



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

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

(11)



1-0052208

(51)^{2020.01} H04N 19/573; H04N 19/109; H04N 19/132; H04N 19/70; H04N 19/176; H04N 19/527; H04N 19/105; H04N 19/137 (13) B

(21) 1-2022-00033

(22) 10/06/2020

(86) PCT/KR2020/007523 10/06/2020

(87) WO/2020/251259 17/12/2020

(30) 62/861,989 14/06/2019 US

(45) 27/10/2025 451

(43) 25/03/2022 408A

(73) LG ELECTRONICS INC. (KR)

128, Yeoui-daero, Yeongdeungpo-gu Seoul 07336, Korea

(72) PARK, Naeri (KR); NAM, Junghak (KR); JANG, Hyeongmoon (KR).

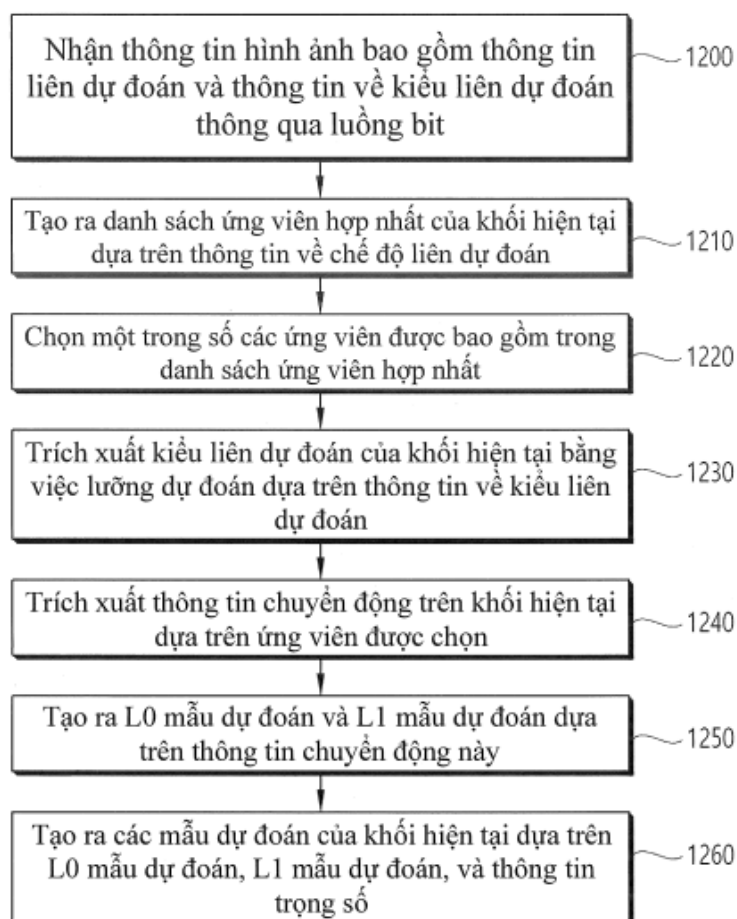
(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP GIẢI MÃ ẢNH, PHƯƠNG PHÁP MÃ HOÁ ẢNH, VÀ PHƯƠNG TIỆN LƯU TRỮ KỸ THUẬT SỐ PHI NHẤT THỜI ĐỌC ĐƯỢC BẰNG MÁY TÍNH

(21) 1-2022-00033

(57) Sáng chế đề xuất phương pháp giải mã ảnh, phương pháp mã hoá ảnh, phương tiện lưu trữ kỹ thuật số phi nhất thời đọc được bằng máy tính, và phương pháp truyền dữ liệu đối với hình ảnh. Theo sáng chế, khi kiểu liên dự đoán của khối hiện tại chỉ thị việc lưỡng dự đoán, thì thông tin chỉ số trọng số dành cho ứng viên trong danh sách ứng viên hợp nhất hoặc danh sách ứng viên hợp nhất khối con có thể được trích xuất, và hiệu quả lập mã có thể được nâng lên.

Fig.12



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp và thiết bị giải mã ảnh để trích xuất thông tin chỉ số trọng số trên việc lưỡng dự đoán.

Tình trạng kỹ thuật của sáng chế

Gần đây, nhu cầu về ảnh/video độ phân giải cao, chất lượng cao, chẳng hạn ảnh/video có độ phân giải siêu cao (Ultra High Definition - UHD) 4K, 8K hoặc hơn nữa, đang ngày càng tăng trong các lĩnh vực khác nhau. Khi độ phân giải hoặc chất lượng ảnh/video trở nên cao hơn, thì lượng thông tin hoặc bit được truyền nhiều hơn tương đối so với dữ liệu ảnh/video thông thường. Do đó, nếu dữ liệu ảnh/video được truyền qua phương tiện chẳng hạn đường băng rộng có dây/không dây hiện có hoặc được lưu giữ trong phương tiện lưu trữ kế thừa, thì chi phí để truyền và lưu giữ dễ bị tăng lên.

Ngoài ra, sự quan tâm và nhu cầu đối với các nội dung thực tế ảo (Virtual Reality - VR) và thực tế nhân tạo (Artificial Reality - AR), và môi trường hoà mình, chẳng hạn ảnh nổi, là đang tăng lên; và việc phát quảng bá các hình ảnh/video có các đặc điểm hình ảnh/video khác với của hình ảnh/video thực tế, chẳng hạn các hình ảnh/video của trò chơi, cũng đang tăng lên.

Do đó, cần phải có kỹ thuật nén ảnh/video hiệu quả cao để nén và truyền, lưu giữ, hoặc phát một cách hiệu quả các hình ảnh/video có độ phân giải cao, chất lượng cao mà có các đặc điểm khác nhau như đã mô tả trên đây.

Bản chất kỹ thuật của sáng chế

Sáng chế đề xuất phương pháp và thiết bị để tăng hiệu quả lập mã ảnh.

Sáng chế cũng đề xuất phương pháp và thiết bị để trích xuất thông tin chỉ số trọng số trên việc lưỡng dự đoán trong quá trình liên dự đoán.

Sáng chế cũng đề xuất phương pháp và thiết bị để trích xuất thông tin chỉ số trọng số trên ứng viên trong danh sách ứng viên hợp nhất afin trong quá trình lưỡng dự đoán.

Theo một phương án, sáng chế đề xuất phương pháp giải mã ảnh được thực hiện bởi thiết bị giải mã. Phương pháp này bao gồm các bước nhận thông tin hình ảnh bao gồm thông tin về chế độ liên dự đoán và thông tin về kiểu liên dự đoán thông qua luồng bit; tạo ra danh sách ứng viên hợp nhất của khối hiện tại dựa trên thông tin về chế độ liên dự đoán; chọn một ứng viên từ trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất này; trích xuất kiểu liên dự đoán của khối hiện tại là lưỡng dự đoán dựa trên thông tin về kiểu liên dự đoán; trích xuất thông tin chuyển động trên khối hiện tại dựa trên ứng viên được chọn; tạo ra L0 mẫu dự đoán và L1 mẫu dự đoán của khối hiện tại dựa trên thông tin chuyển động này; và tạo ra các mẫu dự đoán của khối hiện tại dựa trên L0 mẫu dự đoán, L1 mẫu dự đoán này, và thông tin trọng số, trong đó thông tin trọng số này được trích xuất dựa trên thông tin chỉ số trọng số trên ứng viên được chọn, trong đó các ứng viên bao gồm ứng viên hợp nhất afin, và ứng viên hợp nhất afin này bao gồm các vectơ chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV), khi ứng viên hợp nhất afin bao gồm CPMV dành cho điểm kiểm soát 0 (CP0) được đặt tại đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất afin là được trích xuất dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của CP0, và khi ứng viên hợp nhất afin không bao gồm CPMV dành cho CP0 được đặt tại đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất afin là được trích xuất dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của điểm kiểm soát 1 (CP1) được đặt tại đỉnh phải của khối hiện tại.

Theo phương án khác, sáng chế đề xuất phương pháp mã hoá ảnh được thực

hiện bởi thiết bị mã hoá. Phương pháp này bao gồm các bước xác định chế độ liên dự đoán của khối hiện tại và tạo ra thông tin về chế độ liên dự đoán chỉ thị chế độ liên dự đoán này; tạo ra danh sách ứng viên hợp nhất của khối hiện tại dựa trên thông tin về chế độ liên dự đoán; tạo ra thông tin chọn chỉ thị một trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất này; tạo ra thông tin về kiểu liên dự đoán mà chỉ thị kiểu liên dự đoán của khối hiện tại là lưỡng dự đoán; và mã hoá thông tin hình ảnh bao gồm thông tin về chế độ liên dự đoán, thông tin chọn, và thông tin về kiểu liên dự đoán này, trong đó các ứng viên bao gồm ứng viên hợp nhất afin, và ứng viên hợp nhất afin này bao gồm các vector chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV), khi ứng viên hợp nhất afin bao gồm CPMV dành cho điểm kiểm soát 0 (CP0) được đặt tại đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất afin là được chỉ thị dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của CP0, và khi ứng viên hợp nhất afin không bao gồm CPMV dành cho CP0 được đặt tại đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất afin là được chỉ thị dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của điểm kiểm soát 1 (CP1) được đặt tại đỉnh phải của khối hiện tại.

Theo phương án khác nữa, sáng chế đề xuất phương tiện lưu trữ đọc được bằng máy tính có lưu giữ thông tin được mã hoá để khiến thiết bị giải mã ảnh thực hiện phương pháp giải mã ảnh. Phương pháp giải mã ảnh này bao gồm các bước nhận thông tin hình ảnh bao gồm thông tin về chế độ liên dự đoán và thông tin về kiểu liên dự đoán thông qua luồng bit; tạo ra danh sách ứng viên hợp nhất của khối hiện tại dựa trên thông tin về chế độ liên dự đoán; chọn một ứng viên từ trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất này; trích xuất kiểu liên dự đoán của khối hiện tại là lưỡng dự đoán dựa trên thông tin về kiểu liên dự đoán; trích xuất thông tin chuyển động trên khối hiện tại dựa trên ứng viên được chọn; tạo ra L0 mẫu dự đoán và L1 mẫu dự đoán của khối hiện tại dựa trên thông tin chuyển động này; và tạo ra các mẫu dự đoán của khối hiện tại dựa trên L0 mẫu dự đoán, L1 mẫu dự đoán này, và thông tin trọng số,

trong đó thông tin trọng số này được trích xuất dựa trên thông tin chỉ số trọng số trên ứng viên được chọn, trong đó các ứng viên bao gồm ứng viên hợp nhất affin, và ứng viên hợp nhất affin này bao gồm các vector chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV), khi ứng viên hợp nhất affin bao gồm CPMV dành cho điểm kiểm soát 0 (CP0) được đặt tại đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất affin là được trích xuất dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của CP0, và khi ứng viên hợp nhất affin không bao gồm CPMV dành cho CP0 được đặt tại đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất affin là được trích xuất dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của điểm kiểm soát 1 (CP1) được đặt tại đỉnh phải của khối hiện tại.

Theo sáng chế, có thể tăng hiệu quả nén ảnh/video tổng thể.

Theo sáng chế, có thể xây dựng một cách hiệu quả các ứng viên vector chuyển động trong quá trình liên dự đoán.

Theo sáng chế, có thể thực hiện một cách hiệu quả việc lưỡng dự đoán dựa trên trọng số.

Mô tả vắn tắt các hình vẽ

Fig.1 là hình vẽ thể hiện sơ đồ theo một ví dụ của hệ thống lập mã video/ảnh mà các phương án của sáng chế có thể được áp dụng vào đó.

Fig.2 là hình vẽ thể hiện sơ đồ cấu hình của thiết bị mã hoá video/ảnh mà các phương án của sáng chế có thể được áp dụng vào đó.

Fig.3 là hình vẽ thể hiện sơ đồ cấu hình của thiết bị giải mã video/ảnh mà các phương án của sáng chế có thể được áp dụng vào đó.

Fig.4 là hình vẽ thể hiện sơ đồ mô tả chế độ hợp nhất trong quá trình liên dự đoán.

Fig.5A và Fig.5B là các hình vẽ thể hiện sơ đồ ví dụ CPMV dành cho việc dự

đoán chuyển động afin.

Fig.6 là hình vẽ thể hiện sơ đồ ví dụ trường hợp mà trong đó MVF afin được xác định theo đơn vị là các khối con.

Fig.7 là hình vẽ thể hiện sơ đồ mô tả chế độ hợp nhất afin trong quá trình liên dự đoán.

Fig.8 là hình vẽ thể hiện sơ đồ mô tả các vị trí của các ứng viên trong chế độ hợp nhất afin.

Fig.9 là hình vẽ thể hiện sơ đồ mô tả SbTMVP trong quá trình liên dự đoán.

Fig.10 và Fig.11 là các hình vẽ thể hiện sơ đồ ví dụ của phương pháp giải mã video/ảnh và các thành phần liên quan theo (các) phương án thực hiện của sáng chế.

Fig.12 và Fig.13 là các hình vẽ thể hiện sơ đồ ví dụ của phương pháp mã hoá ảnh/video và các thành phần liên quan theo (các) phương án thực hiện của sáng chế.

Fig.14 là hình vẽ thể hiện sơ đồ ví dụ của hệ thống tạo luồng nội dung mà các phương án được bộc lộ trong bản mô tả này có thể được áp dụng vào đó.

Mô tả các phương án ví dụ

Sáng chế có thể được cải biến theo những cách khác nhau và có một số phương án ví dụ. Do đó, các phương án ví dụ cụ thể của sáng chế sẽ được thể hiện trên các hình vẽ kèm theo và được mô tả chi tiết. Tuy nhiên, điều này không nhằm giới hạn sáng chế ở các phương án cụ thể. Các thuật ngữ được sử dụng trong bản mô tả này là được dùng chỉ để mô tả các phương án ví dụ cụ thể thay vì giới hạn sáng chế. Các dạng số ít cũng được dự định là bao gồm dạng số nhiều, trừ khi ngữ cảnh chỉ rõ khác đi. Cần hiểu rằng các từ ngữ như “bao gồm”, “có”, v.v., mà được sử dụng trong bản mô tả này, là quy định sự có mặt của các dấu hiệu, các chữ số, các bước, các hoạt động, các thành phần, các bộ phận, hoặc tổ hợp của chúng mà được nêu trong bản mô tả này, chứ không loại trừ sự có mặt hoặc sự bổ sung của một hoặc nhiều dấu hiệu, chữ số, bước, hoạt động, thành phần, bộ phận khác, hoặc tổ hợp của chúng.

Trong khi đó, mỗi thành phần trên các hình vẽ được mô tả trong bản mô tả này là được thể hiện một cách độc lập để thuận tiện cho việc mô tả về các chức năng đặc trưng khác nhau, và không có nghĩa là mỗi thành phần được thực hiện dưới dạng phần cứng riêng biệt hoặc phần mềm riêng biệt. Ví dụ, hai hoặc nhiều thành phần trong số mỗi thành phần là có thể được kết hợp để tạo thành một thành phần, hoặc một thành phần có thể được chia thành nhiều thành phần. Các phương án mà trong đó mỗi thành phần là được hợp nhất và/hoặc được tách biệt là cũng được bao gồm trong phạm vi của sáng chế.

Trong bản mô tả này, thì “A hoặc B” có thể có nghĩa là “chỉ A”, “chỉ B” hoặc “cả A và B”. Nói cách khác, “A hoặc B” trong bản mô tả này có thể được diễn dịch là “A và/hoặc B”. Ví dụ, trong bản mô tả này, thì “A, B, hoặc C” có nghĩa là “chỉ A”, “chỉ B”, “chỉ C”, hoặc “bất kỳ trong số và sự kết hợp bất kỳ của A, B, và C”.

Dấu gạch chéo (/) hoặc dấu phẩy (comma) được sử dụng trong bản mô tả này có thể có nghĩa là “và/hoặc”. Ví dụ, “A/B” có thể có nghĩa là “A và/hoặc B”. Theo đó, “A/B” có thể có nghĩa là “chỉ A”, “chỉ B”, hoặc “cả A và B”. Ví dụ, “A, B, C” có thể có nghĩa là “A, B, hoặc C”.

Trong bản mô tả này, thì “ít nhất một trong số A và B” có thể có nghĩa là “chỉ A”, “chỉ B”, hoặc “cả A và B”. Ngoài ra, trong bản mô tả này, thì câu thể hiện “ít nhất một trong số A hoặc B” hoặc “ít nhất một trong số A và/hoặc B” có thể được diễn dịch giống như “ít nhất một trong số A và B”.

Ngoài ra, trong bản mô tả này, thì “ít nhất một trong số A, B và C” có thể có nghĩa là “chỉ A”, “chỉ B”, “chỉ C”, hoặc “sự kết hợp bất kỳ của A, B và C”. Ngoài ra, “ít nhất một trong số A, B hoặc C” hoặc “ít nhất một trong số A, B và/hoặc C” có thể có nghĩa là “ít nhất một trong số A, B và C”.

Ngoài ra, dấu ngoặc đơn được sử dụng trong bản mô tả này có thể có nghĩa là “ví dụ”. Cụ thể là, trong trường hợp thể hiện “dự đoán (nội dự đoán)”, thì có thể biểu thị rằng việc “nội dự đoán” được đề xuất làm một ví dụ của việc “dự đoán”. Nói cách

khác, thuật ngữ “dự đoán” trong bản mô tả này là không chỉ giới hạn ở việc “nội dự đoán”, và nó có thể biểu thị rằng việc “nội dự đoán” là được đề xuất làm một ví dụ của việc “dự đoán”. Ngoài ra, ngay cả trong trường hợp thể hiện “dự đoán (tức là, nội dự đoán)”, thì nó có thể biểu thị rằng việc “nội dự đoán” là được đề xuất làm một ví dụ của việc “dự đoán”.

Trong bản mô tả này, các dấu hiệu kỹ thuật được giải thích riêng lẻ trên một hình vẽ có thể được thực hiện riêng lẻ, hoặc có thể được thực hiện đồng thời.

Sau đây, các phương án của sáng chế sẽ được mô tả chi tiết dựa vào các hình vẽ kèm theo. Ngoài ra, các số chỉ dẫn giống nhau được dùng để chỉ các phần tử giống nhau trên khắp các hình vẽ, và những phần mô tả giống nhau của các phần tử giống nhau có thể được lược bỏ.

Fig.1 là hình vẽ thể hiện một ví dụ của hệ thống lập mã video/ảnh mà các phương án của sáng chế có thể được áp dụng vào đó.

Như được thể hiện trên Fig.1, hệ thống lập mã video/ảnh có thể bao gồm thiết bị thứ nhất (thiết bị nguồn) và thiết bị thứ hai (thiết bị nhận). Thiết bị nguồn có thể truyền thông tin hoặc dữ liệu video/ảnh được mã hoá đến thiết bị nhận qua phương tiện lưu trữ số hoặc mạng dưới dạng tệp tin hoặc việc tạo luồng.

Thiết bị nguồn có thể bao gồm nguồn video, thiết bị mã hoá, và bộ phát. Thiết bị nhận có thể bao gồm bộ thu, thiết bị giải mã, và bộ dựng hình. Thiết bị mã hoá này có thể được gọi là thiết bị mã hoá video/ảnh, và thiết bị giải mã này có thể được gọi là thiết bị giải mã video/ảnh. Bộ phát có thể được bao gồm trong thiết bị mã hoá. Bộ thu có thể được bao gồm trong thiết bị giải mã. Bộ dựng hình có thể bao gồm thiết bị hiển thị, và thiết bị hiển thị này có thể được tạo cấu hình dưới dạng thiết bị riêng biệt hoặc thành phần bên ngoài.

Nguồn video có thể thu được video/ảnh qua tiến trình ghi, tổng hợp, hoặc tạo ra video/ảnh. Nguồn video có thể bao gồm thiết bị ghi video/ảnh và/hoặc thiết bị tạo video/ảnh. Thiết bị ghi video/ảnh có thể bao gồm, ví dụ, một hoặc nhiều camera, các

thư viện lưu trữ video/ảnh bao gồm video/ảnh được ghi trước đó, v.v.. Thiết bị tạo video/ảnh có thể bao gồm, ví dụ, máy tính, máy tính bảng và điện thoại thông minh, và có thể tạo ra video/ảnh (về mặt điện tử). Ví dụ, video/ảnh ảo có thể được tạo ra bằng máy tính, v.v.. Trong trường hợp này, thì tiến trình ghi video/ảnh có thể được thay bằng tiến trình tạo ra dữ liệu liên quan.

Thiết bị mã hoá có thể mã hoá video/ảnh đầu vào. Thiết bị mã hoá có thể thực hiện một loạt thủ tục như dự đoán, biến đổi, và lượng tử hoá để thu được hiệu quả nén và lập mã. Dữ liệu được mã hoá (thông tin video/ảnh được mã hoá) có thể được xuất ra dưới dạng luồng bit.

Bộ phát có thể truyền thông tin video/ảnh được mã hoá hoặc dữ liệu được xuất ra này dưới dạng luồng bit đến bộ thu của thiết bị nhận qua phương tiện lưu trữ số hoặc mạng dưới dạng tệp tin hoặc việc tạo luồng. Phương tiện lưu trữ số này có thể bao gồm các phương tiện lưu trữ khác nhau như USB (Universal Serial Bus - buýt nối tiếp vạn năng), thẻ SD (Secure Digital - thẻ kỹ thuật số bảo mật), đĩa CD (Compact Disc - đĩa compắc), đĩa DVD (Digital Video Disc - đĩa video kỹ thuật số), đĩa Blu-ray, HDD (Hard Disk Drive - ổ đĩa cứng), SSD (Solid State Drive - ổ đĩa thể rắn), v.v.. Bộ phát có thể bao gồm phần tử để tạo ra tệp tin phương tiện qua định dạng tệp tin được định trước và có thể bao gồm phần tử để truyền qua mạng quảng bá/mạng truyền thông. Bộ thu có thể nhận/trích xuất luồng bit này và truyền luồng bit nhận được đến thiết bị giải mã.

Thiết bị giải mã có thể giải mã video/ảnh này bằng cách thực hiện một loạt thủ tục như giải lượng tử, biến đổi ngược, và dự đoán, tương ứng với hoạt động của thiết bị mã hoá.

Bộ dựng hình có thể dựng hình video/ảnh được giải mã này. Video/ảnh được dựng hình có thể được hiển thị qua thiết bị hiển thị.

Sáng chế liên quan đến việc lập mã video/ảnh. Ví dụ, phương pháp/phương án được bộc lộ trong bản mô tả này có thể được áp dụng cho các phương pháp được bộc lộ theo tiêu chuẩn lập mã video vạn năng (Versatile Video Coding - VVC), tiêu chuẩn

lập mã video thiết yếu (Essential Video Coding - EVC), tiêu chuẩn AOMedia Video 1 (AV1), thế hệ thứ hai của tiêu chuẩn lập mã audio video (2nd generation of Audio Video coding Standard - AVS2), hoặc tiêu chuẩn lập mã video/ảnh thế hệ tiếp theo (ví dụ, H.267 hoặc H.268, v.v.).

Sáng chế đề xuất các phương án khác nhau để lập mã video/ảnh, và các phương án nêu trên cũng có thể được thực hiện kết hợp với nhau trừ khi được quy định khác đi.

Theo sáng chế, thì video có thể là loạt ảnh theo thời gian. Ảnh nói chung là đơn vị biểu thị một hình ảnh tại khung thời gian cụ thể, và lát cắt/ô là đơn vị cấu thành một phần của ảnh xét về mặt lập mã. Lát cắt/ô có thể bao gồm một hoặc nhiều đơn vị cây lập mã (Coding Tree Unit - CTU). Một ảnh có thể bao gồm một hoặc nhiều lát cắt/ô.

