == 2 && !(mvd	
l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BD))) {	
amvr_flag[x0][y0]	ae(v)
if(amvr_flag[x0][y0])	
amvr_precision_flag[x0][y0]	ae(v)
}	
// ...	
}	
}	

FIG. 41

31/39

if(cu skip flag[x0][y0] == 0)	
general merge flag [x0][y0]	ae(v)
if(general merge flag[x0][y0]) {	
merge_data(x0, y0, cbWidth, cbHeight)	
} else if(CuPredMode[x0][y0] == MODE_IBC) {	
mvd_coding(x0, y0, 0, 0)	
mvp 10 flag [x0][y0]	ae(v)
if(sps_amvr_enabled_flag &&	
(MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0)) {	
amvr precision flag [x0][y0]	ae(v)
}	
} else {	
if(slice_type == B)	
inter_pred_idc [x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine flag [x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu affine type flag [x0][y0]	ae(v)
}	
if(sps_smvd_enabled_flag && inter_pred_idc[x0][y0] == PRED_BI &&	
!inter_affine_flag[x0][y0] && RefIdxSymL0 > -1 && RefIdxSymL1 >	
-1)	
sym mvd flag [x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx l0 [x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp 10 flag [x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	

if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx l1 [x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
}	
if(sym_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = -MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = -MvdL0[x0][y0][1]	
}	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
mvp 11 flag [x0][y0]	ae(v)
}	
} else {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
}	

FIG. 42

32/39

4301

if(cu skip flag[x0][y0] == 0)	
general merge flag[x0][y0]	ae(v)
if(general merge flag[x0][y0]) {	
merge_data(x0, y0, cbWidth, cbHeight)	
} else if(CuPredMode[x0][y0] == MODE_IBC) {	
mvd_coding(x0, y0, 0, 0)	
mvp_l0_flag[x0][y0]	ae(v)
if(sps_amvr_enabled_flag && (MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0)) {	
amvr_precision_flag[x0][y0]	ae(v)
}	
} else {	
if(slice_type == B)	
inter_pred_idc[x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag[x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
if(sps_symvd_enabled_flag && inter_pred_idc[x0][y0] == PRED_BI && inter_affine_flag[x0][y0] && RefIdxSymL0 > -1 && RefIdxSymL1 > -1)	
sym_mvd_flag[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l0[x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp_l0_flag[x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	

4302

if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l1[x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
} else {	
if(sym_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = -MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = -MvdL0[x0][y0][1]	
} else	
mvd_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
}	
mvp_l1_flag[x0][y0]	ae(v)
} else {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
}	

4303

FIG. 43

33/39

if(cu skip flag[x0][y0] == 0)	
general merge flag [x0][y0]	ae(v)
if(general merge flag[x0][y0]) {	
merge_data(x0, y0, cbWidth, cbHeight)	
} else if(CuPredMode[x0][y0] == MODE_IBC) {	
mvd_coding(x0, y0, 0, 0)	
mvp l0 flag [x0][y0]	ae(v)
if(sps_amvr_enabled_flag &&	
(MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0)) {	
amvr precision flag [x0][y0]	ae(v)
}	
} else {	
if(slice_type == B)	
inter_pred_idc [x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine flag [x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu affine type flag [x0][y0]	ae(v)
}	
if(sps_smvd_enabled_flag && inter_pred_idc[x0][y0] == PRED_BI &&	
(inter_affine_flag[x0][y0] && RefIdxSvmL0 > -1 && RefIdxSvmL1 > -1)	
svm mvd flag [x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !svm_mvd_flag[x0][y0])	
ref_idx l0 [x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp l0 flag [x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	

if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !svm_mvd_flag[x0][y0])	
ref_idx l1 [x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
mvp l1 flag [x0][y0]	ae(v)
} else {	
if(svm_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = -MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = -MvdL0[x0][y0][1]	
} else	
mvd_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
mvp l1 flag [x0][y0]	ae(v)
}	
} else {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
}	