Ô là vùng hình chữ nhật gồm các CTU trong cột ô cụ thể và hàng ô cụ thể trong ảnh. Cột ô là vùng hình chữ nhật của các CTU có chiều cao bằng chiều cao của ảnh và chiều rộng được quy định bởi các phần tử cú pháp trong tập hợp thông số ảnh. Hàng ô là vùng hình chữ nhật gồm các CTU và có chiều cao được quy định bởi các phần tử cú pháp trong tập hợp thông số ảnh và chiều rộng bằng chiều rộng của ảnh. Việc quét ô là việc xếp thứ tự tuần tự cụ thể đối với các CTU phân vùng ảnh mà trong đó các CTU được xếp thứ tự liên tiếp trong quá trình quét mảnh CTU trong ô, còn các ô trong ảnh thì được xếp thứ tự liên tiếp trong quá trình quét mảnh của các ô của ảnh. Lát cắt có thể bao gồm một số lượng ô hoàn chỉnh hoặc một số lượng hàng CTU liên tiếp trong một ô của ảnh mà có thể được chứa trong một đơn vị NAL. Trong bản mô tả này, thì nhóm ô và lát cắt có thể được sử dụng thay thế cho nhau. Ví dụ, trong bản mô tả này, thì nhóm ô/mào đầu nhóm ô có thể được gọi là lát cắt/mào đầu lát cắt.

Trong khi đó, một ảnh có thể được chia thành hai hoặc nhiều ảnh con. Ảnh con có thể là vùng hình chữ nhật của một hoặc nhiều lát cắt trong ảnh.

Điểm ảnh (pixel hay pel) có thể có nghĩa là đơn vị nhỏ nhất cấu thành một ảnh (hoặc hình ảnh). Ngoài ra, ‘mẫu’ có thể được sử dụng làm thuật ngữ tương ứng với điểm ảnh. Mẫu có thể nói chung là biểu thị điểm ảnh hoặc giá trị của điểm ảnh, và có thể chỉ

biểu thị điểm ảnh/giá trị điểm ảnh của thành phần độ chói hoặc chỉ biểu thị điểm ảnh/giá trị điểm ảnh của thành phần sắc độ.

Một đơn vị có thể biểu thị đơn vị cơ sở của quá trình xử lý ảnh. Đơn vị này có thể bao gồm ít nhất một trong số vùng cụ thể của ảnh và thông tin liên quan đến vùng đó. Một đơn vị có thể bao gồm một khối độ chói và hai khối sắc độ (ví dụ, cb, cr). Đơn vị có thể được sử dụng thay thế với các thuật ngữ như khối hoặc vùng trong một số trường hợp. Trong trường hợp tổng quát, thì khối $M \times N$ có thể bao gồm các mẫu (hoặc các mảng mẫu) hoặc tập hợp (hoặc mảng) của các hệ số biến đổi của M cột và N hàng. Theo cách khác, mẫu có thể có nghĩa là giá trị điểm ảnh trong miền không gian, và khi giá trị điểm ảnh đó được biến đổi sang miền tần số, thì nó có thể có nghĩa là hệ số biến đổi trong miền tần số.

Fig.2 là hình vẽ thể hiện sơ đồ cấu hình của thiết bị mã hoá video/ảnh mà sáng chế có thể được áp dụng vào đó. Sau đây, cái được gọi là thiết bị mã hoá video là có thể bao gồm thiết bị mã hoá ảnh.

Như được thể hiện trên Fig.2, thiết bị mã hoá 200 có thể bao gồm và được tạo cấu hình với bộ phân vùng ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hoá entropy 240, bộ cộng 250, bộ lọc 260, và bộ nhớ 270. Bộ dự đoán 220 có thể bao gồm bộ liên dự đoán 221 và bộ nội dự đoán 222. Bộ xử lý phần dư 230 có thể bao gồm bộ biến đổi 232, bộ lượng tử hoá 233, bộ giải lượng tử 234, và bộ biến đổi ngược 235. Bộ xử lý phần dư 230 có thể còn bao gồm bộ trừ 231. Bộ cộng 250 có thể được gọi là bộ tái tạo hoặc bộ tạo khối được tái tạo. Bộ phân vùng ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hoá entropy 240, bộ cộng 250, và bộ lọc 260, mà đã được mô tả trên đây, có thể được tạo cấu hình bởi một hoặc nhiều thành phần phần cứng (ví dụ, các con chip hoặc các bộ xử lý mã hoá) theo một phương án. Ngoài ra, bộ nhớ 270 có thể bao gồm bộ đệm ảnh được giải mã (Decoded Picture Buffer - DPB), và cũng có thể được tạo cấu hình bởi phương tiện lưu trữ số. Thành phần phần cứng này có thể còn bao gồm bộ nhớ 270 dưới dạng thành phần bên trong/bên ngoài.

Bộ phân vùng ảnh 210 có thể chia hình ảnh (hoặc ảnh, khung) đầu vào, mà được nhập vào thiết bị mã hoá 200, thành một hoặc nhiều đơn vị xử lý. Theo một ví dụ, đơn vị xử lý này có thể được gọi là đơn vị lập mã (Coding Unit - CU). Trong trường hợp này, đơn vị lập mã có thể được chia theo cách đệ quy theo cấu trúc cây tứ phân cây nhị phân cây tam phân (Quad-Tree Binary-Tree Ternary-Tree - QTBTNT) từ đơn vị cây lập mã (Coding Tree Unit - CTU) hoặc đơn vị lập mã lớn nhất (Largest Coding Unit - LCU). Ví dụ, một đơn vị lập mã có thể được chia thành nhiều đơn vị lập mã có chiều sâu sâu hơn dựa trên cấu trúc cây tứ phân, cấu trúc cây nhị phân, và/hoặc cấu trúc cây tam phân. Trong trường hợp này thì, ví dụ, cấu trúc cây tứ phân được áp dụng trước, và cấu trúc cây nhị phân và/hoặc cấu trúc cây tam phân có thể được áp dụng sau. Theo cách khác, cấu trúc cây nhị phân cũng có thể được áp dụng trước. Thủ tục lập mã theo sáng chế có thể được thực hiện dựa trên đơn vị lập mã cuối cùng mà không được chia tiếp nữa. Trong trường hợp này, dựa trên hiệu quả lập mã theo các đặc điểm hình ảnh, v.v., thì đơn vị lập mã lớn nhất có thể được sử dụng trực tiếp làm đơn vị lập mã cuối cùng, hoặc nếu cần, thì đơn vị lập mã này có thể được chia theo cách đệ quy thành các đơn vị lập mã có chiều sâu sâu hơn, để đơn vị lập mã có kích thước tối ưu có thể được sử dụng làm đơn vị lập mã cuối cùng. Ở đây, thủ tục lập mã có thể bao gồm thủ tục như dự đoán, biến đổi, và tái tạo, mà sẽ được mô tả sau. Theo ví dụ khác, đơn vị xử lý có thể còn bao gồm bộ dự đoán (PU) hoặc đơn vị biến đổi (Transform Unit - TU). Trong trường hợp này, mỗi trong số bộ dự đoán và đơn vị biến đổi có thể được chia hoặc được phân vùng từ đơn vị lập mã cuối cùng nêu trên. Bộ dự đoán có thể là đơn vị dự đoán mẫu, và đơn vị biến đổi có thể là đơn vị để sinh ra hệ số biến đổi và/hoặc đơn vị để sinh ra tín hiệu dư từ hệ số biến đổi.

Đơn vị này có thể được sử dụng hoán đổi với thuật ngữ chẳng hạn như khối hoặc vùng trong một số trường hợp. Nói chung, khối $M \times N$ có thể biểu thị các mẫu bao gồm M cột và N hàng hoặc nhóm hệ số biến đổi. Mẫu có thể nói chung là biểu thị điểm ảnh hoặc giá trị của điểm ảnh, và cũng có thể chỉ biểu thị điểm ảnh/giá trị điểm ảnh của

thành phần độ chói, và cũng có thể chỉ biểu thị điểm ảnh/giá trị điểm ảnh của thành phần sắc độ. Mẫu có thể được sử dụng làm thuật ngữ tương ứng với điểm ảnh hoặc pel mà tạo cấu hình một ảnh (hoặc hình ảnh).

Thiết bị mã hoá 200 có thể trừ đi tín hiệu dự đoán (khối được dự đoán, mảng mẫu dự đoán), mà được xuất ra từ bộ liên dự đoán 221 hoặc bộ nội dự đoán 222, từ tín hiệu hình ảnh đầu vào (khối gốc, mảng mẫu gốc) để tạo ra tín hiệu dư (khối dư, mảng mẫu dư), và tín hiệu dư được tạo ra này được truyền đến bộ biến đổi 232. Trong trường hợp này, như được thể hiện, đơn vị để trừ đi tín hiệu dự đoán (khối dự đoán, mảng mẫu dự đoán) từ tín hiệu hình ảnh đầu vào (khối gốc, mảng mẫu gốc) ở bộ mã hoá 200 có thể được gọi là bộ trừ 231. Bộ dự đoán có thể thực hiện việc dự đoán trên khối mục tiêu xử lý (sau đây được gọi là khối hiện tại) và tạo ra khối được dự đoán bao gồm các mẫu dự đoán dành cho khối hiện tại. Bộ dự đoán có thể xác định xem việc nội dự đoán hay việc liên dự đoán được áp dụng ở các đơn vị của khối hiện tại hoặc CU. Bộ dự đoán có thể tạo ra các thông tin khác nhau trên việc dự đoán, chẳng hạn thông tin về chế độ dự đoán, và truyền thông tin được tạo ra đó đến bộ mã hoá entropy 240, như được mô tả dưới đây trong phần mô tả về mỗi chế độ dự đoán. Thông tin về việc dự đoán này có thể được mã hoá bởi bộ mã hoá entropy 240 và được xuất ra dưới dạng luồng bit.

Bộ nội dự đoán 222 có thể dự đoán khối hiện tại dựa vào các mẫu trong ảnh hiện tại. Các mẫu được tham chiếu có thể được đặt ở lân cận khối hiện tại, hoặc cũng có thể được đặt cách khối khối hiện tại theo chế độ dự đoán. Các chế độ dự đoán trong việc nội dự đoán có thể bao gồm các chế độ không có hướng và các chế độ có hướng. Chế độ không có hướng có thể bao gồm, ví dụ, chế độ DC (Direct Current - dòng điện một chiều) hoặc chế độ phẳng. Chế độ có hướng có thể bao gồm, ví dụ, 33 chế độ dự đoán có hướng hoặc 65 chế độ dự đoán có hướng theo độ mịn của chiều dự đoán. Tuy nhiên, đây chỉ là minh họa, và các chế độ dự đoán có hướng với số lượng nhiều hoặc ít hơn so với số lượng nêu trên cũng có thể được sử dụng theo sự thiết đặt. Bộ nội dự đoán 222 cũng có thể xác định chế độ dự đoán được áp dụng cho khối hiện tại nhờ sử dụng

chế độ dự đoán được áp dụng cho khối lân cận.

Bộ liên dự đoán 221 có thể sinh ra khối được dự đoán của khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được quy định bởi vector chuyển động trên ảnh tham chiếu. Lúc này, để giảm lượng thông tin chuyển động được truyền trong chế độ liên dự đoán, thì thông tin chuyển động này có thể được dự đoán theo các đơn vị là khối, khối con, hoặc mẫu dựa trên sự tương quan của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động này có thể bao gồm vector chuyển động và chỉ số ảnh tham chiếu. Thông tin chuyển động này có thể còn bao gồm thông tin về chiều liên dự đoán (dự đoán L0, dự đoán L1, dự đoán Bi, v.v.). Trong trường hợp liên dự đoán, thì khối lân cận có thể bao gồm khối lân cận về không gian tồn tại trong ảnh hiện tại và khối lân cận về thời gian tồn tại trong ảnh tham chiếu. Ảnh tham chiếu mà bao gồm khối tham chiếu và ảnh tham chiếu mà bao gồm khối lân cận về thời gian này cũng có thể là giống nhau, và cũng có thể là khác nhau. Khối lân cận về thời gian có thể được gọi với tên gọi chẳng hạn như khối tham chiếu được đặt cùng vị trí, CU được đặt cùng vị trí (collocated CU - colCU), v.v., và ảnh tham chiếu bao gồm khối lân cận về thời gian cũng có thể được gọi là ảnh được đặt cùng vị trí (collocated Picture - colPic). Ví dụ, bộ liên dự đoán 221 có thể tạo cấu hình danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận, và tạo ra thông tin chỉ thị xem ứng viên nào được dùng để trích xuất vector chuyển động và/hoặc chỉ số ảnh tham chiếu của khối hiện tại. Việc liên dự đoán có thể được thực hiện dựa trên các chế độ dự đoán khác nhau, và, ví dụ, trong trường hợp chế độ bỏ cóc và chế độ hợp nhất, thì bộ liên dự đoán 221 có thể sử dụng thông tin chuyển động của khối lân cận làm thông tin chuyển động của khối hiện tại. Trong trường hợp chế độ bỏ cóc, thì tín hiệu dư có thể không được truyền, không giống như chế độ hợp nhất. Chế độ dự đoán vector chuyển động (Motion Vector Prediction - MVP) có thể chỉ thị vector chuyển động của khối hiện tại bằng cách sử dụng vector chuyển động của khối lân cận làm bộ dự đoán vector chuyển động, và báo hiệu hiệu vector chuyển động.

Bộ dự đoán 220 có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau, sẽ được mô tả dưới đây. Ví dụ, bộ dự đoán có thể áp dụng việc nội dự đoán hoặc việc liên dự đoán cho việc dự đoán một khối và có thể đồng thời áp dụng việc nội dự đoán và việc liên dự đoán. Việc này có thể được gọi là liên dự đoán và nội dự đoán kết hợp (Combined Inter and Intra Prediction - CIIP). Ngoài ra, bộ dự đoán có thể dựa trên chế độ dự đoán sao chép nội khối (Intra Block Copy - IBC) hoặc dựa trên chế độ bảng màu đối với việc dự đoán khối. Chế độ dự đoán IBC hoặc chế độ bảng màu có thể được sử dụng để lập mã ảnh/video của nội dung, chẳng hạn các trò chơi, ví dụ, lập mã nội dung trên màn hình (Screen Content Coding - SCC). IBC về cơ bản thực hiện việc dự đoán trong ảnh hiện tại, nhưng có thể được thực hiện tương tự như việc liên dự đoán ở chỗ khối tham chiếu là được trích xuất trong ảnh hiện tại. Tức là, IBC có thể sử dụng ít nhất một trong số các kỹ thuật liên dự đoán được mô tả trong bản mô tả này. Chế độ bảng màu có thể được xem như là một ví dụ của việc nội lập mã hoặc nội dự đoán. Khi chế độ bảng màu được áp dụng, thì giá trị mẫu trong ảnh có thể được báo hiệu dựa trên thông tin về bảng màu và chỉ số bảng màu.

Tín hiệu dự đoán mà được tạo ra bởi bộ dự đoán (bao gồm bộ liên dự đoán 221 và/hoặc bộ nội dự đoán 222) có thể được dùng để tạo ra tín hiệu được tái tạo hoặc có thể được dùng để tạo ra tín hiệu dư. Bộ biến đổi 232 có thể tạo ra các hệ số biến đổi bằng cách áp dụng kỹ thuật biến đổi cho tín hiệu dư này. Ví dụ, kỹ thuật biến đổi có thể bao gồm ít nhất một trong số kỹ thuật biến đổi cosin rời rạc (Discrete Cosine Transform - DCT), biến đổi sin rời rạc (Discrete Sine Transform - DST), biến đổi Karhunen–Loève (Karhunen–Loeve Transform - KLT), biến đổi dựa trên đồ thị (Graph-Based Transform - GBT), hoặc biến đổi phi tuyến có điều kiện (Conditionally Non-linear Transform - CNT). Ở đây, GBT là phép biến đổi thu được từ đồ thị khi biểu diễn thông tin quan hệ giữa các điểm ảnh trên đồ thị. CNT là phép biến đổi thu được dựa trên tín hiệu dự đoán được tạo ra nhờ sử dụng tất cả các điểm ảnh được tái tạo trước đó. Ngoài ra, tiến trình biến đổi này có thể được áp dụng cho khối điểm ảnh có cùng kích thước như hình vuông

hoặc có thể được áp dụng cho khối có kích thước biến thiên mà không phải là hình vuông.

Bộ lượng tử hoá 233 lượng tử hoá các hệ số biến đổi và truyền chúng đến bộ mã hoá entropy 240, và bộ mã hoá entropy 240 mã hoá tín hiệu được lượng tử hoá (thông tin về các hệ số biến đổi được lượng tử hoá) và xuất tín hiệu được mã hoá này ra dưới dạng luồng bit. Thông tin về các hệ số biến đổi được lượng tử hoá có thể được gọi là thông tin dư. Bộ lượng tử hoá 233 có thể sắp xếp lại các hệ số biến đổi được lượng tử hoá ở dạng khối thành dạng vector một chiều dựa trên thứ tự quét hệ số và có thể tạo ra thông tin về các hệ số biến đổi dựa trên các hệ số biến đổi được lượng tử hoá ở dạng vector một chiều này. Bộ mã hoá entropy 240 có thể thực hiện các phương pháp mã hoá khác nhau, ví dụ như Golomb hàm số mũ, lập mã chiều dài biến thiên thích ứng ngữ cảnh (Context-Adaptive Variable Length Coding - CAVLC), và lập mã số học nhị phân thích ứng ngữ cảnh (Context-Adaptive Binary Arithmetic Coding - CABAC). Bộ mã hoá entropy 240 có thể mã hoá các thông tin cần thiết cho việc tái tạo video/ảnh (ví dụ, các giá trị của các phần tử cú pháp, v.v.), ngoài các hệ số biến đổi được lượng tử hoá, cùng nhau hoặc một cách riêng biệt. Thông tin được mã hoá (ví dụ, thông tin video/ảnh được mã hoá) có thể được truyền hoặc được lưu giữ theo đơn vị là đơn vị lớp trừu tượng mạng (Network Abstraction Layer - NAL) dưới dạng luồng bit. Thông tin video/ảnh có thể còn bao gồm thông tin về các tập hợp thông số khác nhau, chẳng hạn tập hợp thông số thích ứng (Adaptation Parameter Set - APS), tập hợp thông số ảnh (Picture Parameter Set - PPS), tập hợp thông số chuỗi (Sequence Parameter Set - SPS), hoặc tập hợp thông số video (Video Parameter Set - VPS). Ngoài ra, thông tin video/ảnh có thể còn bao gồm thông tin ràng buộc chung. Trong bản mô tả này, thì thông tin và/hoặc các phần tử cú pháp mà được truyền/được báo hiệu từ thiết bị mã hoá đến thiết bị giải mã là có thể được bao gồm trong thông tin video/ảnh. Thông tin video/ảnh này có thể được mã hoá thông qua thủ tục mã hoá được mô tả trên đây và được bao gồm trong luồng bit. Luồng bit này có thể được truyền qua mạng hoặc có thể được lưu giữ

trong phương tiện lưu trữ số. Ở đây, mạng có thể bao gồm mạng quảng bá và/hoặc mạng truyền thông, và phương tiện lưu trữ kỹ thuật số này có thể bao gồm các phương tiện lưu trữ khác nhau như USB, thẻ SD, đĩa CD, đĩa DVD, đĩa Blu-ray, ổ HDD, và ổ SSD. Đơn vị truyền (không được thể hiện trên hình vẽ) và/hoặc đơn vị lưu trữ (không được thể hiện trên hình vẽ), để truyền hoặc lưu giữ tín hiệu được xuất ra từ bộ mã hoá entropy 240, là có thể được tạo cấu hình dưới dạng các phần tử bên trong/bên ngoài của thiết bị mã hoá 200, hoặc đơn vị truyền có thể được bao gồm trong bộ mã hoá entropy 240.

Các hệ số biến đổi được lượng tử hoá, mà được xuất ra từ bộ lượng tử hoá 233, có thể được dùng để tạo ra tín hiệu dự đoán. Ví dụ, tín hiệu dư (khối dư hoặc các mẫu dư) có thể được tái tạo bằng cách áp dụng phép giải lượng tử và phép biến đổi ngược cho các hệ số biến đổi được lượng tử hoá thông qua bộ giải lượng tử 234 và đơn vị biến đổi ngược 235. Bộ cộng 250 có thể cộng tín hiệu dư được tái tạo vào tín hiệu dự đoán được xuất ra từ bộ liên dự đoán 221 hoặc bộ nội dự đoán 222 để tạo ra tín hiệu được tái tạo (ảnh được tái tạo, khối được tái tạo, mảng mẫu được tái tạo). Khi không có phần dư nào cho khối mục tiêu xử lý, chẳng hạn khi chế độ bỏ cóc được áp dụng, thì khối được dự đoán có thể được sử dụng làm khối được tái tạo. Bộ cộng 250 có thể được gọi là đơn vị khôi phục hoặc bộ tạo khối khôi phục. Tín hiệu được tái tạo được tạo ra có thể được sử dụng cho việc nội dự đoán khối mục tiêu xử lý tiếp theo trong ảnh hiện tại, hoặc có thể được sử dụng cho việc liên dự đoán ảnh tiếp theo sau khi được lọc, như được mô tả dưới đây.

Trong khi đó, việc ánh xạ độ chói có thay đổi tỷ lệ sắc độ (Luma Mapping with Chroma Scaling - LMCS) có thể được áp dụng trong tiến trình mã hoá và/hoặc tái tạo ảnh.

Bộ lọc 260 có thể cải thiện chất lượng hình ảnh chủ quan/khách quan bằng cách áp dụng việc lọc cho tín hiệu được tái tạo. Ví dụ, bộ lọc 260 có thể tạo ra ảnh được tái tạo được cải biến bằng cách áp dụng các phương pháp lọc khác nhau cho ảnh được tái tạo, và lưu giữ ảnh được tái tạo được cải biến này trong bộ nhớ 270, cụ thể là, trong

DPB của bộ nhớ 270. Các phương pháp lọc khác nhau này có thể bao gồm, ví dụ, lọc khử khối, dịch thích ứng mẫu, lọc vòng thích ứng, lọc hai chiều, v.v.. Bộ lọc 260 có thể tạo ra các loại thông tin khác nhau liên quan đến việc lọc, và truyền thông tin được tạo ra đến bộ mã hoá entropy 240, như được mô tả sau trong phần mô tả về mỗi phương pháp lọc. Thông tin liên quan đến việc lọc này có thể được mã hoá bởi bộ mã hoá entropy 240 và được xuất ra dưới dạng luồng bit.

Ảnh được tái tạo được cải biến mà được truyền đến bộ nhớ 270 là có thể được sử dụng làm ảnh tham chiếu ở bộ liên dự đoán 221. Khi việc liên dự đoán được áp dụng thông qua thiết bị mã hoá, thì có thể tránh được sự không khớp dự đoán giữa thiết bị mã hoá 200 và thiết bị giải mã, và hiệu quả mã hoá có thể được cải thiện.

DPB của bộ nhớ 270 có thể lưu giữ ảnh được tái tạo được cải biến để sử dụng làm ảnh tham chiếu ở bộ liên dự đoán 221. Bộ nhớ 270 có thể lưu giữ thông tin chuyển động của khối mà thông tin chuyển động trong ảnh hiện tại được trích xuất (hoặc được mã hoá) từ đó và/hoặc thông tin chuyển động của các khối trong ảnh, mà đã được tái tạo. Thông tin chuyển động được lưu giữ này có thể được truyền đến bộ liên dự đoán 221 để sử dụng làm thông tin chuyển động của khối lân cận về không gian hoặc thông tin chuyển động của khối lân cận về thời gian. Bộ nhớ 270 có thể lưu giữ các mẫu được tái tạo của các khối được tái tạo trong ảnh hiện tại, và có thể truyền các mẫu được tái tạo này đến bộ nội dự đoán 222.

Trong khi đó, theo sáng chế, thì ít nhất một trong số việc lượng tử hoá/giải lượng tử và/hoặc phép biến đổi/biến đổi ngược là có thể được lược bỏ. Khi việc lượng tử hoá/giải lượng tử được lược bỏ, thì hệ số biến đổi được lượng tử hoá có thể được gọi là hệ số biến đổi. Khi phép biến đổi/biến đổi ngược được lược bỏ, thì hệ số biến đổi có thể được gọi là hệ số hoặc hệ số dư hoặc có thể vẫn được gọi là hệ số biến đổi để nhất quán trong việc biểu diễn.