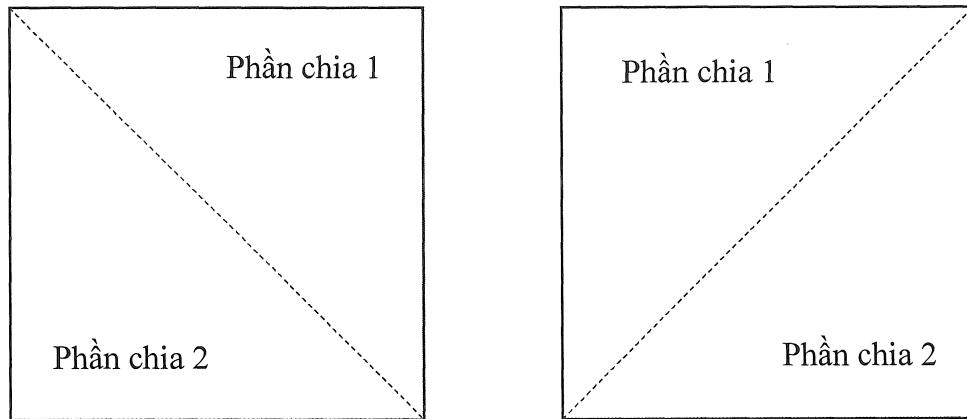
FIG. 44

34/39

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Bộ mô tả
{ /* MODE INTER */	
if(cu_skip_flag[x0][y0]) {	
if(sps_affine_enabled_flag && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 0 && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1 && MaxNumSubblockMergeCand > 1)	
merge_subblock_idx[x0][y0]	ae(v)
} else {	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
if(sps_affine_enabled_flag && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 0 && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1 && MaxNumSubblockMergeCand > 1)	
merge_subblock_idx[x0][y0]	ae(v)
} else {	
if(slice_type == B)	
inter_pred_idc[x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag[x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(num_ref_idx_l0_active_minus1 > 0)	
ref_idx_l0[x0][y0]	ae(v)
mvld_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvld_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvld_coding(x0, y0, 0, 2)	
mvp_l0_flag[x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(num_ref_idx_l1_active_minus1 > 0)	
ref_idx_l1[x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
} else {	
mvld_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvld_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvld_coding(x0, y0, 1, 2)	
mvp_l1_flag[x0][y0]	ae(v)
} else {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
}	
if(sps_amvr_enabled_flag && inter_affine_flag == 0 &&	
(MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0	
MvdL1[x0][y0][0] != 0 MvdL1[x0][y0][1] != 0)	
amvr_mode[x0][y0]	ae(v)
}	
}	
if(CuPredMode[x0][y0] != MODE_INTRA && cu_skip_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
}	

FIG. 45

35/39

*FIG. 46*

	Bộ mô tả
merge_data(x0, y0, cbWidth, cbHeight) {	
if (CuPredMode[x0][y0] == MODE_IBC) {	
if(MaxNumMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
} else {	
if(sps_mmvd_enabled_flag cbWidth * cbHeight != 32)	
regular_merge_flag [x0][y0]	ae(v)
if (regular_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
} else {	
if(sps_mmvd_enabled_flag && cbWidth * cbHeight != 32)	
mmvd_merge_flag [x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag [x0][y0]	ae(v)
mmvd_distance_idx [x0][y0]	ae(v)
mmvd_direction_idx [x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag [x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx [x0][y0]	ae(v)
} else {	
if(sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 &&	
(cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	
ciip_flag [x0][y0]	ae(v)
if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
}	
if(MergeTriangleFlag[x0][y0]) {	
merge_triangle_split_dir [x0][y0]	ae(v)
merge_triangle_idx0 [x0][y0]	ae(v)
merge_triangle_idx1 [x0][y0]	ae(v)
}	
}	
}	
}	

37/39

slice header() {	Bộ mô tả
// ...	
if (slice type != I) {	
if(sps temporal mvp enabled flag)	
slice temporal mvp enabled flag	u(1)
if(slice type == B)	
mvd l1 zero flag	u(1)
if(cabac init present flag)	
cabac init flag	u(1)
if(slice temporal mvp enabled flag) {	
if(slice type == B)	
collocated from l0 flag	u(1)
}	
if((weighted_pred_flag && slice_type == P)	
(weighted bipred flag && slice type == B))	
pred weight table()	
six minus max num merge cand	ue(v)
if(sps affine enabled flag)	
five minus max num subblock merge cand	ue(v)
if(sps fpel mmvd enabled flag)	
slice fpel mmvd enabled flag	u(1)
if(sps triangle enabled flag && MaxNumMergeCand >= 2)	
max num merge cand minus max num triangle cand	ue(v)
} else if (sps ibc enabled flag)	
six minus max num merge cand	ue(v)
// ...	
}	

FIG. 48

max num merge cand minus max num triangle cand xác định số lượng ứng viên chế độ hợp nhất tam giác tối đa trong lát được trừ vào MaxNumMergeCand. Số lượng ứng viên chế độ hợp nhất tam giác tối đa, MaxNumTriangleMergeCand được suy ra như sau:

$$\text{MaxNumTriangleMergeCand} = \text{MaxNumMergeCand} - \text{max_num_merge_cand_minus_max_num_triangle_cand} \quad (7-59)$$

Khi **max_num_merge_cand_minus_max_num_triangle_cand** có mặt, thì giá trị của MaxNumTriangleMergeCand nên nằm trong khoảng từ 2 đến MaxNumMergeCand. Khi **max_num_merge_cand_minus_max_num_triangle_cand** không có mặt, thì MaxNumTriangleMergeCand được thiết lập bằng 0. Khi MaxNumTriangleMergeCand bằng 0, thì chế độ hợp nhất tam giác không được cho phép đối với lát hiện thời.

FIG. 49

5001	slice header() {	Bộ mô tả
	// ...	
	if(sps triangle enabled flag && MaxNumMergeCand > 2)	
	max num merge cand minus max num triangle cand	ue(v)
	// ...	
	}	

FIG. 50

38/39

max_num_merge_cand_minus_max_num_triangle_cand xác định số lượng ứng viên chế độ hợp nhất tam giác tối đa được hỗ trợ trong lát được trừ vào **MaxNumMergeCand**. Số lượng ứng viên chế độ hợp nhất tam giác tối đa, **MaxNumTriangleMergeCand** được suy ra như sau:

$$5101 \quad \text{MaxNumTriangleMergeCand} = \text{MaxNumMergeCand} - \text{max_num_merge_cand_minus_max_num_triangle_cand} \quad (7-60)$$

Khi **max_num_merge_cand_minus_max_num_triangle_cand** có mặt, thì giá trị của **MaxNumTriangleMergeCand** nên nằm trong khoảng từ 2 đến **MaxNumMergeCand**. Khi **max_num_merge_cand_minus_max_num_triangle_cand** không có mặt,

5102 Nếu **sps_triangle_enabled_flag** bằng 1 và **MaxNumMergeCand** bằng 2, thì **MaxNumTriangleMergeCand** được thiết lập bằng 2.

5103 Theo cách khác (**sps_triangle_enabled_flag** bằng 0 hoặc **MaxNumMergeCand** không bằng 2), thì **MaxNumTriangleMergeCand** được thiết lập bằng 0.

Khi **MaxNumTriangleMergeCand** bằng 0, thì chế độ hợp nhất tam giác không được cho phép đối với lát hiện thời.

AG. 51

if(MergeTriangleFlag[x0][y0]) {	
merge_triangle_split_dir [x0][y0]	ae(v)
if(MaxNumTriangleMergeCand > 2) {	
merge_triangle_idx0 [x0][y0]	ae(v)
merge_triangle_idx1 [x0][y0]	ae(v)
} else {	
merge_triangle_idx_indicator [x0][y0]	ae(v)
}	
}	

AG. 52

merge_triangle_idx0[x0][y0] xác định chỉ số ứng viên hợp nhất thứ nhất của danh mục ứng viên bù chuyển động dựa trên hình tam giác trong đó x0, y0 xác định vị trí (x0, y0) của mẫu độ sáng trên cùng bên trái của khối mã hóa được xem xét so với mẫu độ sáng trên cùng bên trái của ảnh.

Khi **merge_triangle_idx0[x0][y0]** không có mặt, thì giá trị này được suy ra như sau:

Nếu **MergeTriangleFlag[x0][y0]** bằng 1, **MaxNumTriangleMergeCand** bằng 2 và **merge_triangle_idx_indicator[x0][y0]** bằng 1, thì giá trị này được suy ra là bằng 1.

Theo cách khác, giá trị này được suy ra là bằng 0.

merge_triangle_idx1[x0][y0] xác định chỉ số ứng viên hợp nhất thứ hai của danh mục ứng viên bù chuyển động dựa trên hình tam giác trong đó x0, y0 xác định vị trí (x0, y0) của mẫu độ sáng trên cùng bên trái của khối mã hóa được xem xét so với mẫu độ sáng trên cùng bên trái của ảnh.

Khi **merge_triangle_idx1[x0][y0]** không có mặt, thì giá trị này được suy ra là bằng 0.

AG. 53

39/39

if(MergeTriangleFlag[x0][y0]) {	
merge_triangle_split_dir[x0][y0]	ae(v)
merge_triangle_idx0[x0][y0]	ae(v)
merge_triangle_idx1[x0][y0]	ae(v)
merge_triangle_bigger[x0][y0]	ae(v)
}	

Các biến m và n, lần lượt là chỉ số hợp nhất cho phần chia tam giác 0 và 1, được suy ra nhờ sử dụng merge_triangle_idx0[xCb][yCb] và merge_triangle_idx1[xCb][yCb] như sau:

Nếu merge_triangle_bigger[xCb][yCb] bằng 0, thì
m = merge_triangle_idx0[xCb][yCb]
n = merge_triangle_idx1[xCb][yCb] + m + 1

Theo cách khác,

n = merge_triangle_idx0[xCb][yCb]
m = merge_triangle_idx1[xCb][yCb] + n + 1

FIG. 54

if(MergeTriangleFlag[x0][y0]) {	
merge_triangle_split_dir[x0][y0]	ae(v)
merge_triangle_idx0[x0][y0]	ae(v)
if(merge_triangle_idx0[x0][y0] > 0)	
merge_triangle_idx1[x0][y0]	ae(v)
merge_triangle_bigger[x0][y0]	ae(v)
}	

Các biến m và n, lần lượt là chỉ số hợp nhất cho phần chia tam giác 0 và 1, được suy ra nhờ sử dụng merge_triangle_idx0[xCb][yCb] và merge_triangle_idx1[xCb][yCb] như sau:

Nếu merge_triangle_bigger[xCb][yCb] bằng 0, thì
m = merge_triangle_idx0[xCb][yCb] + 1
n = merge_triangle_idx1[xCb][yCb]

Theo cách khác,

n = merge_triangle_idx0[xCb][yCb] + 1
m = merge_triangle_idx1[xCb][yCb]

FIG. 55