Ngoài ra, trong bản mô tả này, thì hệ số biến đổi được lượng tử hoá và hệ số biến đổi có thể được gọi lần lượt là hệ số biến đổi và hệ số biến đổi được thay đổi tỷ lệ.

Trong trường hợp này, thông tin dư có thể bao gồm thông tin về (các) hệ số biến đổi, và thông tin về (các) hệ số biến đổi này có thể được báo hiệu thông qua cú pháp lập mã phần dư. Các hệ số biến đổi có thể được trích xuất dựa trên thông tin dư (hoặc thông tin về (các) hệ số biến đổi), và các hệ số biến đổi được thay đổi tỷ lệ có thể được trích xuất thông qua việc (thay đổi tỷ lệ) biến đổi ngược trên các hệ số biến đổi này. Các mẫu dư có thể được trích xuất dựa trên việc biến đổi ngược (biến đổi) đối với các hệ số biến đổi được thay đổi tỷ lệ. Điều này cũng có thể được áp dụng/được biểu diễn ở các phần khác trong bản mô tả này.

Fig.3 là hình vẽ thể hiện sơ đồ mô tả cấu hình của thiết bị giải mã video/ảnh mà sáng chế có thể được áp dụng vào đó.

Như được thể hiện trên Fig.3, thiết bị giải mã 300 có thể bao gồm và được tạo cấu hình với bộ giải mã entropy 310, bộ xử lý phần dư 320, bộ dự đoán 330, bộ cộng 340, bộ lọc 350, và bộ nhớ 360. Bộ dự đoán 330 có thể bao gồm bộ nội dự đoán 331 và bộ liên dự đoán 332. Bộ xử lý phần dư 320 có thể bao gồm bộ giải lượng tử 321 và bộ biến đổi ngược 322. Bộ giải mã entropy 310, bộ xử lý phần dư 320, bộ dự đoán 330, bộ cộng 340, và bộ lọc 350, mà đã được mô tả trên đây, có thể được tạo cấu hình bởi một hoặc nhiều thành phần phần cứng (ví dụ, các con chip hoặc các bộ xử lý giải mã) theo một phương án. Ngoài ra, bộ nhớ 360 có thể bao gồm bộ đệm ảnh được giải mã (Decoded Picture Buffer - DPB), và có thể được tạo cấu hình bởi phương tiện lưu trữ số. Thành phần phần cứng này có thể còn bao gồm bộ nhớ 360 dưới dạng thành phần bên trong/bên ngoài.

Khi luồng bit bao gồm thông tin video/ảnh này được nhập vào, thì thiết bị giải mã 300 có thể tái tạo hình ảnh đáp lại tiến trình mà trong đó thông tin video/ảnh này được xử lý ở thiết bị mã hoá được thể hiện trên Fig.2. Ví dụ, thiết bị giải mã 300 có thể trích xuất các đơn vị/các khối dựa trên thông tin liên quan đến việc chia khối thu thập được từ luồng bit này. Thiết bị giải mã 300 có thể thực hiện việc giải mã nhờ sử dụng đơn vị xử lý được áp dụng cho thiết bị mã hoá. Do đó, đơn vị xử lý dành cho việc giải

mã này có thể là, ví dụ, đơn vị lập mã, và đơn vị lập mã này có thể được chia theo cấu trúc cây tứ phân, cấu trúc cây nhị phân, và/hoặc cấu trúc cây tam phân từ đơn vị cây lập mã hoặc đơn vị lập mã lớn nhất. Một hoặc nhiều đơn vị biến đổi có thể được trích xuất từ đơn vị lập mã này. Ngoài ra, tín hiệu ảnh được tái tạo mà được giải mã và được xuất ra qua thiết bị giải mã 300 là có thể được tái tạo qua thiết bị tái tạo.

Thiết bị giải mã 300 có thể nhận tín hiệu được xuất ra từ thiết bị mã hoá trên Fig.2 dưới dạng luồng bit, và tín hiệu nhận được này có thể được giải mã qua bộ giải mã entropy 310. Ví dụ, bộ giải mã entropy 310 có thể phân tích luồng bit này để trích xuất thông tin (ví dụ, thông tin video/ảnh) cần thiết cho việc tái tạo hình ảnh (hoặc tái tạo ảnh). Thông tin video/ảnh có thể còn bao gồm thông tin về các tập hợp thông số khác nhau, chẳng hạn tập hợp thông số thích ứng (Adaptation Parameter Set - APS), tập hợp thông số ảnh (Picture Parameter Set - PPS), tập hợp thông số chuỗi (Sequence Parameter Set - SPS), hoặc tập hợp thông số video (Video Parameter Set - VPS). Ngoài ra, thông tin video/ảnh có thể còn bao gồm thông tin ràng buộc chung. Thiết bị giải mã này có thể còn giải mã ảnh dựa trên thông tin về tập hợp thông số và/hoặc thông tin ràng buộc chung. Thông tin và/hoặc các phần tử cú pháp được báo hiệu/nhận được, mà được mô tả sau trong bản mô tả này, là có thể được giải mã bởi thủ tục giải mã và được thu thập từ luồng bit. Ví dụ, bộ giải mã entropy 310 giải mã thông tin trong luồng bit này dựa trên phương pháp lập mã chẳng hạn như lập mã Golomb hàm số mũ, lập mã chiều dài biến thiên thích ứng ngữ cảnh (Context-Adaptive Variable Length Coding - CAVLC), hoặc lập mã số học thích ứng ngữ cảnh (Context-Adaptive Arithmetic Coding - CABAC), và xuất ra các phần tử cú pháp cần thiết cho việc tái tạo ảnh và các giá trị được lượng tử hoá của các hệ số biến đổi cho phần dư. Cụ thể hơn, phương pháp giải mã entropy CABAC có thể nhận cụm điểm ảnh (bin) tương ứng với mỗi phần tử cú pháp trong luồng bit, xác định mô hình ngữ cảnh nhờ sử dụng thông tin về phần tử cú pháp mục tiêu giải mã, thông tin giải mã của khối mục tiêu giải mã hoặc thông tin của ký hiệu/cụm điểm ảnh được giải mã trong giai đoạn trước đó, và thực hiện việc giải mã

số học trên cụm điểm ảnh này bằng cách dự đoán xác suất xuất hiện của cụm điểm ảnh theo mô hình ngữ cảnh xác định được, và tạo ra ký hiệu tương ứng với giá trị của mỗi phần tử cú pháp. Trong trường hợp này, phương pháp giải mã entropy CABAC có thể cập nhật mô hình ngữ cảnh nhờ sử dụng thông tin của ký hiệu/cụm điểm ảnh được giải mã cho mô hình ngữ cảnh của ký hiệu/cụm điểm ảnh tiếp theo sau khi xác định mô hình ngữ cảnh này. Thông tin liên quan đến việc dự đoán, trong số các thông tin được giải mã bởi bộ giải mã entropy 310, là có thể được cung cấp cho bộ dự đoán (bộ liên dự đoán 332 và bộ nội dự đoán 331), và các giá trị dư, mà việc giải mã entropy đã được thực hiện trên đó ở bộ giải mã entropy 310, tức là các hệ số biến đổi được lượng tử hoá và thông tin thông số liên quan, là có thể được nhập vào bộ xử lý phần dư 320.

Bộ giải lượng tử 321 có thể giải lượng tử các hệ số biến đổi được lượng tử hoá để xuất ra các hệ số biến đổi. Bộ giải lượng tử 321 có thể sắp xếp lại các hệ số biến đổi được lượng tử hoá thành dạng khối hai chiều. Trong trường hợp này, việc sắp xếp lại có thể được thực hiện dựa trên thứ tự quét hệ số được thực hiện bởi thiết bị mã hoá. Bộ giải lượng tử 321 có thể thực hiện việc giải lượng tử đối với các hệ số biến đổi được lượng tử hoá nhờ sử dụng thông số lượng tử hoá (ví dụ, thông tin kích thước bước lượng tử), và thu thập các hệ số biến đổi.

Bộ biến đổi ngược 322 biến đổi ngược các hệ số biến đổi này để thu được tín hiệu dư (khối dư, mảng mẫu dư).

Bộ dự đoán 330 có thể thực hiện việc dự đoán đối với khối hiện tại, và tạo ra khối được dự đoán bao gồm các mẫu dự đoán của khối hiện tại. Bộ dự đoán có thể xác định xem việc nội dự đoán được áp dụng hay việc liên dự đoán được áp dụng cho khối hiện tại dựa trên thông tin về việc dự đoán được xuất ra từ bộ giải mã entropy 310, và xác định chế độ nội dự đoán/liên dự đoán cụ thể.

Bộ dự đoán 330 có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau, sẽ được mô tả sau. Ví dụ, bộ dự đoán có thể áp dụng việc nội dự đoán hoặc việc liên dự đoán cho việc dự đoán một khối, và có thể đồng thời áp dụng việc nội

dự đoán và việc liên dự đoán. Việc này có thể được gọi là liên dự đoán và nội dự đoán kết hợp (Combined Inter and Intra Prediction - CIIP). Ngoài ra, bộ dự đoán có thể dựa trên chế độ dự đoán sao chép nội khối (Intra Block Copy - IBC) hoặc dựa trên chế độ bảng màu đối với việc dự đoán khối. Chế độ dự đoán IBC hoặc chế độ bảng màu có thể được sử dụng để lập mã ảnh/video của nội dung, chẳng hạn các trò chơi, ví dụ, lập mã nội dung trên màn hình (Screen Content Coding - SCC). IBC có thể về cơ bản thực hiện việc dự đoán trong ảnh hiện tại, nhưng có thể được thực hiện tương tự như việc liên dự đoán ở chỗ khối tham chiếu là được trích xuất trong ảnh hiện tại. Tức là, IBC có thể sử dụng ít nhất một trong số các kỹ thuật liên dự đoán được mô tả trong bản mô tả này. Chế độ bảng màu có thể được coi như là một ví dụ của việc nội lập mã hoặc nội dự đoán. Khi chế độ bảng màu được áp dụng, thì thông tin về bảng màu và chỉ số bảng màu có thể được bao gồm trong thông tin video/ảnh và được báo hiệu.

Bộ nội dự đoán 3321 có thể dự đoán khối hiện tại bằng cách tham chiếu đến các mẫu trong ảnh hiện tại. Các mẫu được tham chiếu có thể được đặt ở gần khối hiện tại, hoặc có thể được đặt cách khối hiện tại theo chế độ dự đoán. Trong việc nội dự đoán, thì các chế độ dự đoán có thể bao gồm các chế độ không có hướng và các chế độ có hướng. Bộ nội dự đoán 331 có thể xác định chế độ dự đoán để áp dụng cho khối hiện tại nhờ sử dụng chế độ dự đoán được áp dụng cho khối lân cận.

Bộ liên dự đoán 332 có thể trích xuất khối được dự đoán cho khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được quy định bởi vector chuyển động trên ảnh tham chiếu. Trong trường hợp này, để giảm lượng thông tin chuyển động được truyền trong chế độ liên dự đoán, thì thông tin chuyển động có thể được dự đoán theo đơn vị là các khối, các khối con, hoặc các mẫu dựa trên sự tương quan của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động này có thể bao gồm vector chuyển động và chỉ số ảnh tham chiếu. Thông tin chuyển động này có thể còn bao gồm thông tin về chiều liên dự đoán (dự đoán L0, dự đoán L1, dự đoán Bi, v.v.). Trong trường hợp liên dự đoán, thì khối lân cận có thể bao gồm khối lân cận về

không gian đang tồn tại trong ảnh hiện tại và khối lân cận về thời gian đang tồn tại trong ảnh tham chiếu. Ví dụ, bộ liên dự đoán 332 có thể xây dựng danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận, và trích xuất vector chuyển động của khối hiện tại và/hoặc chỉ số ảnh tham chiếu dựa trên thông tin chọn ứng viên nhận được. Việc liên dự đoán có thể được thực hiện dựa trên các chế độ dự đoán khác nhau, và thông tin về việc dự đoán có thể bao gồm thông tin chỉ thị chế độ liên dự đoán đối với khối hiện tại.

Bộ cộng 340 có thể tạo ra tín hiệu được tái tạo (ảnh được tái tạo, khối được tái tạo, hoặc mảng mẫu được tái tạo) bằng cách cộng tín hiệu dự thu được vào tín hiệu dự đoán (khối được dự đoán hoặc mảng mẫu được dự đoán) mà được xuất ra từ bộ dự đoán (bao gồm bộ liên dự đoán 332 và/hoặc bộ nội dự đoán 331). Nếu không có phần dư nào cho khối mục tiêu xử lý, chẳng hạn trường hợp mà chế độ bỏ cóc được áp dụng, thì khối được dự đoán có thể được sử dụng làm khối được tái tạo.

Bộ cộng 340 có thể được gọi là bộ tái tạo hoặc bộ tạo khối được tái tạo. Tín hiệu được tái tạo được tạo ra có thể được sử dụng cho việc nội dự đoán khối tiếp theo cần được xử lý trong ảnh hiện tại, và như được mô tả sau, cũng có thể được xuất ra qua việc lọc hoặc cũng có thể được sử dụng cho việc liên dự đoán đối với ảnh tiếp theo.

Trong khi đó, việc ánh xạ độ chói có thay đổi tỷ lệ sắc độ (Luma Mapping with Chroma Scaling - LMCS) cũng có thể được áp dụng trong tiến trình giải mã ảnh.

Bộ lọc 350 có thể cải thiện chất lượng hình ảnh chủ quan/khách quan bằng cách áp dụng việc lọc cho tín hiệu được tái tạo. Ví dụ, bộ lọc 350 có thể tạo ra ảnh được tái tạo được cải biến bằng cách áp dụng các phương pháp lọc khác nhau cho ảnh được tái tạo, và lưu giữ ảnh được tái tạo được cải biến này trong bộ nhớ 360, cụ thể là, trong DPB của bộ nhớ 360. Các phương pháp lọc khác nhau này có thể bao gồm, ví dụ, lọc khử khối, dịch thích ứng mẫu, lọc vòng thích ứng, lọc hai chiều, v.v..

Ảnh được tái tạo (được cải biến) được lưu giữ trong DPB của bộ nhớ 360 có thể được sử dụng làm ảnh tham chiếu ở bộ liên dự đoán 332. Bộ nhớ 360 có thể lưu giữ

thông tin chuyển động của khối mà thông tin chuyển động trong ảnh hiện tại được trích xuất (hoặc được giải mã) từ đó và/hoặc thông tin chuyển động của các khối trong ảnh mà đã được tái tạo. Thông tin chuyển động được lưu giữ này có thể được truyền đến bộ liên dự đoán 332 để sử dụng làm thông tin chuyển động của khối lân cận về không gian hoặc thông tin chuyển động của khối lân cận về thời gian. Bộ nhớ 360 có thể lưu giữ các mẫu được tái tạo của các khối được tái tạo trong ảnh hiện tại, và truyền các mẫu được tái tạo này đến bộ nội dự đoán 331.

Theo sáng chế, thì các phương án mà được mô tả ở bộ lọc 260, bộ liên dự đoán 221, và bộ nội dự đoán 222 của thiết bị mã hoá 200 là có thể được áp dụng tương đương hoặc để tương ứng với bộ lọc 350, bộ liên dự đoán 332, và bộ nội dự đoán 331.

Trong khi đó, như đã mô tả trên đây, khi thực hiện việc lập mã video, thì việc dự đoán được thực hiện để cải thiện hiệu quả nén. Thông qua hoạt động này, thì khối được dự đoán mà bao gồm các mẫu dự đoán cho khối hiện tại dưới dạng khối cần được lập mã (tức là, khối mục tiêu lập mã) là có thể được tạo ra. Ở đây, khối được dự đoán này bao gồm các mẫu dự đoán trong miền không gian (hoặc miền điểm ảnh). Khối được dự đoán này được trích xuất theo cách giống nhau ở thiết bị mã hoá và thiết bị giải mã, và thiết bị mã hoá có thể báo hiệu thông tin (thông tin dư) về phần dư giữa khối gốc và khối được dự đoán, thay vì giá trị mẫu gốc của khối gốc, cho thiết bị giải mã, nhờ đó tăng hiệu quả lập mã ảnh. Thiết bị giải mã có thể trích xuất khối dư bao gồm các mẫu dư dựa trên thông tin dư này, cộng khối dư và khối được dự đoán này để tạo ra các khối được tái tạo bao gồm các mẫu được tái tạo, và tạo ra ảnh được tái tạo bao gồm các khối được tái tạo này.

Thông tin dư này có thể được tạo ra qua thủ tục biến đổi và lượng tử hoá. Ví dụ, thiết bị mã hoá có thể trích xuất khối dư giữa khối gốc và khối được dự đoán, thực hiện thủ tục biến đổi trên các mẫu dư (mảng mẫu dư) được bao gồm trong khối dư này để trích xuất các hệ số biến đổi, thực hiện thủ tục lượng tử hoá trên các hệ số biến đổi này để trích xuất các hệ số biến đổi được lượng tử hoá, và báo hiệu thông tin dư liên quan

cho thiết bị giải mã (qua luồng bit). Ở đây, thông tin dư này có thể bao gồm thông tin về giá trị của các hệ số biến đổi được lượng tử hoá, thông tin vị trí, kỹ thuật biến đổi, nhân biến đổi, thông số lượng tử hoá, v.v.. Thiết bị giải mã có thể thực hiện thủ tục giải lượng tử/biến đổi ngược dựa trên thông tin dư và trích xuất các mẫu dư (hoặc các khối dư). Thiết bị giải mã có thể tạo ra ảnh được tái tạo dựa trên khối được dự đoán và khối dư. Ngoài ra, đối với sự tham chiếu cho việc liên dự đoán ảnh sau đó, thì thiết bị mã hoá cũng có thể giải lượng tử/biến đổi ngược các hệ số biến đổi được lượng tử hoá để trích xuất khối dư và tạo ra ảnh được tái tạo dựa trên đó.

Fig.4 là hình vẽ thể hiện sơ đồ mô tả chế độ hợp nhất trong quá trình liên dự đoán.

Khi chế độ hợp nhất được áp dụng, thì thông tin chuyển động trên khối dự đoán hiện tại là không được truyền trực tiếp, mà thông tin chuyển động trên khối dự đoán hiện tại là được trích xuất nhờ sử dụng thông tin chuyển động trên khối dự đoán lân cận. Theo đó, thông tin chuyển động trên khối dự đoán hiện tại có thể được chỉ thị bằng cách truyền thông tin cờ chỉ thị rằng chế độ hợp nhất được sử dụng và chỉ số hợp nhất chỉ thị khối dự đoán nào ở gần được sử dụng. Chế độ hợp nhất này có thể được gọi là chế độ hợp nhất bình thường. Ví dụ, chế độ hợp nhất này có thể được áp dụng khi giá trị của phần tử cú pháp `regular_merge_flag` là 1.

Để thực hiện chế độ hợp nhất, thì thiết bị mã hoá cần phải tìm kiếm khối ứng viên hợp nhất mà được dùng để trích xuất thông tin chuyển động trên khối dự đoán hiện tại. Ví dụ, lên tới năm khối ứng viên hợp nhất có thể được sử dụng, nhưng (các) phương án của sáng chế không chỉ giới hạn ở đó. Ngoài ra, số lượng tối đa các khối ứng viên hợp nhất có thể được truyền trong mào đầu lát cắt hoặc mào đầu nhóm ô, nhưng (các) phương án của sáng chế không chỉ giới hạn ở đó. Sau khi tìm được các khối ứng viên hợp nhất, thì thiết bị mã hoá có thể tạo ra danh sách ứng viên hợp nhất, và có thể chọn khối ứng viên hợp nhất có chi phí nhỏ nhất trong số các khối ứng viên hợp nhất này làm khối ứng viên hợp nhất cuối cùng.

Sáng chế có thể cung cấp các phương án khác nhau về các khối ứng viên hợp nhất mà cấu thành danh sách ứng viên hợp nhất.

Ví dụ, danh sách ứng viên hợp nhất có thể sử dụng năm khối ứng viên hợp nhất. Ví dụ, bốn ứng viên hợp nhất không gian và một ứng viên hợp nhất thời gian có thể được sử dụng. Theo một ví dụ cụ thể, trong trường hợp ứng viên hợp nhất không gian, thì các khối được thể hiện trên Fig.4 có thể được sử dụng làm các ứng viên hợp nhất không gian. Sau đây, ứng viên hợp nhất không gian hoặc ứng viên MVP không gian, mà sẽ được mô tả sau, có thể được gọi là SMVP, và ứng viên hợp nhất thời gian hoặc ứng viên MVP thời gian, mà sẽ được mô tả sau, có thể được gọi là TMVP.

Danh sách ứng viên hợp nhất đối với khối hiện tại có thể được xây dựng, ví dụ, dựa trên thủ tục sau đây.

Thiết bị mã hoá/thiết bị giải mã có thể tìm kiếm các khối lân cận về không gian của khối hiện tại và chèn các ứng viên hợp nhất không gian trích xuất được vào danh sách ứng viên hợp nhất. Ví dụ, các khối lân cận về không gian có thể bao gồm các khối lân cận ở góc đáy trái, các khối lân cận bên trái, các khối lân cận ở góc đỉnh phải, các khối lân cận ở góc đỉnh trái, và các khối lân cận ở góc đỉnh phải của khối hiện tại. Tuy nhiên, đây là một ví dụ, và ngoài các khối lân cận về không gian được mô tả trên đây, thì các khối lân cận bổ sung, chẳng hạn khối lân cận bên phải, khối lân cận dưới đáy, và khối lân cận đáy phải có thể còn được sử dụng làm các khối lân cận về không gian. Thiết bị lập mã có thể dò các khối khả dụng bằng cách tìm kiếm các khối lân cận về không gian dựa trên độ ưu tiên, và có thể trích xuất thông tin chuyển động trên các khối dò được làm các ứng viên hợp nhất không gian. Ví dụ, thiết bị mã hoá hoặc thiết bị giải mã có thể được tạo cấu hình để tuần tự tìm kiếm năm khối được thể hiện trên Fig.4 theo thứ tự chẳng hạn $A_1 \rightarrow B_1 \rightarrow B_0 \rightarrow A_0 \rightarrow B_2$, và có thể tuần tự đánh chỉ số các ứng viên khả dụng để cấu thành danh sách ứng viên hợp nhất.

Thiết bị lập mã có thể tìm kiếm khối lân cận về thời gian của khối hiện tại và chèn ứng viên hợp nhất thời gian trích xuất được vào danh sách ứng viên hợp nhất. Khối

lân cận về thời gian có thể được đặt tại ảnh tham chiếu mà là ảnh khác với ảnh hiện tại mà khối hiện tại được đặt trong đó. Ảnh tham chiếu mà các khối lân cận về thời gian được đặt trong đó có thể được gọi là ảnh được đặt cùng vị trí hoặc ảnh cùng vị trí. Các khối lân cận về thời gian có thể được tìm kiếm theo thứ tự là khối lân cận ở góc đáy phải và khối trung tâm ở đáy phải của khối được đặt cùng vị trí so với khối hiện tại trên ảnh cùng vị trí. Trong khi đó, khi việc nén dữ liệu chuyển động được áp dụng, thì thông tin chuyển động cụ thể có thể được lưu giữ dưới dạng thông tin chuyển động đại diện trên mỗi đơn vị lưu trữ được định trước ở ảnh cùng vị trí. Trong trường hợp này, thì không cần phải lưu giữ thông tin chuyển động trên tất cả các khối trong đơn vị lưu trữ được định trước, và thông qua việc này, thì có thể thu được tác dụng nén dữ liệu chuyển động. Trong trường hợp này, thì đơn vị lưu trữ được định trước có thể được xác định trước dưới dạng, ví dụ, các đơn vị là 16x16 mẫu hoặc các đơn vị là 8x8 mẫu, hoặc thông tin kích thước trên đơn vị lưu trữ được định trước có thể được báo hiệu từ thiết bị mã hoá đến thiết bị giải mã. Khi việc nén dữ liệu chuyển động được áp dụng, thì thông tin chuyển động trên các khối lân cận về thời gian có thể được thay thế bằng thông tin chuyển động đại diện trên đơn vị lưu trữ được định trước mà các khối lân cận về thời gian được đặt trong đó. Tức là, trong trường hợp này, từ góc độ thực hiện, thay vì khối được dự đoán mà được đặt tại các tọa độ của các khối lân cận về thời gian, thì ứng viên hợp nhất thời gian có thể được trích xuất dựa trên thông tin chuyển động trên khối dự đoán mà bao phủ vị trí được dịch trái số học sau khi dịch phải số học một giá trị nhất định dựa trên các tọa độ (vị trí mẫu đỉnh trái) của khối lân cận về thời gian. Ví dụ, khi đơn vị lưu trữ được định trước là các đơn vị 2nx2n mẫu, nếu các tọa độ của các khối lân cận về thời gian là (xTnb, yTnb), thì thông tin chuyển động trên khối dự đoán được đặt tại vị trí được hiệu chỉnh ((xTnb>>n)<<n), (yTnb>>n)<<n)) có thể được sử dụng cho ứng viên hợp nhất thời gian. Cụ thể là, khi đơn vị lưu trữ được định trước là các đơn vị 16x16 mẫu, nếu các tọa độ của các khối lân cận về thời gian là (xTnb, yTnb), thì thông tin chuyển động trên khối dự đoán được đặt tại vị trí được hiệu chỉnh

$((x_{Tnb} > 4) < 4), (y_{Tnb} > 4) < 4))$ có thể được sử dụng cho ứng viên hợp nhất thời gian. Theo cách khác, khi đơn vị lưu trữ được định trước là các đơn vị 8x8 mẫu, nếu các tọa độ của các khối lân cận về thời gian là (x_{Tnb}, y_{Tnb}) , thì thông tin chuyển động trên khối dự đoán được đặt tại vị trí được hiệu chỉnh $((x_{Tnb} > 3) < 3), (y_{Tnb} > 3) < 3))$ có thể được sử dụng cho ứng viên hợp nhất thời gian.

Thiết bị lập mã có thể kiểm tra xem số lượng ứng viên hợp nhất hiện tại có nhỏ hơn số lượng ứng viên hợp nhất tối đa hay không. Số lượng ứng viên hợp nhất tối đa có thể được định trước hoặc được báo hiệu từ thiết bị mã hoá đến thiết bị giải mã. Ví dụ, thiết bị mã hoá có thể tạo ra và mã hoá thông tin về số lượng ứng viên hợp nhất tối đa, và truyền thông tin này đến bộ giải mã dưới dạng luồng bit. Khi số lượng ứng viên hợp nhất tối đa đã được đạt đến, thì tiến trình bổ sung ứng viên sau đó có thể không tiến hành.

Theo kết quả của việc kiểm tra, khi số lượng ứng viên hợp nhất hiện tại là nhỏ hơn số lượng ứng viên hợp nhất tối đa, thì thiết bị lập mã có thể chèn ứng viên hợp nhất bổ sung vào danh sách ứng viên hợp nhất. Ví dụ, các ứng viên hợp nhất bổ sung có thể bao gồm ít nhất một trong số (các) ứng viên hợp nhất dựa trên lịch sử, (các) ứng viên hợp nhất trung bình theo cặp, ATMVP, ứng viên hợp nhất lưỡng dự đoán kết hợp (khi loại lát cắt/nhóm ô của lát cắt/nhóm ô hiện tại là loại B) và/hoặc ứng viên hợp nhất vectơ không, mà sẽ được mô tả sau.

Theo kết quả của việc kiểm tra, khi số lượng ứng viên hợp nhất hiện tại không nhỏ hơn số lượng ứng viên hợp nhất tối đa, thì thiết bị lập mã có thể chấm dứt việc xây dựng danh sách ứng viên hợp nhất. Trong trường hợp này, thiết bị mã hoá có thể chọn ứng viên hợp nhất tối ưu từ trong số các ứng viên hợp nhất mà cấu thành danh sách ứng viên hợp nhất dựa trên chi phí về tốc độ-độ méo (Rate-Distortion - RD), và báo hiệu thông tin chọn mà chỉ thị ứng viên hợp nhất được chọn (ví dụ, chỉ số hợp nhất) cho thiết bị giải mã. Thiết bị giải mã có thể chọn ứng viên hợp nhất tối ưu dựa trên danh sách ứng viên hợp nhất và thông tin chọn này.

Như đã mô tả trên đây, thông tin chuyển động trên ứng viên hợp nhất được chọn có thể được sử dụng làm thông tin chuyển động trên khối hiện tại, và các mẫu dự đoán của khối hiện tại có thể được trích xuất dựa trên thông tin chuyển động trên khối hiện tại. Thiết bị mã hoá có thể trích xuất các mẫu dư của khối hiện tại dựa trên các mẫu dự đoán, và có thể báo hiệu thông tin dư trên các mẫu dư cho thiết bị giải mã. Như đã mô tả trên đây, thiết bị giải mã có thể tạo ra các mẫu được tái tạo dựa trên các mẫu dư được trích xuất dựa trên thông tin dư và các mẫu dự đoán, và có thể tạo ra ảnh được tái tạo dựa trên đó.

Khi chế độ bỏ cóc được áp dụng, thì thông tin chuyển động trên khối hiện tại có thể được trích xuất theo cách giống như khi chế độ hợp nhất được áp dụng. Tuy nhiên, khi chế độ bỏ cóc được áp dụng, thì tín hiệu dư dành cho khối tương ứng là được lược bỏ, và do đó, các mẫu dự đoán có thể được sử dụng trực tiếp làm các mẫu được tái tạo. Chế độ bỏ cóc có thể được áp dụng, ví dụ, khi giá trị của phần tử cú pháp `cu_skip_flag` là 1.

Trong khi đó, ứng viên hợp nhất trung bình theo cặp có thể được gọi là ứng viên trung bình theo cặp hoặc ứng viên theo cặp. (Các) ứng viên trung bình theo cặp có thể được tạo ra bằng cách lấy trung bình các cặp ứng viên được định trước trong danh sách ứng viên hợp nhất hiện có. Ngoài ra, các cặp được định trước có thể được xác định là $\{(0, 1), (0, 2), (1, 2), (0, 3), (1, 3), (2, 3)\}$. Ở đây, các con số này có thể chỉ thị các chỉ số hợp nhất dành cho danh sách ứng viên hợp nhất. Vector chuyển động được lấy trung bình có thể được tính toán riêng biệt đối với mỗi danh sách tham chiếu. Ví dụ, khi hai vector chuyển động khả dụng trong một danh sách, thì hai vector chuyển động đó có thể được lấy trung bình ngay cả khi chúng trỏ đến các ảnh tham chiếu khác nhau. Ví dụ, khi chỉ một vector chuyển động khả dụng, thì một vector chuyển động có thể được sử dụng trực tiếp. Ví dụ, khi không có vector chuyển động nào khả dụng, thì danh sách đó có thể là không hợp lệ.

Ví dụ, khi danh sách ứng viên hợp nhất không đầy ngay cả sau khi các ứng viên

hợp nhất trung bình theo cặp được bổ sung, tức là, khi số lượng ứng viên hợp nhất hiện tại trong danh sách ứng viên hợp nhất là nhỏ hơn số lượng ứng viên hợp nhất tối đa, thì vector không (MVP không) có thể được chen cuối cùng cho đến khi đạt tới số lượng ứng viên hợp nhất tối đa. Tức là, vector không có thể được chen vào cho đến khi số lượng ứng viên hợp nhất hiện tại trong danh sách ứng viên hợp nhất trở thành số lượng ứng viên hợp nhất tối đa.

Trong khi đó, theo cách thông thường, thì chỉ một vector chuyển động có thể được dùng để biểu thị sự chuyển động của khối lập mã. Tức là, mô hình chuyển động tịnh tiến có thể được sử dụng. Tuy nhiên, mặc dù phương pháp này có thể biểu diễn sự chuyển động tối ưu theo các đơn vị là các khối, nhưng nó không thực sự là sự chuyển động tối ưu của mỗi mẫu, và hiệu quả lập mã có thể được tăng lên nếu vector chuyển động tối ưu có thể được xác định theo các đơn vị là các mẫu. Về mặt này, thì mô hình chuyển động afin có thể được sử dụng. Phương pháp dự đoán chuyển động afin để lập mã nhờ sử dụng mô hình chuyển động afin có thể là như sau.

Phương pháp dự đoán chuyển động afin này có thể biểu thị vector chuyển động theo các đơn vị của mỗi mẫu của khối nhờ sử dụng hai, ba, hoặc bốn vector chuyển động. Ví dụ, mô hình chuyển động afin này có thể biểu diễn bốn kiểu chuyển động. Mô hình chuyển động afin mà biểu diễn ba kiểu chuyển động (tịnh tiến, thay đổi tỷ lệ, và quay) trong số các kiểu chuyển động mà mô hình chuyển động afin có thể biểu diễn thì có thể được gọi là mô hình chuyển động afin tương tự (hoặc giản lược). Tuy nhiên, mô hình chuyển động afin không chỉ giới hạn ở mô hình chuyển động được mô tả trên đây.

Fig.5A và Fig.5B là các hình vẽ thể hiện sơ đồ ví dụ CPMV dành cho việc dự đoán chuyển động afin.

Việc dự đoán chuyển động afin có thể xác định vector chuyển động của vị trí mẫu được bao gồm trong khối nhờ sử dụng hai hoặc nhiều vector chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV). Trong trường hợp này, thì tập hợp vector chuyển động có thể được gọi là trường vector chuyển động (Motion Vector Field

- MVF) afin.

Ví dụ, Fig.5A có thể thể hiện trường hợp mà trong đó hai CPMV được sử dụng, mà có thể được gọi là mô hình afin 4 thông số. Trong trường hợp này, thì vector chuyển động tại vị trí mẫu (x, y) có thể được xác định dưới dạng, ví dụ, Phương trình (1).

Phương trình 1

$$\begin{cases} mv_x = \frac{mv_{1x}-mv_{0x}}{W}x + \frac{mv_{1y}-mv_{0y}}{W}y + mv_{0x} \\ mv_y = \frac{mv_{1y}-mv_{0y}}{W}x + \frac{mv_{1x}-mv_{0x}}{W}y + mv_{0y} \end{cases}$$

Ví dụ, Fig.5B có thể thể hiện trường hợp mà trong đó ba CPMV được sử dụng, mà có thể được gọi là mô hình afin 6 thông số. Trong trường hợp này, thì vector chuyển động tại vị trí mẫu (x, y) có thể được xác định dưới dạng, ví dụ, Phương trình (2).

Phương trình 2

$$\begin{cases} mv_x = \frac{mv_{1x}-mv_{0x}}{W}x + \frac{mv_{2x}-mv_{0x}}{H}y + mv_{0x} \\ mv_y = \frac{mv_{1y}-mv_{0y}}{W}x + \frac{mv_{2y}-mv_{0y}}{H}y + mv_{0y} \end{cases}$$

Ở các phương trình 1 và 2, thì $\{v_x, v_y\}$ có thể biểu thị vector chuyển động tại vị trí (x, y). Ngoài ra, $\{v_{0x}, v_{0y}\}$ có thể biểu thị CPMV của điểm kiểm soát (Control Point - CP) tại vị trí góc đỉnh trái của khối lập mã, và $\{v_{1x}, v_{1y}\}$ có thể biểu thị CPMV của CP tại vị trí góc đỉnh phải, $\{v_{2x}, v_{2y}\}$ có thể biểu thị CPMV của CP tại vị trí góc đáy trái. Ngoài ra, W có thể biểu thị chiều rộng của khối hiện tại, và H có thể biểu thị chiều cao của khối hiện tại.

Fig.6 là hình vẽ thể hiện sơ đồ ví dụ trường hợp mà trong đó MVF afin được xác định theo đơn vị là các khối con.

Trong tiến trình mã hoá/giải mã, thì MVF afin có thể được xác định theo các đơn vị là các mẫu hoặc theo các đơn vị là các khối con được xác định trước đó. Ví dụ, khi xác định theo các đơn vị là các mẫu, thì vector chuyển động có thể được thu thập dựa trên mỗi giá trị mẫu. Theo cách khác, ví dụ, khi xác định theo các đơn vị là các khối con, thì vector chuyển động của khối tương ứng có thể được thu thập dựa trên giá trị

mẫu của tâm của khối con (tức là, đáy phải của tâm, tức là, mẫu ở đáy phải trong số bốn mẫu ở tâm). Tức là, trong quá trình dự đoán chuyển động afin, thì vector chuyển động của khối hiện tại có thể được trích xuất theo các đơn vị là các mẫu hoặc các khối con.

Trong trường hợp trên Fig.6, thì MVF afin là được xác định theo đơn vị 4x4 khối con, nhưng kích thước của các khối con có thể được cải biến theo những cách khác nhau.

Tức là, khi việc dự đoán afin là khả dụng, thì ba mô hình chuyển động áp dụng được cho khối hiện tại là có thể bao gồm mô hình chuyển động tịnh tiến, mô hình chuyển động afin 4 thông số, và mô hình chuyển động afin 6 thông số. Ở đây, mô hình chuyển động tịnh tiến có thể biểu thị mô hình mà trong đó vector chuyển động của đơn vị khối hiện có được sử dụng, mô hình chuyển động afin 4 thông số có thể biểu thị mô hình mà trong đó hai CPMV được sử dụng, và mô hình chuyển động afin 6 thông số có thể biểu thị mô hình mà trong đó ba CPMV được sử dụng.

Trong khi đó, việc dự đoán chuyển động afin có thể bao gồm chế độ MVP afin (hoặc afin liên) hoặc chế độ hợp nhất afin.

Fig.7 là hình vẽ thể hiện sơ đồ mô tả chế độ hợp nhất afin trong quá trình liên dự đoán.

Ví dụ, trong chế độ hợp nhất afin, thì CPMV có thể được xác định theo mô hình chuyển động afin của khối lân cận được lập mã bởi việc dự đoán chuyển động afin. Ví dụ, các khối lân cận mà được lập mã dưới dạng dự đoán chuyển động afin theo thứ tự tìm kiếm là có thể được sử dụng cho chế độ hợp nhất afin. Tức là, khi ít nhất một trong số các khối lân cận được lập mã trong quá trình dự đoán chuyển động afin, thì khối hiện tại có thể được lập mã trong chế độ hợp nhất afin. Ở đây, chế độ hợp nhất afin có thể được gọi là AF_MERGE.

Khi chế độ hợp nhất afin được áp dụng, thì các CPMV của khối hiện tại có thể được trích xuất nhờ sử dụng các CPMV của các khối lân cận. Trong trường hợp này, các CPMV của khối lân cận có thể được sử dụng nguyên làm các CPMV của khối hiện

tại, và các CPMV của khối lân cận có thể được cải biến dựa trên kích thước của khối lân cận và kích thước của khối hiện tại và được sử dụng làm các CPMV của khối hiện tại.

Mặt khác, trong trường hợp chế độ hợp nhất afin mà trong đó vector chuyển động (Motion Vector - MV) được trích xuất theo các đơn vị là các khối con, thì nó có thể được gọi là chế độ hợp nhất khối con, mà có thể được chỉ thị dựa trên cờ hợp nhất khối con (hoặc phần tử cú pháp `merge_subblock_flag`). Theo cách khác, khi giá trị của phần tử cú pháp `merge_subblock_flag` là 1, thì nó có thể chỉ thị rằng chế độ hợp nhất khối con được áp dụng. Trong trường hợp này, danh sách ứng viên hợp nhất afin, mà sẽ được mô tả sau, có thể được gọi là danh sách ứng viên hợp nhất khối con. Trong trường hợp này, danh sách ứng viên hợp nhất khối con có thể còn bao gồm ứng viên được trích xuất bởi SbTMVP, mà sẽ được mô tả sau. Trong trường hợp này, ứng viên được trích xuất bởi SbTMVP có thể được sử dụng làm ứng viên của chỉ số 0 của danh sách ứng viên hợp nhất khối con. Nói cách khác, ứng viên mà được trích xuất từ SbTMVP là có thể được đặt trước ứng viên afin được kế thừa hoặc ứng viên afin được xây dựng, mà sẽ được mô tả sau, trong danh sách ứng viên hợp nhất khối con.

Khi chế độ hợp nhất afin được áp dụng, thì danh sách ứng viên hợp nhất afin có thể được xây dựng để trích xuất các CPMV cho khối hiện tại. Ví dụ, danh sách ứng viên hợp nhất afin có thể bao gồm ít nhất một trong số các ứng viên sau đây. 1) Ứng viên hợp nhất afin được kế thừa. 2) Ứng viên hợp nhất afin được xây dựng. 3) Ứng viên vector chuyển động không (hoặc vector không). Ở đây, ứng viên hợp nhất afin được kế thừa là ứng viên được trích xuất dựa trên các CPMV của khối lân cận khi khối lân cận được lập mã trong chế độ afin, ứng viên hợp nhất afin được xây dựng là ứng viên được trích xuất bằng cách xây dựng các CPMV dựa trên các MV của các khối lân cận của CP tương ứng theo các đơn vị của mỗi CPMV, và ứng viên vector chuyển động không có thể chỉ thị ứng viên bao gồm các CPMV mà có giá trị là 0.

Ví dụ, danh sách ứng viên hợp nhất afin có thể được xây dựng như sau.

Có thể có tới hai ứng viên afin được kế thừa, và các ứng viên afin được kế thừa này có thể được trích xuất từ các mô hình chuyển động afin của các khối lân cận. Các khối lân cận có thể chứa một khối lân cận bên trái và khối lân cận phía trên. Các khối ứng viên có thể được đặt như được thể hiện trên Fig.4. Thứ tự quét đối với bộ dự đoán bên trái có thể là $A_1 \rightarrow A_0$, và thứ tự quét đối với bộ dự đoán phía trên có thể là $B_1 \rightarrow B_0 \rightarrow B_2$. Chỉ một ứng viên được kế thừa từ mỗi trong số bên trái và trên đỉnh là có thể được chọn. Việc kiểm tra lược bớt có thể không được thực hiện giữa hai ứng viên được kế thừa.

Khi khối afin lân cận được kiểm tra, thì các vector chuyển động của điểm kiểm soát của khối được kiểm tra có thể được dùng để trích xuất ứng viên CPMVP trong danh sách hợp nhất afin của khối hiện tại. Ở đây, khối afin lân cận có thể chỉ thị khối được lập mã trong chế độ dự đoán afin trong số các khối lân cận của khối hiện tại. Ví dụ, như được thể hiện trên Fig.7, khi khối lân cận A ở đáy trái được lập mã theo chế độ dự đoán afin, thì các vector chuyển động v_2 , v_3 , và v_4 của góc đỉnh trái (top-left), góc đỉnh phải, và góc đáy trái của khối lân cận A này có thể được thu thập. Khi khối lân cận A này được lập mã với mô hình chuyển động afin 4 thông số, thì hai CPMV của khối hiện tại có thể được tính toán theo v_2 và v_3 . Khi khối lân cận A này được lập mã với mô hình chuyển động afin 6 thông số, thì hai CPMV của khối hiện tại có thể được tính toán theo v_2 , v_3 , và v_4 .

Fig.8 là hình vẽ thể hiện sơ đồ mô tả các vị trí của các ứng viên trong chế độ hợp nhất afin.

Ứng viên afin được xây dựng có thể có nghĩa là ứng viên được xây dựng bằng cách kết hợp thông tin chuyển động tịnh tiến xung quanh mỗi điểm kiểm soát. Thông tin chuyển động trên các điểm kiểm soát có thể được trích xuất từ các đường bao không gian và thời gian được quy định. $CPMV_k$ ($k=0, 1, 2, 3$) có thể biểu thị điểm kiểm soát thứ k .

Như được thể hiện trên Fig.8, các khối có thể được kiểm tra theo thứ tự là

$B_2 \rightarrow B_3 \rightarrow A_2$ đối với CPMV0, và vector chuyển động của khối khả dụng thứ nhất có thể được sử dụng. Đối với CPMV1 thì các khối có thể được kiểm tra theo thứ tự là $B_1 \rightarrow B_0$, và đối với CPMV2 thì các khối có thể được kiểm tra theo thứ tự là $A_1 \rightarrow A_0$. Bộ dự đoán vector chuyển động theo thời gian (Temporal Motion Vector Predictor - TVMP) có thể được sử dụng với CPMV3 nếu khả dụng.

Sau khi thu được các vector chuyển động của bốn điểm kiểm soát, thì các ứng viên hợp nhất affin có thể được tạo ra dựa trên thông tin chuyển động thu thập được. Sự kết hợp của các vector chuyển động của điểm kiểm soát có thể tương ứng với bất kỳ trong số $\{\text{CPMV0}, \text{CPMV1}, \text{CPMV2}\}$, $\{\text{CPMV0}, \text{CPMV1}, \text{CPMV3}\}$, $\{\text{CPMV0}, \text{CPMV2}, \text{CPMV3}\}$, $\{\text{CPMV1}, \text{CPMV2}, \text{CPMV3}\}$, $\{\text{CPMV0}, \text{CPMV1}\}$, và $\{\text{CPMV0}, \text{CPMV2}\}$.

Sự kết hợp của ba CPMV có thể cấu thành ứng viên hợp nhất affin 6 thông số, và sự kết hợp của hai CPMV có thể cấu thành ứng viên hợp nhất affin 4 thông số. Để tránh tiến trình thay đổi tỷ lệ chuyển động, khi các chỉ số tham chiếu của các điểm kiểm soát là khác nhau, thì những sự kết hợp liên quan của các vector chuyển động của điểm kiểm soát có thể được loại bỏ.

Fig.9 là hình vẽ thể hiện sơ đồ mô tả SbTMVP trong quá trình liên dự đoán.

Trong khi đó, phương pháp dự đoán vector chuyển động theo thời gian dựa trên khối con (Subblock-based Temporal Motion Vector Prediction - SbTMVP) cũng có thể được sử dụng. Ví dụ, SbTMVP có thể được gọi là dự đoán vector chuyển động theo thời gian nâng cao (Advanced Temporal Motion Vector Prediction - ATMVP). SbTMVP có thể sử dụng trường chuyển động trong ảnh được đặt cùng vị trí để cải thiện việc dự đoán vector chuyển động và chế độ hợp nhất đối với các CU trong ảnh hiện tại. Ở đây, ảnh được đặt cùng vị trí có thể được gọi là ảnh cùng vị trí.

Ví dụ, SbTMVP có thể dự đoán sự chuyển động ở mức độ khối con (hoặc CU con). Ngoài ra, SbTMVP có thể áp dụng phép dịch chuyển động trước khi tìm nạp thông tin chuyển động theo thời gian từ ảnh cùng vị trí. Ở đây, lượng dịch chuyển động này

có thể được thu thập từ vector chuyển động của một trong số các khối lân cận về không gian của khối hiện tại.

SbTMVP có thể dự đoán vector chuyển động của khối con (hoặc CU con) trong khối (hoặc CU) hiện tại theo hai bước.

Ở bước thứ nhất, thì các khối lân cận về không gian có thể được kiểm tra theo thứ tự là A_1 , B_1 , B_0 và A_0 trên Fig.4. Khối lân cận về không gian thứ nhất mà có vector chuyển động sử dụng ảnh cùng vị trí làm ảnh tham chiếu của nó có thể được kiểm tra, và vector chuyển động này có thể được chọn làm lượng dịch chuyển động để áp dụng. Khi sự chuyển động đó không được kiểm tra từ các khối lân cận về không gian, thì lượng dịch chuyển động này có thể được thiết đặt bằng $(0, 0)$.

Ở bước thứ hai, thì phép dịch chuyển động mà được kiểm tra ở bước thứ nhất có thể được áp dụng để thu được thông tin chuyển động ở mức độ khối con (vector chuyển động và các chỉ số tham chiếu) từ ảnh cùng vị trí. Ví dụ, lượng dịch chuyển động này có thể được cộng vào các tọa độ của khối hiện tại. Ví dụ, sự dịch chuyển động này có thể được thiết đặt bằng sự chuyển động của A_1 trên Fig.4. Trong trường hợp này, đối với mỗi khối con, thì thông tin chuyển động trên khối tương ứng trong ảnh cùng vị trí có thể được dùng để trích xuất thông tin chuyển động trên khối con. Việc thay đổi tỷ lệ chuyển động theo thời gian có thể được áp dụng để đồng chỉnh các ảnh tham chiếu của các vector chuyển động về thời gian với các ảnh tham chiếu của khối hiện tại.

Danh sách hợp nhất dựa trên khối con được kết hợp mà bao gồm cả các ứng viên SbTMVP và các ứng viên hợp nhất afin có thể được dùng để báo hiệu chế độ hợp nhất afin. Ở đây, chế độ hợp nhất afin có thể được gọi là chế độ hợp nhất dựa trên khối con. Chế độ SbTMVP có thể khả dụng hoặc không khả dụng theo cờ được bao gồm trong tập hợp thông số chuỗi (Sequence Parameter Set - SPS). Khi chế độ SbTMVP là khả dụng, thì bộ dự đoán SbTMVP có thể được bổ sung dưới dạng mục nhập đầu tiên của danh sách ứng viên hợp nhất dựa trên khối con, và các ứng viên hợp nhất afin có thể theo sau. Kích thước tối đa cho phép của danh sách ứng viên hợp nhất afin có thể là

năm.

Kích thước của CU con (hoặc khối con) được sử dụng trong SbTMVP có thể được cố định bằng 8×8 , và như trong chế độ hợp nhất afin, chế độ SbTMVP có thể được áp dụng chỉ cho các khối có cả chiều rộng và chiều cao lớn hơn hoặc bằng 8. Logic mã hoá của ứng viên hợp nhất SbTMVP bổ sung có thể giống như của các ứng viên hợp nhất khác. Tức là, đối với mỗi CU trong lát cắt P hoặc lát cắt B, thì việc kiểm tra RD nhờ sử dụng chi phí về tốc độ-độ méo (Rate-Distortion - RD) bổ sung có thể được thực hiện để xác định xem có sử dụng ứng viên SbTMVP hay không.

Trong khi đó, khối được dự đoán đối với khối hiện tại có thể được trích xuất dựa trên thông tin chuyển động được trích xuất theo chế độ dự đoán. Khối được dự đoán có thể bao gồm các mẫu dự đoán (mảng mẫu dự đoán) của khối hiện tại. Khi vector chuyển động của khối hiện tại chỉ thị đơn vị phân đoạn mẫu, thì thủ tục nội suy có thể được thực hiện. Thông qua việc này, thì các mẫu dự đoán của khối hiện tại có thể được trích xuất dựa trên các mẫu tham chiếu của đơn vị phân đoạn mẫu trong ảnh tham chiếu. Khi việc liên dự đoán afin (chế độ dự đoán afin) được áp dụng cho khối hiện tại, thì các mẫu dự đoán có thể được tạo ra dựa trên mẫu/đơn vị khối con MV. Khi việc lưỡng dự đoán được áp dụng, thì các mẫu dự đoán có thể được sử dụng làm các mẫu dự đoán của khối hiện tại được trích xuất thông qua tổng được lấy trọng số hoặc trung bình được lấy trọng số (theo pha) của các mẫu dự đoán được trích xuất dựa trên việc dự đoán L0 (tức là, việc dự đoán sử dụng ảnh tham chiếu và MVL0 trong danh sách ảnh tham chiếu L0) và các mẫu dự đoán được trích xuất dựa trên việc dự đoán L1 (tức là, việc dự đoán sử dụng ảnh tham chiếu và MVL1 trong danh sách ảnh tham chiếu L1). Ở đây, vector chuyển động theo chiều L0 có thể được gọi là vector chuyển động L0 hoặc MVL0, và vector chuyển động theo chiều L1 có thể được gọi là vector chuyển động L1 hoặc MVL1. Trong trường hợp việc lưỡng dự đoán được áp dụng, khi ảnh tham chiếu mà được sử dụng cho việc dự đoán L0 và ảnh tham chiếu mà được sử dụng cho việc dự đoán L1 là được đặt theo các chiều thời gian khác nhau đối với ảnh hiện tại (tức là, trường hợp

tương ứng với việc sự đoán hai chiều hoặc lưỡng dự đoán), thì trường hợp đó có thể được gọi là lưỡng dự đoán thật.

Ngoài ra, như đã mô tả trên đây, các mẫu được tái tạo và các ảnh được tái tạo có thể được tạo ra dựa trên các mẫu dự đoán trích xuất được, và sau đó, các thủ tục chẳng hạn như lọc trong vòng lặp có thể được thực hiện.

Trong khi đó, khi việc lưỡng dự đoán được áp dụng cho khôi hiện tại, thì các mẫu dự đoán có thể được trích xuất dựa trên trung bình được lấy trọng số. Ví dụ, việc lưỡng dự đoán sử dụng trung bình được lấy trọng số có thể được gọi là lưỡng dự đoán với trọng số mức độ CU (Bi-prediction with CU-level Weight - BCW), lưỡng dự đoán với trung bình được lấy trọng số (Bi-prediction With weighted Average - BWA), hoặc lưỡng dự đoán lấy trung bình có trọng số.

Theo cách thông thường, thì tín hiệu lưỡng dự đoán (tức là các mẫu lưỡng dự đoán) có thể được trích xuất thông qua trung bình đơn giản của L0 tín hiệu dự đoán (L0 mẫu dự đoán) và L1 tín hiệu dự đoán. Tức là, các mẫu lưỡng dự đoán là được trích xuất dưới dạng trung bình của L0 mẫu dự đoán dựa trên L0 ảnh tham chiếu và MVL0 và L1 mẫu dự đoán dựa trên L1 ảnh tham chiếu và MVL1. Tuy nhiên, khi việc lưỡng dự đoán được áp dụng, thì tín hiệu lưỡng dự đoán (các mẫu lưỡng dự đoán) có thể được trích xuất thông qua trung bình được lấy trọng số của L0 tín hiệu dự đoán và L1 tín hiệu dự đoán như sau. Ví dụ, các tín hiệu lưỡng dự đoán (các mẫu lưỡng dự đoán) có thể được trích xuất như trong Phương trình 3.

Phương trình 3

$$P_{\text{bi-pred}} = ((8 - w) * P_0 + w * P_1 + 4) \gg 3$$

Trong Phương trình 3, thì $P_{\text{bi-pred}}$ có thể biểu thị giá trị của tín hiệu lưỡng dự đoán, tức là, giá trị mẫu dự đoán được trích xuất bằng cách áp dụng việc lưỡng dự đoán, và w có thể biểu thị trọng số. Ngoài ra, P_0 có thể biểu thị giá trị của L0 tín hiệu dự đoán, tức là, giá trị mẫu dự đoán trích xuất được bằng cách áp dụng việc dự đoán L0, và P_1

có thể biểu thị giá trị của L1 tín hiệu dự đoán, tức là, giá trị mẫu dự đoán trích xuất được bằng cách áp dụng việc dự đoán L1.

Ví dụ, 5 trọng số có thể được phép trong quá trình lưỡng dự đoán trung bình được lấy trọng số. Ví dụ, năm trọng số w này có thể bao gồm -2, 3, 4, 5, hoặc 10. Tức là, trọng số w có thể được xác định dưới dạng một trong số các ứng viên trọng số bao gồm -2, 3, 4, 5, hoặc 10. Đối với mỗi CU mà việc lưỡng dự đoán được áp dụng vào đó, thì trọng số w có thể được xác định bởi một trong hai phương pháp. Theo phương pháp thứ nhất, thì chỉ số trọng số có thể được báo hiệu sau hiệu vector chuyển động dành cho CU không được hợp nhất. Theo phương pháp thứ hai, thì chỉ số trọng số dành cho CU được hợp nhất có thể được suy ra từ các khối lân cận dựa trên chỉ số ứng viên hợp nhất.

Ví dụ, việc lưỡng dự đoán trung bình được lấy trọng số có thể được áp dụng cho CU có nhiều hơn hoặc bằng 256 mẫu độ chói. Tức là, khi tích của chiều rộng và chiều cao của CU là lớn hơn hoặc bằng 256, thì việc lưỡng dự đoán trung bình được lấy trọng số có thể được áp dụng. Trong trường hợp ảnh độ trễ thấp (low-delay), thì năm trọng số có thể được sử dụng, và trong trường hợp không phải ảnh độ trễ thấp, thì ba trọng số có thể được sử dụng. Ví dụ, ba trọng số này có thể bao gồm 3, 4 hoặc 5.

Ví dụ, ở thiết bị mã hoá này, thì thuật toán tìm kiếm nhanh có thể được áp dụng để tìm chỉ số trọng số mà không làm tăng đáng kể độ phức tạp của thiết bị mã hoá. Thuật toán này có thể được nêu vắn tắt như sau. Ví dụ, khi ảnh hiện tại là ảnh độ trễ thấp khi được kết hợp với độ phân giải vector chuyển động thích ứng (Adaptive Motion Vector Resolution - AMVR) (khi AMVR được sử dụng làm chế độ liên dự đoán), thì các trọng số không bằng nhau có thể được kiểm tra có điều kiện đối với độ chính xác vector chuyển động 1 điểm ảnh và 4 điểm ảnh. Ví dụ, khi được kết hợp với afin (khi chế độ dự đoán afin được sử dụng làm chế độ liên dự đoán), trong trường hợp mà chế độ dự đoán afin được chọn hiện tại làm chế độ tốt nhất, thì việc ước lượng chuyển động (Motion Estimation - ME) afin có thể được thực hiện trên các trọng số không bằng nhau. Ví dụ, khi hai ảnh tham chiếu lưỡng dự đoán là giống nhau, thì các trọng số không bằng

nhau có thể được kiểm tra có điều kiện. Ví dụ, khi điều kiện cụ thể được thoả mãn tùy theo khoảng cách POC giữa ảnh hiện tại và ảnh tham chiếu, thì thông số lượng tử hoá (Quantization Parameter - QP) lập mã, và các trọng số cấp thời gian và không bằng nhau có thể không được tìm kiếm.

Ví dụ, chỉ số trọng số BCW có thể được lập mã nhờ sử dụng một cụm điểm ảnh được lập mã ngữ cảnh theo sau bởi cụm điểm ảnh được lập mã đi vòng. Cụm điểm ảnh được lập mã ngữ cảnh thứ nhất có thể chỉ thị xem trọng số giống nhau có được sử dụng hay không. Khi các trọng số không bằng nhau được sử dụng dựa trên cụm điểm ảnh được lập mã ngữ cảnh thứ nhất, thì các cụm điểm ảnh bổ sung có thể được báo hiệu nhờ sử dụng việc lập mã đi vòng để chỉ thị các trọng số không bằng nhau cần được sử dụng.

Trong khi đó, khi việc lưỡng dự đoán được áp dụng, thì thông tin trọng số mà được dùng để tạo ra các mẫu dự đoán là có thể được trích xuất dựa trên thông tin chỉ số trọng số trên ứng viên được chọn từ trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất.

Theo một phương án của sáng chế, khi xây dựng ứng viên vector chuyển động cho chế độ hợp nhất, thì thông tin chỉ số trọng số trên ứng viên vector chuyển động theo thời gian có thể được trích xuất như sau. Ví dụ, khi ứng viên vector chuyển động theo thời gian sử dụng việc lưỡng dự đoán, thì thông tin chỉ số trọng số trên trung bình được lấy trọng số có thể được trích xuất. Tức là, khi kiểu liên dự đoán là lưỡng dự đoán, thì thông tin chỉ số trọng số (hoặc ứng viên vector chuyển động theo thời gian) trên ứng viên hợp nhất thời gian trong danh sách ứng viên hợp nhất là có thể được trích xuất.

Ví dụ, thông tin chỉ số trọng số trên trung bình được lấy trọng số đối với ứng viên vector chuyển động theo thời gian có thể luôn luôn được trích xuất là 0. Ở đây, thông tin chỉ số trọng số trên 0 có thể có nghĩa là việc các trọng số của mỗi chiều tham chiếu (tức là chiều dự đoán L0 và chiều dự đoán L1 trong quá trình lưỡng dự đoán) là giống nhau. Ví dụ, thủ tục để trích xuất vector chuyển động của thành phần độ chói cho chế độ hợp nhất có thể được thể hiện ở Bảng 1 dưới đây.

Bảng 1

8.4.2.2 Derivation process for luma motion vectors for merge mode

This process is only invoked when `merge_flag[ xCb ][ yCb ]` is equal to 1, where (xCb, yCb) specify the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture.

Inputs to this process are:

- a luma location (xCb, yCb) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable `cbWidth` specifying the width of the current coding block in luma samples,
- a variable `cbHeight` specifying the height of the current coding block in luma samples.

Outputs of this process are:

- the luma motion vectors in 1/16 fractional-sample accuracy `mvL0[ 0 ][ 0 ]` and `mvL1[ 0 ][ 0 ]`,
- the reference indices `refIdxL0` and `refIdxL1`,
- the prediction list utilization flags `predFlagL0[ 0 ][ 0 ]` and `predFlagL1[ 0 ][ 0 ]`,
- the bi-prediction weight index `gbIdx`.

The bi-prediction weight index `gbIdx` is set equal to 0.

The motion vectors `mvL0[ 0 ][ 0 ]` and `mvL1[ 0 ][ 0 ]`, the reference indices `refIdxL0` and `refIdxL1` and the prediction utilization flags `predFlagL0[ 0 ][ 0 ]` and `predFlagL1[ 0 ][ 0 ]` are derived by the following ordered steps:

1. The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width `cbWidth`, and the luma coding block height `cbHeight` as inputs, and the output being the availability flags `availableFlagA0`, `availableFlagA1`, `availableFlagB0`, `availableFlagB1` and `availableFlagB2`, the reference indices `refIdxLXA0`, `refIdxLXA1`, `refIdxLXB0`, `refIdxLXB1` and `refIdxLXB2`, the prediction list utilization flags `predFlagLXA0`, `predFlagLXA1`, `predFlagLXB0`, `predFlagLXB1` and `predFlagLXB2`, and the motion vectors `mvLXA0`, `mvLXA1`, `mvLXB0`, `mvLXB1` and `mvLXB2`, with X being 0 or 1, and the bi-prediction weight indices `gbIdxA0`, `gbIdxA1`, `gbIdxB0`, `gbIdxB1`, `gbIdxB2`.
2. The reference indices, `refIdxLXCol`, with X being 0 or 1, and the bi-prediction weight index `gbIdxCol` for the temporal merging candidate Col are set equal to 0.

3. The derivation process for temporal luma motion vector prediction as specified in clause 8.4.2.11 is invoked with the luma location (xCb, yCb) , the luma coding block width `cbWidth`, the luma coding block height `cbHeight` and the variable `refIdxL0Col` as inputs, and the output being the availability flag `availableFlagL0Col` and the temporal motion vector `mvL0Col`. The variables `availableFlagCol`, `predFlagL0Col` and `predFlagL1Col` are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \quad (8-283)$$

$$\text{predFlagL0Col} = \text{availableFlagL0Col} \quad (8-284)$$

$$\text{predFlagL1Col} = 0 \quad (8-285)$$

$$\text{gbIdxCol} = 0$$

(8-xxx)

4. When `tile_group_type` is equal to B, the derivation process for temporal luma motion vector prediction as specified in clause 8.4.2.11 is invoked with the luma location (xCb, yCb) , the luma coding block width `cbWidth`, the luma coding block height `cbHeight` and the variable `refIdxL1Col` as inputs, and the output being the availability flag `availableFlagL1Col` and the temporal motion vector `mvL1Col`. The variables `availableFlagCol` and `predFlagL1Col` are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \mid \mid \text{availableFlagL1Col} \quad (8-286)$$

predFlagL1Col = availableFlagL1Col (8-287)

5. The merging candidate list, mergeCandList, is constructed as follows:

```

i = 0
if( availableFlagA1 )
    mergeCandList[ i++ ] = A1
if( availableFlagB1 )
    mergeCandList[ i++ ] = B1
if( availableFlagB0 )
    mergeCandList[ i++ ] = B0
if( availableFlagA0 )
    mergeCandList[ i++ ] = A0
if( availableFlagB2 )
    mergeCandList[ i++ ] = B2
if( availableFlagCol )
    mergeCandList[ i++ ] = Col

```

(8-288)

6. The variable numCurrMergeCand and numOrigMergeCand are set equal to the number of merging candidates in the mergeCandList.
7. When numCurrMergeCand is less than (MaxNumMergeCand - 1) and NumHmvpCand is greater than 0, the following applies:
- The derivation process of history-based merging candidates as specified in 8.4.2.6 is invoked with mergeCandList, and numCurrMergeCand as inputs, and modified mergeCandList and numCurrMergeCand as outputs.
 - numOrigMergeCand is set equal to numCurrMergeCand.
8. The derivation process for pairwise average merging candidates specified in clause 8.4.2.4 is invoked with mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList, numCurrMergeCand and numOrigMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0avgCand_k and refIdxL1avgCand_k, the prediction list utilization flags predFlagL0avgCand_k and predFlagL1avgCand_k and the motion vectors mvL0avgCand_k and mvL1avgCand_k of every new candidate avgCand_k being added into mergeCandList. The bi-prediction weight index gbIdx of every new candidate avgCand_k being added into mergeCandList is set equal to 0. The number of candidates being added, numAvgMergeCand, is set equal to (numCurrMergeCand - numOrigMergeCand). When numAvgMergeCand is greater than 0, k ranges from 0 to numAvgMergeCand - 1, inclusive.
9. The derivation process for zero motion vector merging candidates specified in clause 8.4.2.5 is invoked with the mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList and numCurrMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0zeroCand_m and refIdxL1zeroCand_m, the prediction list utilization flags predFlagL0zeroCand_m and predFlagL1zeroCand_m and the motion vectors mvL0zeroCand_m and mvL1zeroCand_m of every new candidate zeroCand_m being added into mergeCandList. The bi-prediction weight index gbIdx of every new candidate zeroCand_m being added into mergeCandList is set equal to 0. The number of candidates being added, numZeroMergeCand, is set equal to (numCurrMergeCand - numOrigMergeCand - numAvgMergeCand). When numZeroMergeCand is greater than 0, m ranges from 0 to numZeroMergeCand - 1, inclusive.
10. The variable mergeIdxOffset is set equal to 0.
11. When mmvd_flag[xCb][yCb] is equal to 1, the variable mmvdCnt is set equal to 0 and tthe following applies until mmvdCnt is greater than (merge_idx[xCb][yCb] + mergeIdxOffset) or mmvdCnt is equal to MaxNumMergeCand:
- When candidate mergeCandList[mmvdCnt] uses the current decoded picture as its reference picture, mergeIdxOffset is incremented by 1.
 - The variable mmvdCnt is incremented by 1.

12. The following assignments are made with N being the candidate at position $\text{merge_idx}[\text{xCb}][\text{yCb}] + \text{mergeIdxOffset}$ in the merging candidate list mergeCandList ($N = \text{mergeCandList}[\text{merge_idx}[\text{xCb}][\text{yCb}] + \text{mergeIdxOffset}]$) and X being replaced by 0 or 1:

$$\text{refIdxLX} = \text{refIdxLXN} \quad (8-289)$$

$$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-290)$$

$$\text{mvLX}[0][0][0] = \text{mvLXN}[0] \quad (8-291)$$

$$\text{mvLX}[0][0][1] = \text{mvLXN}[1] \quad (8-292)$$

$$\text{gbIdx} = \text{gbIdxN} \quad (8-293)$$

13. When $\text{mmvd_flag}[\text{xCb}][\text{yCb}]$ is equal to 1, the following applies:

- The derivation process for merge motion vector difference as specified in 8.4.2.7 is invoked with the luma location (xCb, yCb), the luma motion vectors $\text{mvL0}[0][0]$, $\text{mvL1}[0][0]$, the reference indices refIdxL0 , refIdxL1 and the prediction list utilization flags $\text{predFlagL0}[0][0]$ and $\text{predFlagL1}[0][0]$ as inputs, and the motion vector differences mMvdL0 and mMvdL1 as outputs.
- The motion vector difference mMvdLX is added to the merge motion vectors mvLX for X being 0 and 1 as follows:

$$\text{mvLX}[0][0][0] += \text{mMvdLX}[0] \quad (8-294)$$

$$\text{mvLX}[0][0][1] += \text{mMvdLX}[1] \quad (8-295)$$

Như được thể hiện trên Bảng 1, gbIdx có thể biểu thị chỉ số trọng số lưỡng dự đoán, và gbIdxCol có thể biểu thị chỉ số trọng số lưỡng dự đoán cho ứng viên hợp nhất thời gian (ví dụ, ứng viên vector chuyển động theo thời gian trong danh sách ứng viên hợp nhất). Trong thủ tục để trích xuất vector chuyển động của thành phần độ chói cho chế độ hợp nhất (mục lục 3 của 8.4.2.2), thì gbIdxCol có thể được trích xuất là 0. Tức là, chỉ số trọng số của ứng viên vector chuyển động theo thời gian có thể được trích xuất là 0.

Theo cách khác, chỉ số trọng số dành cho trung bình được lấy trọng số của các ứng viên vector chuyển động theo thời gian có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối được đặt cùng vị trí. Ở đây, khối được đặt cùng vị trí có thể được gọi là khối cùng vị trí, khối nằm cùng vị trí, hoặc khối tham chiếu được đặt cùng vị trí, và khối cùng vị trí có thể chỉ thị khối tại cùng vị trí với khối hiện tại trên ảnh tham chiếu. Ví dụ, thủ tục để trích xuất vector chuyển động của thành phần độ chói cho chế độ hợp nhất có thể như được thể hiện ở Bảng 2 dưới đây.

Bảng 2

8.4.2.2 Derivation process for luma motion vectors for merge mode

This process is only invoked when `merge_flag[ xCb ][ yCb ]` is equal to 1, where (xCb, yCb) specify the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture.

Inputs to this process are:

- a luma location (xCb, yCb) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable `cbWidth` specifying the width of the current coding block in luma samples,
- a variable `cbHeight` specifying the height of the current coding block in luma samples.

Outputs of this process are:

- the luma motion vectors in 1/16 fractional-sample accuracy `mvL0[ 0 ][ 0 ]` and `mvL1[ 0 ][ 0 ]`,
- the reference indices `refIdxL0` and `refIdxL1`,
- the prediction list utilization flags `predFlagL0[ 0 ][ 0 ]` and `predFlagL1[ 0 ][ 0 ]`,
- the bi-prediction weight index `gbIdx`.

The bi-prediction weight index `gbIdx` is set equal to 0.

The motion vectors `mvL0[ 0 ][ 0 ]` and `mvL1[ 0 ][ 0 ]`, the reference indices `refIdxL0` and `refIdxL1` and the prediction utilization flags `predFlagL0[ 0 ][ 0 ]` and `predFlagL1[ 0 ][ 0 ]` are derived by the following ordered steps:

1. The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width `cbWidth`, and the luma coding block height `cbHeight` as inputs, and the output being the availability flags `availableFlagA0`, `availableFlagA1`, `availableFlagB0`, `availableFlagB1` and `availableFlagB2`, the reference indices `refIdxLXA0`, `refIdxLXA1`, `refIdxLXB0`, `refIdxLXB1` and `refIdxLXB2`, the prediction list utilization flags `predFlagLXA0`, `predFlagLXA1`, `predFlagLXB0`, `predFlagLXB1` and `predFlagLXB2`, and the motion vectors `mvLXA0`, `mvLXA1`, `mvLXB0`, `mvLXB1` and `mvLXB2`, with X being 0 or 1, and the bi-prediction weight indices `gbIdxA0`, `gbIdxA1`, `gbIdxB0`, `gbIdxB1`, `gbIdxB2`.
2. The reference indices, `refIdxLXCol`, with X being 0 or 1, and the bi-prediction weight index `gbIdxCol` for the temporal merging candidate Col are set equal to 0.

3. The derivation process for temporal luma motion vector prediction as specified in in clause 8.4.2.11 is invoked with the luma location (xCb, yCb) , the luma coding block width `cbWidth`, the luma coding block height `cbHeight` and the variable `refIdxL0Col` as inputs, and the output being the availability flag `availableFlagL0Col` and the temporal motion vector `mvL0Col`. The variables `availableFlagCol`, `predFlagL0Col` and `predFlagL1Col` are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \quad (8-283)$$

$$\text{predFlagL0Col} = \text{availableFlagL0Col} \quad (8-284)$$

$$\text{predFlagL1Col} = 0 \quad (8-285)$$

$$\text{gbIdxCol} = 0 \quad (8-xxx)$$

4. When `tile_group_type` is equal to B, the derivation process for temporal luma motion vector prediction as specified in clause 8.4.2.11 is invoked with the luma location (xCb, yCb) , the luma coding block width `cbWidth`, the luma coding block height `cbHeight` and the variable `refIdxL1Col` as inputs, and the output being the availability flag `availableFlagL1Col` and the temporal motion vector `mvL1Col`. The variables `availableFlagCol` and `predFlagL1Col` are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \mid \mid \text{availableFlagL1Col} \quad (8-286)$$

predFlagL1Col = availableFlagL1Col (8-287)

gbIdxCol = gbIdxCol (x-xxx)

5. The merging candidate list, mergeCandList, is constructed as follows:

```

i = 0
if( availableFlagA1 )
    mergeCandList[ i++ ] = A1
if( availableFlagB1 )
    mergeCandList[ i++ ] = B1
if( availableFlagB0 )
    mergeCandList[ i++ ] = B0
if( availableFlagA0 )
    mergeCandList[ i++ ] = A0
if( availableFlagB2 )
    mergeCandList[ i++ ] = B2
if( availableFlagCol )
    mergeCandList[ i++ ] = Col

```

(8-288)

6. The variable numCurrMergeCand and numOrigMergeCand are set equal to the number of merging candidates in the mergeCandList.
7. When numCurrMergeCand is less than (MaxNumMergeCand - 1) and NumHmvpCand is greater than 0, the following applies:
- The derivation process of history-based merging candidates as specified in 8.4.2.6 is invoked with mergeCandList, and numCurrMergeCand as inputs, and modified mergeCandList and numCurrMergeCand as outputs.
 - numOrigMergeCand is set equal to numCurrMergeCand.
8. The derivation process for pairwise average merging candidates specified in clause 8.4.2.4 is invoked with mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList, numCurrMergeCand and numOrigMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0avgCand_k and refIdxL1avgCand_k, the prediction list utilization flags predFlagL0avgCand_k and predFlagL1avgCand_k and the motion vectors mvL0avgCand_k and mvL1avgCand_k of every new candidate avgCand_k being added into mergeCandList. The bi-prediction weight index gbIdx of every new candidate avgCand_k being added into mergeCandList is set equal to 0. The number of candidates being added, numAvgMergeCand, is set equal to (numCurrMergeCand - numOrigMergeCand). When numAvgMergeCand is greater than 0, k ranges from 0 to numAvgMergeCand - 1, inclusive.
9. The derivation process for zero motion vector merging candidates specified in clause 8.4.2.5 is invoked with the mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList and numCurrMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0zeroCand_m and refIdxL1zeroCand_m, the prediction list utilization flags predFlagL0zeroCand_m and predFlagL1zeroCand_m and the motion vectors mvL0zeroCand_m and mvL1zeroCand_m of every new candidate zeroCand_m being added into mergeCandList. The bi-prediction weight index gbIdx of every new candidate zeroCand_m being added into mergeCandList is set equal to 0. The number of candidates being added, numZeroMergeCand, is set equal to (numCurrMergeCand - numOrigMergeCand - numAvgMergeCand). When numZeroMergeCand is greater than 0, m ranges from 0 to numZeroMergeCand - 1, inclusive.
10. The variable mergeIdxOffset is set equal to 0.
11. When mmvd_flag[xCb][yCb] is equal to 1, the variable mmvdCnt is set equal to 0 and the following applies until mmvdCnt is greater than (merge_idx[xCb][yCb] + mergeIdxOffset) or mmvdCnt is equal to MaxNumMergeCand:

- When candidate $\text{mergeCandList}[\text{mmvdCnt}]$ uses the current decoded picture as its reference picture, mergeIdxOffset is incremented by 1.
 - The variable mmvdCnt is incremented by 1.
12. The following assignments are made with N being the candidate at position $\text{merge_idx}[\text{xCb}][\text{yCb}] + \text{mergeIdxOffset}$ in the merging candidate list mergeCandList ($N = \text{mergeCandList}[\text{merge_idx}[\text{xCb}][\text{yCb}] + \text{mergeIdxOffset}]$) and X being replaced by 0 or 1:
- $$\text{refIdxLX} = \text{refIdxLXN} \quad (8-289)$$
- $$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-290)$$
- $$\text{mvLX}[0][0][0] = \text{mvLXN}[0] \quad (8-291)$$
- $$\text{mvLX}[0][0][1] = \text{mvLXN}[1] \quad (8-292)$$
- $$\text{gbIdx} = \text{gbIdxN} \quad (8-293)$$
13. When $\text{mmvd_flag}[\text{xCb}][\text{yCb}]$ is equal to 1, the following applies:
- The derivation process for merge motion vector difference as specified in 8.4.2.7 is invoked with the luma location (xCb, yCb) , the luma motion vectors $\text{mvL0}[0][0]$, $\text{mvL1}[0][0]$, the reference indices refIdxL0 , refIdxL1 and the prediction list utilization flags $\text{predFlagL0}[0][0]$ and $\text{predFlagL1}[0][0]$ as inputs, and the motion vector differences mMvdL0 and mMvdL1 as outputs.
 - The motion vector difference mMvdLX is added to the merge motion vectors mvLX for X being 0 and 1 as follows:
- $$\text{mvLX}[0][0][0] += \text{mMvdLX}[0] \quad (8-294)$$
- $$\text{mvLX}[0][0][1] += \text{mMvdLX}[1] \quad (8-295)$$

Như được thể hiện trên Bảng 2, gbIdx có thể biểu thị chỉ số trọng số lưỡng dự đoán, và gbIdxCol có thể biểu thị chỉ số trọng số lưỡng dự đoán cho ứng viên hợp nhất thời gian (ví dụ, ứng viên vector chuyển động theo thời gian trong danh sách ứng viên hợp nhất). Trong thủ tục trích xuất vector chuyển động của thành phần độ chói cho chế độ hợp nhất, khi loại lát cắt hoặc loại nhóm ô là B (mục lục 4 của 8.4.2.2), thì gbIdxCol có thể được trích xuất là gbIdxCol . Tức là, chỉ số trọng số của ứng viên vector chuyển động theo thời gian có thể được trích xuất là chỉ số trọng số của khối cùng vị trí.

Trong khi đó, theo phương án khác của sáng chế, khi xây dựng ứng viên vector chuyển động cho chế độ hợp nhất theo các đơn vị là các khối con, thì chỉ số trọng số dành cho trung bình được lấy trọng số của các ứng viên vector chuyển động theo thời gian có thể được trích xuất. Ở đây, chế độ hợp nhất theo các đơn vị là các khối con có thể được gọi là chế độ hợp nhất afin (theo các đơn vị là các khối con). Ứng viên vector chuyển động theo thời gian có thể chỉ thị ứng viên vector chuyển động theo thời gian dựa trên khối con, và có thể được gọi là ứng viên SbTMVP (hoặc ATMVP). Tức là, khi

kiểu liên dự đoán là lưỡng dự đoán, thì thông tin chỉ số trọng số trên ứng viên SbTMVP (hoặc ứng viên vector chuyển động theo thời gian dựa trên khối con) trong danh sách ứng viên hợp nhất affin hoặc danh sách ứng viên hợp nhất khối con là có thể được trích xuất.

Ví dụ, thông tin chỉ số trọng số trên trung bình được lấy trọng số của các ứng viên vector chuyển động theo thời gian dựa trên khối con có thể luôn luôn được trích xuất là 0. Ở đây, thông tin chỉ số trọng số là 0 có thể có nghĩa là việc các trọng số của mỗi chiều tham chiếu (tức là chiều dự đoán L0 và chiều dự đoán L1 trong quá trình lưỡng dự đoán) là giống nhau. Ví dụ, thủ tục để trích xuất vector chuyển động và chỉ số tham chiếu trong chế độ hợp nhất khối con và thủ tục để trích xuất ứng viên hợp nhất thời gian dựa trên khối con có thể như được thể hiện ở Bảng 3 và Bảng 4 dưới đây

Bảng 3

8.4.4.2 Derivation process for motion vectors and reference indices in subblock merge mode

Inputs to this process are:

- a luma location (x_{Cb} , y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- two variables $cbWidth$ and $cbHeight$ specifying the width and the height of the luma coding block.

Outputs of this process are:

- the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$,
- the reference indices $refIdxL0$ and $refIdxL1$,
- the prediction list utilization flag arrays $predFlagL0[xSbIdx][ySbIdx]$ and $predFlagL1[xSbIdx][ySbIdx]$,
- the luma subblock motion vector arrays in 1/16 fractional-sample accuracy $mvL0[xSbIdx][ySbIdx]$ and $mvL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the chroma subblock motion vector arrays in 1/32 fractional-sample accuracy $mvCL0[xSbIdx][ySbIdx]$ and $mvCL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the bi-prediction weight index $gbIdx$.

The variables $numSbX$, $numSbY$ and the subblock merging candidate list, $subblockMergeCandList$ are derived by the following ordered steps:

1. When $sps_sbtmvp_enabled_flag$ is equal to 1, the following applies:

- The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ and the luma coding block width as inputs, and the output being the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$ and $availableFlagB_2$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$ and $refIdxLXB_2$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and $predFlagLXB_2$, and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ and $mvLXB_2$, with X being 0 or 1.

- The derivation process for subblock-based temporal merging candidates as specified in clause 8.4.4.3 is invoked with the luma location (xCb, yCb), the luma coding block width cbWidth, the luma coding block height cbHeight, the availability flags availableFlagA₀, availableFlagA₁, availableFlagB₀, availableFlagB₁, the reference indices refIdxLXA₀, refIdxLXA₁, refIdxLXB₀, refIdxLXB₁, the prediction list utilization flags predFlagLXA₀, predFlagLXA₁, predFlagLXB₀, predFlagLXB₁ and the motion vectors mvLXA₀, mvLXA₁, mvLXB₀, mvLXB₁ as inputs and the output being the availability flag availableFlagSbCol, the bi-prediction weight index gbIdxSbCol, the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY, the reference indices refIdxLXSbCol, the luma motion vectors mvLXSbCol[xSbIdx][ySbIdx] and the prediction list utilization flags predFlagLXSbCol[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX - 1, ySbIdx = 0..numSbY - 1 and X being 0 or 1.

2. When sps_affine_enabled_flag is equal to 1, the sample locations (xNbA₀, yNbA₀), (xNbA₁, yNbA₁), (xNbA₂, yNbA₂), (xNbB₀, yNbB₀), (xNbB₁, yNbB₁), (xNbB₂, yNbB₂), (xNbB₃, yNbB₃), and the variables numSbX and numSbY are derived as follows:

$$(xA_0, yA_0) = (xCb - 1, yCb + cbHeight) \quad (8-536)$$

$$(xA_1, yA_1) = (xCb - 1, yCb + cbHeight - 1) \quad (8-537)$$

$$(xA_2, yA_2) = (xCb - 1, yCb) \quad (8-538)$$

$$(xB_0, yB_0) = (xCb + cbWidth, yCb - 1) \quad (8-539)$$

$$(xB_1, yB_1) = (xCb + cbWidth - 1, yCb - 1) \quad (8-540)$$

$$(xB_2, yB_2) = (xCb - 1, yCb - 1) \quad (8-541)$$

$$(xB_3, yB_3) = (xCb, yCb - 1) \quad (8-542)$$

$$numSbX = cbWidth >> 2 \quad (8-543)$$

$$numSbY = cbHeight >> 2 \quad (8-544)$$

3. When sps_affine_enabled_flag is equal to 1, the variable availableFlagA is set equal to FALSE and the following applies for (xNbA_k, yNbA_k) from (xNbA₀, yNbA₀) to (xNbA₁, yNbA₁):

- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location (xCurr, yCurr) set equal to (xCb, yCb) and the neighbouring luma location (xNbA_k, yNbA_k) as inputs, and the output is assigned to the block availability flag availableA_k.

- When availableA_k is equal to TRUE and MotionModelIdc[xNbA_k][yNbA_k] is greater than 0 and availableFlagA is equal to FALSE, the following applies:

- The variable availableFlagA is set equal to TRUE, motionModelIdcA is set equal to MotionModelIdc[xNbA_k][yNbA_k], (xNb, yNb) is set equal to (CbPosX[xNbA_k][yNbA_k], CbPosY[xNbA_k][yNbA_k]), nbW is set equal to CbWidth[xNbA_k][yNbA_k], nbH is set equal to CbHeight[xNbA_k][yNbA_k], numCpMv is set equal to MotionModelIdc[xNbA_k][yNbA_k] + 1, and gbIdxA is set equal to GbIdx[xNbA_k][yNbA_k].

- For X being replaced by either 0 or 1, the following applies:

- When PredFlagLX[xNbA_k][yNbA_k] is equal to 1, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb), the luma coding block width and height (cbWidth, cbHeight), the neighbouring luma coding block location (xNb, yNb), the neighbouring luma coding block width and height (nbW, nbH), and the number of control point motion vectors numCpMv as input, the control point motion vector predictor candidates cpMvLXA[cpIdx] with cpIdx = 0..numCpMv - 1 as output.

- The following assignments are made:

$$predFlagLXA = PredFlagLX[xNbA_k][yNbA_k] \quad (8-545)$$

$$refIdxLXA = RefIdxLX[xNbA_k][yNbA_k] \quad (8-546)$$

4. When `sps_affine_enabled_flag` is equal to 1, the variable `availableFlagB` is set equal to `FALSE` and the following applies for $(xNbB_k, yNbB_k)$ from $(xNbB_0, yNbB_0)$ to $(xNbB_2, yNbB_2)$:
 - The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location $(xCurr, yCurr)$ set equal to (xCb, yCb) and the neighbouring luma location $(xNbB_k, yNbB_k)$ as inputs, and the output is assigned to the block availability flag `availableBk`.
 - When `availableBk` is equal to `TRUE` and `MotionModelIdc[ xNbBk ][ yNbBk ]` is greater than 0 and `availableFlagB` is equal to `FALSE`, the following applies:
 - The variable `availableFlagB` is set equal to `TRUE`, `motionModelIdcB` is set equal to `MotionModelIdc[ xNbBk ][ yNbBk ]`, (xNb, yNb) is set equal to $(CbPosX[xNbAB][yNbB_k], CbPosY[xNbB_k][yNbB_k])$, `nbW` is set equal to `CbWidth[ xNbB_k ][ yNbB_k ]`, `nbH` is set equal to `CbHeight[ xNbB_k ][ yNbB_k ]`, `numCpMv` is set equal to `MotionModelIdc[ xNbB_k ][ yNbB_k ] + 1`, and `gbIdxB` is set equal to `GbIdx[ xNbB_k ][ yNbB_k ]`.
 - For `X` being replaced by either 0 or 1, the following applies:
 - When `PredFlagLX[ xNbB_k ][ yNbB_k ]` is equal to `TRUE`, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width and height $(cbWidth, cbHeight)$, the neighbouring luma coding block location (xNb, yNb) , the neighbouring luma coding block width and height (nbW, nbH) , and the number of control point motion vectors `numCpMv` as input, the control point motion vector predictor candidates `cpMvLXB[ cpIdx ]` with `cpIdx = 0 .. numCpMv - 1` as output.
 - The following assignments are made:

$$\text{predFlagLXB} = \text{PredFlagLX}[xNbB_k][yNbB_k] \quad (8-547)$$

$$\text{refIdxLXB} = \text{RefIdxLX}[xNbB_k][yNbB_k] \quad (8-548)$$

5. When `sps_affine_enabled_flag` is equal to 1, the derivation process for constructed affine control point motion vector merging candidates as specified in clause 8.4.4.6 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width and height $(cbWidth, cbHeight)$, the availability flags `availableA0`, `availableA1`, `availableA2`, `availableB0`, `availableB1`, `availableB2`, `availableB3` as inputs, and the availability flags `availableFlagConstK`, the reference indices `refIdxLXConstK`, prediction list utilization flags `predFlagLXConstK`, motion model indices `motionModelIdcConstK` and `cpMvpLXConstK[ cpIdx ]` with `X` being 0 or 1, `K = 1..6`, `cpIdx = 0..2` as outputs and `gbIdxConstK` is set equal to 0 with `K = 1..6`.
6. The initial subblock merging candidate list, `subblockMergeCandList`, is constructed as follows:


```

i = 0
if( availableFlagSbCol )
  subblockMergeCandList[ i++ ] = SbCol
if( availableFlagA && i < MaxNumSubblockMergeCand )
  subblockMergeCandList[ i++ ] = A
if( availableFlagB && i < MaxNumSubblockMergeCand )
  subblockMergeCandList[ i++ ] = B
if( availableFlagConst1 && i < MaxNumSubblockMergeCand )
  subblockMergeCandList[ i++ ] = Const1
if( availableFlagConst2 && i < MaxNumSubblockMergeCand )
  subblockMergeCandList[ i++ ] = Const2
if( availableFlagConst3 && i < MaxNumSubblockMergeCand )
  subblockMergeCandList[ i++ ] = Const3
if( availableFlagConst4 && i < MaxNumSubblockMergeCand )
  subblockMergeCandList[ i++ ] = Const4
if( availableFlagConst5 && i < MaxNumSubblockMergeCand )
  subblockMergeCandList[ i++ ] = Const5
if( availableFlagConst6 && i < MaxNumSubblockMergeCand )
  subblockMergeCandList[ i++ ] = Const6
      
```

(8-549)

7. The variable numCurrMergeCand and numOrigMergeCand are set equal to the number of merging candidates in the subblockMergeCandList.
8. When numCurrMergeCand is less than MaxNumSubblockMergeCand, the following is repeated until numCurrMergeCand is equal to MaxNumSubblockMergeCand, with mvZero[0] and mvZero[1] both being equal to 0:

- The reference indices, the prediction list utilization flags and the motion vectors of zeroCand_m with m equal to (numCurrMergeCand – numOrigMergeCand) are derived as follows:

$$\text{refIdxL0ZeroCand}_m = 0 \quad (8-550)$$

$$\text{predFlagL0ZeroCand}_m = 1 \quad (8-551)$$

$$\text{cpMvL0ZeroCand}_m[0] = \text{mvZero} \quad (8-552)$$

$$\text{cpMvL0ZeroCand}_m[1] = \text{mvZero} \quad (8-553)$$

$$\text{cpMvL0ZeroCand}_m[2] = \text{mvZero} \quad (8-554)$$

$$\text{refIdxL1ZeroCand}_m = (\text{tile_group_type} == B) ? 0 : -1 \quad (8-555)$$

$$\text{predFlagL1ZeroCand}_m = (\text{tile_group_type} == B) ? 1 : 0 \quad (8-556)$$

$$\text{cpMvL1ZeroCand}_m[0] = \text{mvZero} \quad (8-557)$$

$$\text{cpMvL1ZeroCand}_m[1] = \text{mvZero} \quad (8-558)$$

$$\text{cpMvL1ZeroCand}_m[2] = \text{mvZero} \quad (8-559)$$

$$\text{motionModelIdxZeroCand}_m = 1 \quad (8-560)$$

$$\text{gbIdxZeroCand}_m = 0 \quad (8-561)$$

- The candidate zeroCand_m with m equal to (numCurrMergeCand – numOrigMergeCand) is added at the end of subblockMergeCandList and numCurrMergeCand is incremented by 1 as follows:

$$\text{subblockMergeCandList}[\text{numCurrMergeCand}++] = \text{zeroCand}_m \quad (8-562)$$

The variables refIdxL0, refIdxL1, predFlagL0[xSbIdx][ySbIdx], predFlagL1[xSbIdx][ySbIdx], mvL0[xSbIdx][ySbIdx], mvL1[xSbIdx][ySbIdx], mvCL0[xSbIdx][ySbIdx], and mvCL1[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX – 1, ySbIdx = 0..numSbY – 1 are derived as follows:

- If subblockMergeCandList[merge_subblock_idx[xCb][yCb]] is equal to SbCol, the bi-prediction weight index gbIdx is set equal to 0 and the following applies with X being 0 or 1:

$$\text{refIdxLX} = \text{refIdxLXSbCol} \quad (8-563)$$

- For xSbIdx = 0..numSbX – 1, ySbIdx = 0..numSbY – 1, the following applies:

$$\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}] = \text{predFlagLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] \quad (8-564)$$

$$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][0] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][0] \quad (8-565)$$

$$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][1] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][1] \quad (8-566)$$

- When predFlagLX[xSbIdx][ySbIdx], is equal to 1, the derivation process for chroma motion vectors in clause 8.4.2.13 is invoked with mvLX[xSbIdx][ySbIdx] and refIdxLX as inputs, and the output being mvCLX[xSbIdx][ySbIdx].
- The following assignment is made for x = xCb ..xCb + cbWidth – 1 and y = yCb ..yCb + cbHeight – 1:

$$\text{MotionModelIdx}[x][y] = 0 \quad (8-567)$$

- Otherwise (subblockMergeCandList[merge_subblock_idx[xCb][yCb]] is not equal to SbCol), the following applies with X being 0 or 1:

- The following assignments are made with N being the candidate at position $\text{merge_subblock_idx}[x_{Cb}][y_{Cb}]$ in the subblock merging candidate list $\text{subblockMergeCandList}$ ($N = \text{subblockMergeCandList}[\text{merge_subblock_idx}[x_{Cb}][y_{Cb}]]$):

$$\text{refIdxLX} = \text{refIdxLXN} \quad (8-568)$$

$$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-569)$$

$$\text{cpMvLX}[0] = \text{cpMvLXN}[0] \quad (8-570)$$

$$\text{cpMvLX}[1] = \text{cpMvLXN}[1] \quad (8-571)$$

$$\text{cpMvLX}[2] = \text{cpMvLXN}[2] \quad (8-572)$$

$$\text{numCpMv} = \text{motionModelIdxN} + 1 \quad (8-573)$$

$$\text{gbIdx} = \text{gbIdxN} \quad (8-574)$$

- For $x_{SbIdx} = 0..numSbX - 1$, $y_{SbIdx} = 0..numSbY - 1$, the following applies:

$$\text{predFlagLX}[x_{SbIdx}][y_{SbIdx}] = \text{predFlagLX}[0][0] \quad (8-575)$$

- When $\text{predFlagLX}[0][0]$ is equal to 1, the derivation process for motion vector arrays from affine control point motion vectors as specified in subclause 8.4.4.9 is invoked with the luma coding block location (x_{Cb}, y_{Cb}) , the luma coding block width $cbWidth$, the luma prediction block height $cbHeight$, the number of control point motion vectors $numCpMv$, the control point motion vectors $\text{cpMvLX}[cpIdx]$ with $cpIdx$ being 0..2, and the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$ as inputs, the luma subblock motion vector array $\text{mvLX}[x_{SbIdx}][y_{SbIdx}]$ and the chroma subblock motion vector array $\text{mvCLX}[x_{SbIdx}][y_{SbIdx}]$ with $x_{SbIdx} = 0..numSbX - 1$, $y_{SbIdx} = 0..numSbY - 1$ as outputs.

- The following assignment is made for $x = x_{Cb}..x_{Cb} + cbWidth - 1$ and $y = y_{Cb}..y_{Cb} + cbHeight - 1$:

$$\text{MotionModelIdx}[x][y] = \text{numCpMv} - 1 \quad (8-576)$$

Bảng 4

8.4.4.3 Derivation process for subblock-based temporal merging candidates

Inputs to this process are:

- a luma location (x_{Cb}, y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable $cbWidth$ specifying the width of the current coding block in luma samples,
- a variable $cbHeight$ specifying the height of the current coding block in luma samples.
- the availability flags availableFlagA_0 , availableFlagA_1 , availableFlagB_0 , and availableFlagB_1 of the neighbouring coding units,
- the reference indices refIdxLXA_0 , refIdxLXA_1 , refIdxLXB_0 , and refIdxLXB_1 of the neighbouring coding units,
- the prediction list utilization flags predFlagLXA_0 , predFlagLXA_1 , predFlagLXB_0 , and predFlagLXB_1 of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy mvLXA_0 , mvLXA_1 , mvLXB_0 , and mvLXB_1 of the neighbouring coding units.

Outputs of this process are:

- the availability flag $\text{availableFlagSbCol}$,
- the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$,
- the reference indices refIdxL0SbCol and refIdxL1SbCol ,
- the luma motion vectors in 1/16 fractional-sample accuracy $\text{mvL0SbCol}[x_{SbIdx}][y_{SbIdx}]$ and $\text{mvL1SbCol}[x_{SbIdx}][y_{SbIdx}]$ with $x_{SbIdx} = 0..numSbX - 1$, $y_{SbIdx} = 0..numSbY - 1$,
- the prediction list utilization flags $\text{predFlagL0SbCol}[x_{SbIdx}][y_{SbIdx}]$ and $\text{predFlagL1SbCol}[x_{SbIdx}][y_{SbIdx}]$ with $x_{SbIdx} = 0..numSbX - 1$, $y_{SbIdx} = 0..numSbY - 1$,
- the bi-prediction weight index gbIdxSbCol .

The gbIdxSbCol is set equal to 0.

...

Như được thể hiện ở Bảng 3 và Bảng 4 trên đây, `gbIdx` có thể biểu thị chỉ số trọng số lưỡng dự đoán, `gbIdxSbCol` có thể biểu thị chỉ số trọng số lưỡng dự đoán dành cho ứng viên hợp nhất thời gian dựa trên khối con (ví dụ, ứng viên vector chuyển động theo thời gian trong danh sách ứng viên hợp nhất dựa trên khối con), và trong thủ tục (8.4.4.3) để trích xuất ứng viên hợp nhất thời gian dựa trên khối con, thì `gbIdxSbCol` có thể được trích xuất là 0. Tức là, chỉ số trọng số của ứng viên vector chuyển động theo thời gian dựa trên khối con có thể được trích xuất là 0.

Theo cách khác, thông tin chỉ số trọng số trên trung bình được lấy trọng số của các ứng viên vector chuyển động theo thời gian dựa trên khối con có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối trung tâm theo thời gian. Ví dụ, khối trung tâm theo thời gian có thể chỉ thị khối con hoặc mẫu được đặt tại tâm của khối cùng vị trí hoặc khối cùng vị trí, và cụ thể là, có thể chỉ thị khối con được đặt tại đáy phải của bốn khối con hoặc mẫu trung tâm của khối cùng vị trí hoặc mẫu. Ví dụ, trong trường hợp này, thì thủ tục để trích xuất vector chuyển động và chỉ số tham chiếu trong chế độ hợp nhất khối con, thủ tục để trích xuất ứng viên hợp nhất thời gian dựa trên khối con, và thủ tục để trích xuất thông tin chuyển động cơ sở cho việc hợp nhất thời gian dựa trên khối con, là có thể được thể hiện ở Bảng 5, Bảng 6, và Bảng 7.

Bảng 5

8.4.4.2 Derivation process for motion vectors and reference indices in subblock merge mode

Inputs to this process are:

- a luma location (x_{Cb} , y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- two variables $cbWidth$ and $cbHeight$ specifying the width and the height of the luma coding block.

Outputs of this process are:

- the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$,
- the reference indices $refIdxL0$ and $refIdxL1$,
- the prediction list utilization flag arrays $predFlagL0[xSbIdx][ySbIdx]$ and $predFlagL1[xSbIdx][ySbIdx]$,
- the luma subblock motion vector arrays in 1/16 fractional-sample accuracy $mvL0[xSbIdx][ySbIdx]$ and $mvL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the chroma subblock motion vector arrays in 1/32 fractional-sample accuracy $mvCL0[xSbIdx][ySbIdx]$ and $mvCL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the bi-prediction weight index $gbIdx$.

The variables $numSbX$, $numSbY$ and the subblock merging candidate list, $subblockMergeCandList$ are derived by the following ordered steps:

1. When $sps_sbtmp_enabled_flag$ is equal to 1, the following applies:

- The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ and the luma coding block width as inputs, and the output being the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$ and $availableFlagB_2$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$ and $refIdxLXB_2$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and $predFlagLXB_2$, and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ and $mvLXB_2$, with X being 0 or 1.

- The derivation process for subblock-based temporal merging candidates as specified in clause 8.4.4.3 is invoked with the luma location (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$, the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ as inputs and the output being the availability flag $availableFlagSbCol$, the bi-prediction weight index $gbIdxSbCol$, the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$, the reference indices $refIdxLXSbCol$, the luma motion vectors $mvLXSbCol[xSbIdx][ySbIdx]$ and the prediction list utilization flags $predFlagLXSbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$ and X being 0 or 1.

2. When $sps_affine_enabled_flag$ is equal to 1, the sample locations (x_{NbA_0} , y_{NbA_0}), (x_{NbA_1} , y_{NbA_1}), (x_{NbA_2} , y_{NbA_2}), (x_{NbB_0} , y_{NbB_0}), (x_{NbB_1} , y_{NbB_1}), (x_{NbB_2} , y_{NbB_2}), (x_{NbB_3} , y_{NbB_3}), and the variables $numSbX$ and $numSbY$ are derived as follows:

$$(x_{A_0}, y_{A_0}) = (x_{Cb} - 1, y_{Cb} + cbHeight) \quad (8-536)$$

$$(x_{A_1}, y_{A_1}) = (x_{Cb} - 1, y_{Cb} + cbHeight - 1) \quad (8-537)$$

$$(x_{A_2}, y_{A_2}) = (x_{Cb} - 1, y_{Cb}) \quad (8-538)$$

$$(x_{B_0}, y_{B_0}) = (x_{Cb} + cbWidth, y_{Cb} - 1) \quad (8-539)$$

$$(x_{B_1}, y_{B_1}) = (x_{Cb} + cbWidth - 1, y_{Cb} - 1) \quad (8-540)$$

$$(x_{B_2}, y_{B_2}) = (x_{Cb} - 1, y_{Cb} - 1) \quad (8-541)$$

$$(x_{B_3}, y_{B_3}) = (x_{Cb}, y_{Cb} - 1) \quad (8-542)$$

$\text{numSbX} = \text{cbWidth} >> 2$ (8-543)

$\text{numSbY} = \text{cbHeight} >> 2$ (8-544)

3. When `sps_affine_enabled_flag` is equal to 1, the variable `availableFlagA` is set equal to FALSE and the following applies for $(\text{xNbA}_k, \text{yNbA}_k)$ from $(\text{xNbA}_0, \text{yNbA}_0)$ to $(\text{xNbA}_1, \text{yNbA}_1)$:

- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location $(\text{xCurr}, \text{yCurr})$ set equal to (xCb, yCb) and the neighbouring luma location $(\text{xNbA}_k, \text{yNbA}_k)$ as inputs, and the output is assigned to the block availability flag `availableAk`.
- When `availableAk` is equal to TRUE and `MotionModelIdc[ xNbAk ][ yNbAk ]` is greater than 0 and `availableFlagA` is equal to FALSE, the following applies:
 - The variable `availableFlagA` is set equal to TRUE, `motionModelIdcA` is set equal to `MotionModelIdc[ xNbAk ][ yNbAk ]`, (xNb, yNb) is set equal to $(\text{CbPosX}[\text{xNbA}_k][\text{yNbA}_k], \text{CbPosY}[\text{xNbA}_k][\text{yNbA}_k])$, `nbW` is set equal to `CbWidth[ xNbAk ][ yNbAk ]`, `nbH` is set equal to `CbHeight[ xNbAk ][ yNbAk ]`, `numCpMv` is set equal to `MotionModelIdc[ xNbAk ][ yNbAk ] + 1`, and `gbldxA` is set equal to `GblIdx[ xNbAk ][ yNbAk ]`.
 - For X being replaced by either 0 or 1, the following applies:
 - When `PredFlagLX[ xNbAk ][ yNbAk ]` is equal to 1, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width and height $(\text{cbWidth}, \text{cbHeight})$, the neighbouring luma coding block location (xNb, yNb) , the neighbouring luma coding block width and height (nbW, nbH) , and the number of control point motion vectors `numCpMv` as input, the control point motion vector predictor candidates `cpMvLXA[ cpIdx ]` with `cpIdx = 0 .. numCpMv - 1` as output.
- The following assignments are made:

$\text{predFlagLXA} = \text{PredFlagLX}[\text{xNbA}_k][\text{yNbA}_k]$ (8-545)

$\text{refIdxLXA} = \text{RefIdxLX}[\text{xNbA}_k][\text{yNbA}_k]$ (8-546)

4. When `sps_affine_enabled_flag` is equal to 1, the variable `availableFlagB` is set equal to FALSE and the following applies for $(\text{xNbB}_k, \text{yNbB}_k)$ from $(\text{xNbB}_0, \text{yNbB}_0)$ to $(\text{xNbB}_2, \text{yNbB}_2)$:

- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location $(\text{xCurr}, \text{yCurr})$ set equal to (xCb, yCb) and the neighbouring luma location $(\text{xNbB}_k, \text{yNbB}_k)$ as inputs, and the output is assigned to the block availability flag `availableBk`.
- When `availableBk` is equal to TRUE and `MotionModelIdc[ xNbBk ][ yNbBk ]` is greater than 0 and `availableFlagB` is equal to FALSE, the following applies:
 - The variable `availableFlagB` is set equal to TRUE, `motionModelIdcB` is set equal to `MotionModelIdc[ xNbBk ][ yNbBk ]`, (xNb, yNb) is set equal to $(\text{CbPosX}[\text{xNbB}_k][\text{yNbB}_k], \text{CbPosY}[\text{xNbB}_k][\text{yNbB}_k])$, `nbW` is set equal to `CbWidth[ xNbBk ][ yNbBk ]`, `nbH` is set equal to `CbHeight[ xNbBk ][ yNbBk ]`, `numCpMv` is set equal to `MotionModelIdc[ xNbBk ][ yNbBk ] + 1`, and `gbldxB` is set equal to `GblIdx[ xNbBk ][ yNbBk ]`.
 - For X being replaced by either 0 or 1, the following applies:
 - When `PredFlagLX[ xNbBk ][ yNbBk ]` is equal to TRUE, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width and height $(\text{cbWidth}, \text{cbHeight})$, the neighbouring luma coding block location (xNb, yNb) , the neighbouring luma coding block width and height (nbW, nbH) , and the number of control point motion vectors `numCpMv` as input, the control point motion vector predictor candidates `cpMvLXB[ cpIdx ]` with `cpIdx = 0 .. numCpMv - 1` as output.
- The following assignments are made:

$$\text{predFlagLXB} = \text{PredFlagLX}[\text{xNbB}_k][\text{yNbB}_k] \quad (8-547)$$

$$\text{refIdxLXB} = \text{RefIdxLX}[\text{xNbB}_k][\text{yNbB}_k] \quad (8-548)$$

5. When `sps_affine_enabled_flag` is equal to 1, the derivation process for constructed affine control point motion vector merging candidates as specified in clause 8.4.4.6 is invoked with the luma coding block location (`xCb`, `yCb`), the luma coding block width and height (`cbWidth`, `cbHeight`), the availability flags `availableA0`, `availableA1`, `availableA2`, `availableB0`, `availableB1`, `availableB2`, `availableB3` as inputs, and the availability flags `availableFlagConstK`, the reference indices `refIdxLXConstK`, prediction list utilization flags `predFlagLXConstK`, motion model indices `motionModelIdxConstK` and `cpMvpLXConstK[ cpIdx ]` with `X` being 0 or 1, `K = 1..6`, `cpIdx = 0..2` as outputs and `gbIdxConstK` is set equal to 0 with `K = 1..6`.

6. The initial subblock merging candidate list, `subblockMergeCandList`, is constructed as follows:

```

i = 0
if( availableFlagSbCol )
    subblockMergeCandList[ i++ ] = SbCol
if( availableFlagA && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = A
if( availableFlagB && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = B
if( availableFlagConst1 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const1
if( availableFlagConst2 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const2
if( availableFlagConst3 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const3
if( availableFlagConst4 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const4
if( availableFlagConst5 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const5
if( availableFlagConst6 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const6

```

(8-549)

7. The variable `numCurrMergeCand` and `numOrigMergeCand` are set equal to the number of merging candidates in the `subblockMergeCandList`.

8. When `numCurrMergeCand` is less than `MaxNumSubblockMergeCand`, the following is repeated until `numCurrMergeCand` is equal to `MaxNumSubblockMergeCand`, with `mvZero[0]` and `mvZero[1]` both being equal to 0:

- The reference indices, the prediction list utilization flags and the motion vectors of `zeroCandm` with `m` equal to (`numCurrMergeCand – numOrigMergeCand`) are derived as follows:

$$\text{refIdxL0ZeroCand}_m = 0 \quad (8-550)$$

$$\text{predFlagL0ZeroCand}_m = 1 \quad (8-551)$$

$$\text{cpMvL0ZeroCand}_m[0] = \text{mvZero} \quad (8-552)$$

$$\text{cpMvL0ZeroCand}_m[1] = \text{mvZero} \quad (8-553)$$

$$\text{cpMvL0ZeroCand}_m[2] = \text{mvZero} \quad (8-554)$$

$$\text{refIdxL1ZeroCand}_m = (\text{tile_group_type} == B) ? 0 : -1 \quad (8-555)$$

$$\text{predFlagL1ZeroCand}_m = (\text{tile_group_type} == B) ? 1 : 0 \quad (8-556)$$

$$\text{cpMvL1ZeroCand}_m[0] = \text{mvZero} \quad (8-557)$$

$$\text{cpMvL1ZeroCand}_m[1] = \text{mvZero} \quad (8-558)$$

$$\text{cpMvL1ZeroCand}_m[2] = \text{mvZero} \quad (8-559)$$

$$\text{motionModelIdxZeroCand}_m = 1 \quad (8-560)$$

$$\text{gbIdxZeroCand}_m = 0 \quad (8-561)$$

- The candidate zeroCand_m with m equal to $(\text{numCurrMergeCand} - \text{numOrigMergeCand})$ is added at the end of $\text{subblockMergeCandList}$ and numCurrMergeCand is incremented by 1 as follows:

$$\text{subblockMergeCandList}[\text{numCurrMergeCand}++] = \text{zeroCand}_m \quad (8-562)$$

The variables refIdxL0 , refIdxL1 , $\text{predFlagL0}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{predFlagL1}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{mvL0}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{mvL1}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{mvCL0}[\text{xSbIdx}][\text{ySbIdx}]$, and $\text{mvCL1}[\text{xSbIdx}][\text{ySbIdx}]$ with $\text{xSbIdx} = 0.. \text{numSbX} - 1$, $\text{ySbIdx} = 0.. \text{numSbY} - 1$ are derived as follows:

- If $\text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$ is equal to SbCol , the bi-prediction weight index gbIdx is set equal to 0 and the following applies with X being 0 or 1:

$$\text{refIdxLX} = \text{refIdxLXSbCol} \quad (8-563)$$

- For $\text{xSbIdx} = 0.. \text{numSbX} - 1$, $\text{ySbIdx} = 0.. \text{numSbY} - 1$, the following applies:

$$\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}] = \text{predFlagLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] \quad (8-564)$$

$$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][0] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][0] \quad (8-565)$$

$$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][1] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][1] \quad (8-566)$$

- When $\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}]$ is equal to 1, the derivation process for chroma motion vectors in clause 8.4.2.13 is invoked with $\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}]$ and refIdxLX as inputs, and the output being $\text{mvCLX}[\text{xSbIdx}][\text{ySbIdx}]$.
- The following assignment is made for $x = \text{xCb}.. \text{xCb} + \text{cbWidth} - 1$ and $y = \text{yCb}.. \text{yCb} + \text{cbHeight} - 1$:

$$\text{MotionModelIdx}[x][y] = 0 \quad (8-567)$$

- Otherwise ($\text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$ is not equal to SbCol), the following applies with X being 0 or 1:

- The following assignments are made with N being the candidate at position $\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]$ in the subblock merging candidate list $\text{subblockMergeCandList}$ ($N = \text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$):

$$\text{refIdxLX} = \text{refIdxLXN} \quad (8-568)$$

$$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-569)$$

$$\text{cpMvLX}[0] = \text{cpMvLXN}[0] \quad (8-570)$$

$$\text{cpMvLX}[1] = \text{cpMvLXN}[1] \quad (8-571)$$

$$\text{cpMvLX}[2] = \text{cpMvLXN}[2] \quad (8-572)$$

$$\text{numCpMv} = \text{motionModelIdxN} + 1 \quad (8-573)$$

$$\text{gbIdx} = \text{gbIdxN} \quad (8-574)$$

- For $\text{xSbIdx} = 0.. \text{numSbX} - 1$, $\text{ySbIdx} = 0.. \text{numSbY} - 1$, the following applies:

$$\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}] = \text{predFlagLX}[0][0] \quad (8-575)$$

- When $\text{predFlagLX}[0][0]$ is equal to 1, the derivation process for motion vector arrays from affine control point motion vectors as specified in subclause 8.4.4.9 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width cbWidth , the luma prediction block height cbHeight , the number of control point motion vectors numCpMv , the control point motion vectors $\text{cpMvLX}[\text{cpIdx}]$ with cpIdx being $0..2$, and the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY as inputs, the luma subblock motion vector array $\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}]$ and the chroma subblock motion vector array $\text{mvCLX}[\text{xSbIdx}][\text{ySbIdx}]$ with $\text{xSbIdx} = 0.. \text{numSbX} - 1$, $\text{ySbIdx} = 0.. \text{numSbY} - 1$ as outputs.
- The following assignment is made for $x = \text{xCb}.. \text{xCb} + \text{cbWidth} - 1$ and $y = \text{yCb}.. \text{yCb} + \text{cbHeight} - 1$:

$$\text{MotionModelIdx}[x][y] = \text{numCpMv} - 1 \quad (8-576)$$

Bảng 6

8.4.4.3 Derivation process for subblock-based temporal merging candidates

Inputs to this process are:

- a luma location (x_{Cb} , y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable $cbWidth$ specifying the width of the current coding block in luma samples,
- a variable $cbHeight$ specifying the height of the current coding block in luma samples.
- the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, and $availableFlagB_1$ of the neighbouring coding units,
- the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, and $refIdxLXB_1$ of the neighbouring coding units,
- the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, and $predFlagLXB_1$ of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, and $mvLXB_1$ of the neighbouring coding units.

Outputs of this process are:

- the availability flag $availableFlagSbCol$,
- the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$,
- the reference indices $refIdxL0SbCol$ and $refIdxL1SbCol$,
- the luma motion vectors in 1/16 fractional-sample accuracy $mvL0SbCol[xSbIdx][ySbIdx]$ and $mvL1SbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the prediction list utilization flags $predFlagL0SbCol[xSbIdx][ySbIdx]$ and $predFlagL1SbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the bi-prediction weight index $gbIdxSbCol$.

The availability flag $availableFlagSbCol$ is derived as follows.

- If one or more of the following conditions is true, $availableFlagSbCol$ is set equal to 0.
 - $tile_group_temporal_mvp_enable_flag$ is equal to 0.
 - sps_sbtmvp_flag is equal to 0.
 - $cbWidth$ is less than 8.
 - $cbHeight$ is less than 8.
- Otherwise, the following ordered steps apply:
 1. The location (x_{Ctb} , y_{Ctb}) of the top-left sample of the luma coding tree block that contains the current coding block and the location (x_{Ctr} , y_{Ctr}) of the below-right center sample of the current luma coding block are derived as follows:

$$x_{Ctb} = (x_{Cb} \gg CtuLog2Size) \ll CtuLog2Size \quad (8-577)$$

$$y_{Ctb} = (y_{Cb} \gg CtuLog2Size) \ll CtuLog2Size \quad (8-578)$$

$$x_{Ctr} = x_{Cb} + (cbWidth / 2) \quad (8-579)$$

$$y_{Ctr} = y_{Cb} + (cbHeight / 2) \quad (8-580)$$
 2. The luma location ($x_{ColCtrCb}$, $y_{ColCtrCb}$) is set equal to the top-left sample of the collocated luma coding block covering the location given by (x_{Ctr} , y_{Ctr}) inside $ColPic$ relative to the top-left luma sample of the collocated picture specified by $ColPic$.

3. The derivation process for subblock-based temporal merging base motion data as specified in clause 8.4.4.4 is invoked with the location (xCtb, yCtb), the location (xCtrCb, yCtrCb), the availability flags availableFlagA₀, availableFlagA₁, availableFlagB₀ and availableFlagB₁, and the prediction list utilization flags predFlagLXA₀, predFlagLXA₁, predFlagLXB₀ and predFlagLXB₁, and the reference indices refIdxLXA₀, refIdxLXA₁, refIdxLXB₀ and refIdxLXB₁, and the motion vectors mvLXA₀, mvLXA₁, mvLXB₀ and mvLXB₁, with X being 0 and 1 as inputs and the motion vectors ctrMvLX, the prediction list utilization flags ctrPredFlagLX and the reference indices ctrRefIdxLX of the collocated block, with X being 0 and 1, the bi-prediction weight index gbIdxSbCol, and the temporal motion vector tempMV as outputs.
4. The variable availableFlagSbCol is derived as follows:
 - If both ctrPredFlagL0 and ctrPredFlagL1 are equal to 0, availableFlagSbCol is set equal to 0.
 - Otherwise, availableFlagSbCol is set equal to 1.

When availableFlagSbCol is equal to 1, the following applies:

- The variables numSbX, numSbY, sbWidth, sbHeight and refIdxLXSbCol are derived as follows:

$$\text{numSbX} = \text{cbWidth} \gg 3 \quad (8-581)$$

$$\text{numSbY} = \text{cbHeight} \gg 3 \quad (8-582)$$

$$\text{sbWidth} = \text{cbWidth} / \text{numSbX} \quad (8-583)$$

$$\text{sbHeight} = \text{cbHeight} / \text{numSbY} \quad (8-584)$$

$$\text{refIdxLXSbCol} = 0 \quad (8-585)$$
- For xSbIdx = 0 .. numSbX – 1 and ySbIdx = 0 .. numSbY – 1, the motion vectors mvLXSbCol[xSbIdx][ySbIdx] and prediction list utilization flags predFlagLXSbCol[xSbIdx][ySbIdx] are derived as follows:

- The luma location (xSb, ySb) specifying the top-left sample of the current coding subblock relative to the top-left luma sample of the current picture is derived as follows:

$$xSb = xCb + xSbIdx * sbWidth \quad (8-586)$$

$$ySb = yCb + ySbIdx * sbHeight \quad (8-587)$$

- The location (xColSb, yColSb) of the collocated subblock inside ColPic is derived as follows.

$$xColSb = Clip3(xCtb, \\ Min(CurPicWidthInSamplesY - 1, xCtb + (1 << CtbLog2SizeY) + 3), \\ xSb + (tempMv[0] >> 4)) \quad (8-588)$$

$$yColSb = Clip3(yCtb, \\ Min(CurPicHeightInSamplesY - 1, yCtb + (1 << CtbLog2SizeY) - 1), \\ ySb + (tempMv[1] >> 4)) \quad (8-589)$$

- The variable currCb specifies the luma coding block covering the current coding subblock inside the current picture.
- The variable colCb specifies the luma coding block covering the modified location given by ((xColSb >> 3) << 3, (yColSb >> 3) << 3) inside the ColPic.
- The luma location (xColCb, yColCb) is set equal to the top-left sample of the collocated luma coding block specified by colCb relative to the top-left luma sample of the collocated picture specified by ColPic.
- The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, (xColCb, yColCb), refIdxL0 set equal to 0 and sbFlag set equal to 1 as inputs and the output being assigned to the motion vector of the subblock mvL0SbCol[xSbIdx][ySbIdx] and availableFlagL0SbCol.
- The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, (xColCb, yColCb), refIdxL1 set equal to 0 and sbFlag set equal to 1 as inputs and the output being assigned to the motion vector of the subblock mvL1SbCol[xSbIdx][ySbIdx] and availableFlagL1SbCol.
- When availableFlagL0SbCol and availableFlagL1SbCol are both equal to 0, the following applies for X being 0 and 1:

$$mvLXSbCol[xSbIdx][ySbIdx] = ctrMvLX \quad (8-590)$$

$$predFlagLXSbCol[xSbIdx][ySbIdx] = ctrPredFlagLX \quad (8-591)$$

Bảng 7

8.4.4.4 Derivation process for subblock-based temporal merging base motion data

Inputs to this process are:

- the location (x_{Ctb} , y_{Ctb}) of the top-left sample of the luma coding tree block that contains the current coding block,
- the location (x_{ColCtb} , y_{ColCtb}) of the top-left sample of the collocated luma coding block that covers the below-right center sample.
- the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, and $availableFlagB_1$ of the neighbouring coding units,
- the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, and $refIdxLXB_1$ of the neighbouring coding units,
- the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, and $predFlagLXB_1$ of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, and $mvLXB_1$ of the neighbouring coding units.

Outputs of this process are:

- the motion vectors $ctrMvL0$ and $ctrMvL1$,
- the prediction list utilization flags $ctrPredFlagL0$ and $ctrPredFlagL1$,
- the reference indices $ctrRefIdxL0$ and $ctrRefIdxL1$,
- the temporal motion vector $tempMV$,
- the bi-prediction weight index $gbIdxSbCol$.

The variable $tempMv$ is set as follows:

$$tempMv[0] = 0 \quad (8-592)$$

$$tempMv[1] = 0 \quad (8-593)$$

The variable $currPic$ specifies the current picture.

The variable $availableFlagN$ is set equal to FALSE, and the following applies:

- When $availableFlagA_1$ is equal to 1, the following applies:
 - $availableFlagN$ is set equal to TRUE,
 - $refIdxLXN$ is set equal to $refIdxLXA_0$ and $mvLXN$ is set equal to $mvLXA_0$, for X being replaced by 0 and 1.
- When $availableFlagN$ is equal to FALSE and $availableFlagLB_1$ is equal to 1, the following applies:
 - $availableFlagN$ is set equal to TRUE,

- refIdxLXN is set equal to refIdxLXB_0 and mvLXN is set equal to mvLXB_0 , for X being replaced by 0 and 1.
- When availableFlagN is equal to FALSE and availableFlagB_0 is equal to 1, the following applies:
 - availableFlagN is set equal to TRUE.
 - refIdxLXN is set equal to refIdxLXB_1 and mvLXN is set equal to mvLXB_1 , for X being replaced by 0 and 1.
- When availableFlagN is equal to FALSE and availableFlagA_0 is equal to 1, the following applies:
 - availableFlagN is set equal to TRUE.
 - refIdxLXN is set equal to refIdxLXA_1 and mvLXN is set equal to mvLXA_1 , for X being replaced by 0 and 1.

When availableFlagN is equal to TRUE, the following applies:

- If all of the following conditions are true, tempMV is set equal to mvL1N :
 - predFlagL1N is equal to 1,
 - $\text{DiffPicOrderCnt}(\text{ColPic}, \text{RefPicList1}[\text{refIdxL1N}])$ is equal to 0,
 - $\text{DiffPicOrderCnt}(\text{aPic}, \text{currPic})$ is less than or equal to 0 for every picture aPic in every reference picture list of the current tile group,
 - tile_group_type is equal to B,
 - $\text{collocated_from_l0_flag}$ is equal to 0.
- Otherwise if all of the following conditions are true, tempMV is set equal to mvL0N :
 - predFlagL0N is equal to 1,
 - $\text{DiffPicOrderCnt}(\text{ColPic}, \text{RefPicList0}[\text{refIdxL0N}])$ is equal to 0.

The location $(\text{xColCb}, \text{yColCb})$ of the collocated block inside ColPic is derived as follows.

```

xColCb = Clip3( xCtb,
                Min( CurPicWidthInSamplesY - 1, xCtb + ( 1 << CtbLog2SizeY ) + 3 ), (8-59
4)
                xColCtCb + ( tempMv[0] >> 4 ) )

yColCb = Clip3( yCtb,
                Min( CurPicHeightInSamplesY - 1, yCtb + ( 1 << CtbLog2SizeY ) - 1 ), (8-5
95)
                yColCtCb + ( tempMv[1] >> 4 ) )

```

The array colPredMode is set equal to the prediction mode array CuPredMode of the collocated picture specified by ColPic .

The motion vectors ctrMvL0 and ctrMvL1 , the prediction list utilization flags ctrPredFlagL0 and ctrPredFlagL1 , and the reference indices ctrRefIdxL0 and ctrRefIdxL1 are derived as follows:

- If $\text{colPredMode}[\text{xColCb}][\text{yColCb}]$ is equal to MODE_INTER, the following applies:
 - The variable currCb specifies the luma coding block covering $(\text{xCtCb}, \text{yCtCb})$ inside the current picture.
 - The variable colCb specifies the luma coding block covering the modified location given by $((\text{xColCb} \gg 3) \ll 3, (\text{yColCb} \gg 3) \ll 3)$ inside the ColPic .
 - The luma location $(\text{xColCb}, \text{yColCb})$ is set equal to the top-left sample of the collocated luma coding block specified by colCb relative to the top-left luma sample of the collocated picture specified by ColPic .
 - The gbIdxSbCol is set equal to gbIdxcolCb .

- The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with `currCb`, `colCb`, `(xCb, yCb)`, `centerRefIdxL0`, and `sbFlag` set equal to 1 as inputs and the output being assigned to `ctrMvL0` and `ctrPredFlagL0`.
- The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with `currCb`, `colCb`, `(xCb, yCb)`, `centerRefIdxL1`, and `sbFlag` set equal to 1 as inputs and the output being assigned to `ctrMvL1` and `ctrPredFlagL1`.
- Otherwise, the following applies:

$$\text{ctrPredFlagL0} = 0 \quad (8-596)$$

$$\text{ctrPredFlagL1} = 0 \quad (8-597)$$

Như được thể hiện trên Bảng 5, Bảng 6, và Bảng 7, `gbiIdx` có thể biểu thị chỉ số trọng số lưỡng dự đoán, và `gbiIdxSbCol` có thể biểu thị chỉ số trọng số lưỡng dự đoán dành cho ứng viên hợp nhất thời gian dựa trên khối con (ví dụ, ứng viên vector chuyển động theo thời gian trong danh sách ứng viên hợp nhất dựa trên khối con). Trong thủ tục (8.4.4.4) để trích xuất thông tin chuyển động cơ sở trên việc hợp nhất thời gian dựa trên khối con, thì `gbiIdxSbCol` có thể được trích xuất dưới dạng `gbiIdxcolCb`. Tức là, chỉ số trọng số của ứng viên vector chuyển động theo thời gian dựa trên khối con có thể được trích xuất dưới dạng chỉ số trọng số của khối trung tâm theo thời gian. Ví dụ, khối trung tâm theo thời gian có thể chỉ thị khối con hoặc mẫu được đặt tại tâm của khối cùng vị trí hoặc khối cùng vị trí, và cụ thể là, có thể chỉ thị khối con được đặt tại đáy phải của bốn khối con hoặc mẫu trung tâm của khối cùng vị trí hoặc mẫu.

Theo cách khác, thông tin chỉ số trọng số trên trung bình được lấy trọng số của các ứng viên vector chuyển động theo thời gian dựa trên khối con có thể được trích xuất dựa trên thông tin chỉ số trọng số theo các đơn vị là mỗi khối con, và có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối trung tâm theo thời gian khi khối con không khả dụng. Ví dụ, khối trung tâm theo thời gian có thể chỉ thị khối con hoặc mẫu được đặt tại tâm của khối cùng vị trí hoặc khối cùng vị trí, và cụ thể là, có thể chỉ thị khối con được đặt tại đáy phải của bốn khối con hoặc mẫu trung tâm của khối cùng vị trí hoặc mẫu. Ví dụ, trong trường hợp này, thì thủ tục để trích xuất vector chuyển động và chỉ số tham chiếu trong chế độ hợp nhất khối con, thủ tục để trích xuất ứng viên hợp nhất thời gian dựa trên khối con, và thủ tục để trích xuất thông tin chuyển động cơ sở

cho việc hợp nhất thời gian dựa trên khối con, là có thể được thể hiện ở Bảng 8, Bảng 9, và Bảng 10.

Bảng 8

8.4.4.2 Derivation process for motion vectors and reference indices in subblock merge mode

Inputs to this process are:

- a luma location (x_{Cb} , y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- two variables $cbWidth$ and $cbHeight$ specifying the width and the height of the luma coding block.

Outputs of this process are:

- the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$,
- the reference indices $refIdxL0$ and $refIdxL1$,
- the prediction list utilization flag arrays $predFlagL0[xSbIdx][ySbIdx]$ and $predFlagL1[xSbIdx][ySbIdx]$,
- the luma subblock motion vector arrays in 1/16 fractional-sample accuracy $mvL0[xSbIdx][ySbIdx]$ and $mvL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the chroma subblock motion vector arrays in 1/32 fractional-sample accuracy $mvCL0[xSbIdx][ySbIdx]$ and $mvCL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the bi-prediction weight index $gbiIdx$.

The variables $numSbX$, $numSbY$ and the subblock merging candidate list, $subblockMergeCandList$ are derived by the following ordered steps:

1. When $sps_sbtmp_enabled_flag$ is equal to 1, the following applies:

- The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ and the luma coding block width as inputs, and the output being the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$ and $availableFlagB_2$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$ and $refIdxLXB_2$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and $predFlagLXB_2$, and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ and $mvLXB_2$, with X being 0 or 1.

- The derivation process for subblock-based temporal merging candidates as specified in clause 8.4.4.3 is invoked with the luma location (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$, the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ as inputs and the output being the availability flag $availableFlagSbCol$, the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$, the reference indices $refIdxLXSbCol$, the bi-prediction weight index $gbiIdxSbCol[xSbIdx][ySbIdx]$, the luma motion vectors $mvLXSbCol[xSbIdx][ySbIdx]$ and the prediction list utilization flags $predFlagLXSbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$ and X being 0 or 1.

2. When $sps_affine_enabled_flag$ is equal to 1, the sample locations (x_{NbA_0} , y_{NbA_0}), (x_{NbA_1} , y_{NbA_1}), (x_{NbA_2} , y_{NbA_2}), (x_{NbB_0} , y_{NbB_0}), (x_{NbB_1} , y_{NbB_1}), (x_{NbB_2} , y_{NbB_2}), (x_{NbB_3} , y_{NbB_3}), and the variables $numSbX$ and $numSbY$ are derived as follows:

$$(x_{A_0}, y_{A_0}) = (x_{Cb} - 1, y_{Cb} + cbHeight) \quad (8-536)$$

$$(x_{A_1}, y_{A_1}) = (x_{Cb} - 1, y_{Cb} + cbHeight - 1) \quad (8-537)$$

$$(x_{A_2}, y_{A_2}) = (x_{Cb} - 1, y_{Cb}) \quad (8-538)$$

$$(x_{B_0}, y_{B_0}) = (x_{Cb} + cbWidth, y_{Cb} - 1) \quad (8-539)$$

$$(x_{B_1}, y_{B_1}) = (x_{Cb} + cbWidth - 1, y_{Cb} - 1) \quad (8-540)$$

$$(x_{B_2}, y_{B_2}) = (x_{Cb} - 1, y_{Cb} - 1) \quad (8-541)$$

$$(x_{B_3}, y_{B_3}) = (x_{Cb}, y_{Cb} - 1) \quad (8-542)$$

numSbX = cbWidth >> 2 (8-543)

numSbY = cbHeight >> 2 (8-544)

3. When `sps_affine_enabled_flag` is equal to 1, the variable `availableFlagA` is set equal to FALSE and the following applies for $(xNbA_k, yNbA_k)$ from $(xNbA_0, yNbA_0)$ to $(xNbA_1, yNbA_1)$:

- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location $(xCurr, yCurr)$ set equal to (xCb, yCb) and the neighbouring luma location $(xNbA_k, yNbA_k)$ as inputs, and the output is assigned to the block availability flag `availableAk`.
- When `availableAk` is equal to TRUE and `MotionModelIdc[ xNbAk ][ yNbAk ]` is greater than 0 and `availableFlagA` is equal to FALSE, the following applies:
 - The variable `availableFlagA` is set equal to TRUE, `motionModelIdcA` is set equal to `MotionModelIdc[ xNbAk ][ yNbAk ]`, (xNb, yNb) is set equal to $(CbPosX[xNbA_k][yNbA_k], CbPosY[xNbA_k][yNbA_k])$, `nbW` is set equal to `CbWidth[ xNbAk ][ yNbAk ]`, `nbH` is set equal to `CbHeight[ xNbAk ][ yNbAk ]`, `numCpMv` is set equal to `MotionModelIdc[ xNbAk ][ yNbAk ] + 1`, and `gbIdxA` is set equal to `GbIdx[ xNbAk ][ yNbAk ]`.
 - For X being replaced by either 0 or 1, the following applies:
 - When `PredFlagLX[ xNbAk ][ yNbAk ]` is equal to 1, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width and height $(cbWidth, cbHeight)$, the neighbouring luma coding block location (xNb, yNb) , the neighbouring luma coding block width and height (nbW, nbH) , and the number of control point motion vectors `numCpMv` as input, the control point motion vector predictor candidates `cpMvLXA[ cpIdx ]` with `cpIdx = 0 .. numCpMv - 1` as output.

- The following assignments are made:

`predFlagLXA = PredFlagLX[ xNbAk ][ yNbAk ]` (8-545)

`refIdxLXA = RefIdxLX[ xNbAk ][ yNbAk ]` (8-546)

4. When `sps_affine_enabled_flag` is equal to 1, the variable `availableFlagB` is set equal to FALSE and the following applies for $(xNbB_k, yNbB_k)$ from $(xNbB_0, yNbB_0)$ to $(xNbB_2, yNbB_2)$:

- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location $(xCurr, yCurr)$ set equal to (xCb, yCb) and the neighbouring luma location $(xNbB_k, yNbB_k)$ as inputs, and the output is assigned to the block availability flag `availableBk`.
- When `availableBk` is equal to TRUE and `MotionModelIdc[ xNbBk ][ yNbBk ]` is greater than 0 and `availableFlagB` is equal to FALSE, the following applies:
 - The variable `availableFlagB` is set equal to TRUE, `motionModelIdcB` is set equal to `MotionModelIdc[ xNbBk ][ yNbBk ]`, (xNb, yNb) is set equal to $(CbPosX[xNbB_k][yNbB_k], CbPosY[xNbB_k][yNbB_k])$, `nbW` is set equal to `CbWidth[ xNbBk ][ yNbBk ]`, `nbH` is set equal to `CbHeight[ xNbBk ][ yNbBk ]`, `numCpMv` is set equal to `MotionModelIdc[ xNbBk ][ yNbBk ] + 1`, and `gbIdxB` is set equal to `GbIdx[ xNbBk ][ yNbBk ]`.
 - For X being replaced by either 0 or 1, the following applies:
 - When `PredFlagLX[ xNbBk ][ yNbBk ]` is equal to TRUE, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width and height $(cbWidth, cbHeight)$, the neighbouring luma coding block location (xNb, yNb) , the neighbouring luma coding block width and height (nbW, nbH) , and the number of control point motion vectors `numCpMv` as input, the control point motion vector predictor candidates `cpMvLXB[ cpIdx ]` with `cpIdx = 0 .. numCpMv - 1` as output.
- The following assignments are made:

$$\text{predFlagLXB} = \text{PredFlagLX}[\text{xNbB}_k][\text{yNbB}_k] \quad (8-547)$$

$$\text{refIdxLXB} = \text{RefIdxLX}[\text{xNbB}_k][\text{yNbB}_k] \quad (8-548)$$

5. When `sps_affine_enabled_flag` is equal to 1, the derivation process for constructed affine control point motion vector merging candidates as specified in clause 8.4.4.6 is invoked with the luma coding block location (`xCb`, `yCb`), the luma coding block width and height (`cbWidth`, `cbHeight`), the availability flags `availableA0`, `availableA1`, `availableA2`, `availableB0`, `availableB1`, `availableB2`, `availableB3` as inputs, and the availability flags `availableFlagConstK`, the reference indices `refIdxLXConstK`, prediction list utilization flags `predFlagLXConstK`, motion model indices `motionModelIdcConstK` and `cpMvpLXConstK[ cpIdx ]` with `X` being 0 or 1, `K = 1..6`, `cpIdx = 0..2` as outputs and `gbIdxConstK` is set equal to 0 with `K = 1..6`.
6. The initial subblock merging candidate list, `subblockMergeCandList`, is constructed as follows:

```

i = 0
if( availableFlagSbCol )
    subblockMergeCandList[ i++ ] = SbCol
if( availableFlagA && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = A
if( availableFlagB && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = B
if( availableFlagConst1 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const1
if( availableFlagConst2 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const2
if( availableFlagConst3 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const3
if( availableFlagConst4 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const4
if( availableFlagConst5 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const5
if( availableFlagConst6 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const6

```

(8-549)

7. The variable `numCurrMergeCand` and `numOrigMergeCand` are set equal to the number of merging candidates in the `subblockMergeCandList`.
8. When `numCurrMergeCand` is less than `MaxNumSubblockMergeCand`, the following is repeated until `numCurrMergeCand` is equal to `MaxNumSubblockMergeCand`, with `mvZero[0]` and `mvZero[1]` both being equal to 0:
 - The reference indices, the prediction list utilization flags and the motion vectors of `zeroCandm` with `m` equal to (`numCurrMergeCand – numOrigMergeCand`) are derived as follows:

$$\text{refIdxL0ZeroCand}_m = 0 \quad (8-550)$$

$$\text{predFlagL0ZeroCand}_m = 1 \quad (8-551)$$

$$\text{cpMvL0ZeroCand}_m[0] = \text{mvZero} \quad (8-552)$$

$$\text{cpMvL0ZeroCand}_m[1] = \text{mvZero} \quad (8-553)$$

$$\text{cpMvL0ZeroCand}_m[2] = \text{mvZero} \quad (8-554)$$

$$\text{refIdxL1ZeroCand}_m = (\text{tile_group_type} == B) ? 0 : -1 \quad (8-555)$$

$$\text{predFlagL1ZeroCand}_m = (\text{tile_group_type} == B) ? 1 : 0 \quad (8-556)$$

$$\text{cpMvL1ZeroCand}_m[0] = \text{mvZero} \quad (8-557)$$

$$\text{cpMvL1ZeroCand}_m[1] = \text{mvZero} \quad (8-558)$$

$$\text{cpMvL1ZeroCand}_m[2] = \text{mvZero} \quad (8-559)$$

$$\text{motionModelIdcZeroCand}_m = 1 \quad (8-560)$$

$$\text{gbIdxZeroCand}_m = 0 \quad (8-561)$$

- The candidate zeroCand_m with m equal to $(\text{numCurrMergeCand} - \text{numOrigMergeCand})$ is added at the end of $\text{subblockMergeCandList}$ and numCurrMergeCand is incremented by 1 as follows:

$$\text{subblockMergeCandList}[\text{numCurrMergeCand}++] = \text{zeroCand}_m \quad (8-562)$$

The variables refIdxL0 , refIdxL1 , $\text{predFlagL0}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{predFlagL1}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{mvL0}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{mvL1}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{mvCL0}[\text{xSbIdx}][\text{ySbIdx}]$, and $\text{mvCL1}[\text{xSbIdx}][\text{ySbIdx}]$ with $\text{xSbIdx} = 0.. \text{numSbX} - 1$, $\text{ySbIdx} = 0.. \text{numSbY} - 1$ are derived as follows:

- If $\text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$ is equal to SbCol , the bi-prediction weight index gbIdx is set equal to 0 and the following applies with X being 0 or 1:

$$\text{refIdxLX} = \text{refIdxLXSbCol} \quad (8-563)$$

- For $\text{xSbIdx} = 0.. \text{numSbX} - 1$, $\text{ySbIdx} = 0.. \text{numSbY} - 1$, the following applies:

$$\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}] = \text{predFlagLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] \quad (8-564)$$

$$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][0] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][0] \quad (8-565)$$

$$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][1] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][1] \quad (8-566)$$

- When $\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}]$ is equal to 1, the derivation process for chroma motion vectors in clause 8.4.2.13 is invoked with $\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}]$ and refIdxLX as inputs, and the output being $\text{mvCLX}[\text{xSbIdx}][\text{ySbIdx}]$.
- The following assignment is made for $x = \text{xCb}.. \text{xCb} + \text{cbWidth} - 1$ and $y = \text{yCb}.. \text{yCb} + \text{cbHeight} - 1$:

$$\text{MotionModelIdx}[x][y] = 0 \quad (8-567)$$

- Otherwise ($\text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$ is not equal to SbCol), the following applies with X being 0 or 1:

- The following assignments are made with N being the candidate at position $\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]$ in the subblock merging candidate list $\text{subblockMergeCandList}$ ($N = \text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$):

$$\text{refIdxLX} = \text{refIdxLXN} \quad (8-568)$$

$$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-569)$$

$$\text{cpMvLX}[0] = \text{cpMvLXN}[0] \quad (8-570)$$

$$\text{cpMvLX}[1] = \text{cpMvLXN}[1] \quad (8-571)$$

$$\text{cpMvLX}[2] = \text{cpMvLXN}[2] \quad (8-572)$$

$$\text{numCpMv} = \text{motionModelIdxN} + 1 \quad (8-573)$$

$$\text{gbIdx} = \text{gbIdxN} \quad (8-574)$$

- For $\text{xSbIdx} = 0.. \text{numSbX} - 1$, $\text{ySbIdx} = 0.. \text{numSbY} - 1$, the following applies:

$$\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}] = \text{predFlagLX}[0][0] \quad (8-575)$$

- When $\text{predFlagLX}[0][0]$ is equal to 1, the derivation process for motion vector arrays from affine control point motion vectors as specified in subclause 8.4.4.9 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width cbWidth , the luma prediction block height cbHeight , the number of control point motion vectors numCpMv , the control point motion vectors $\text{cpMvLX}[\text{cpIdx}]$ with cpIdx being 0..2, and the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY as inputs, the luma subblock motion vector array $\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}]$ and the chroma subblock motion vector array $\text{mvCLX}[\text{xSbIdx}][\text{ySbIdx}]$ with $\text{xSbIdx} = 0.. \text{numSbX} - 1$, $\text{ySbIdx} = 0.. \text{numSbY} - 1$ as outputs.

- The following assignment is made for $x = \text{xCb}.. \text{xCb} + \text{cbWidth} - 1$ and $y = \text{yCb}.. \text{yCb} + \text{cbHeight} - 1$:

$$\text{MotionModelIdx}[x][y] = \text{numCpMv} - 1 \quad (8-576)$$

Bảng 9

8.4.4.3 Derivation process for subblock-based temporal merging candidates

Inputs to this process are:

- a luma location (xCb, yCb) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable cbWidth specifying the width of the current coding block in luma samples,
- a variable cbHeight specifying the height of the current coding block in luma samples.
- the availability flags availableFlagA₀, availableFlagA₁, availableFlagB₀, and availableFlagB₁ of the neighbouring coding units,
- the reference indices refIdxLXA₀, refIdxLXA₁, refIdxLXB₀, and refIdxLXB₁ of the neighbouring coding units,
- the prediction list utilization flags predFlagLXA₀, predFlagLXA₁, predFlagLXB₀, and predFlagLXB₁ of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy mvLXA₀, mvLXA₁, mvLXB₀, and mvLXB₁ of the neighbouring coding units.

Outputs of this process are:

- the availability flag availableFlagSbCol,
- the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY,
- the reference indices refIdxLOSbCol and refIdxL1SbCol,
- the luma motion vectors in 1/16 fractional-sample accuracy mvLOSbCol[xSbIdx][ySbIdx] and mvL1SbCol[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX – 1, ySbIdx = 0 .. numSbY – 1,
- the bi-prediction weight index gbiIdxSbCol[xSbIdx][ySbIdx], the prediction list utilization flags predFlagLOSbCol[xSbIdx][ySbIdx] and predFlagL1SbCol[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX – 1, ySbIdx = 0 .. numSbY – 1.

The availability flag availableFlagSbCol is derived as follows.

- If one or more of the following conditions is true, availableFlagSbCol is set equal to 0.
 - tile_group_temporal_mvp_enable_flag is equal to 0.
 - sps_sbtmvp_flag is equal to 0.
 - cbWidth is less than 8.
 - cbHeight is less than 8.
- Otherwise, the following ordered steps apply:
 1. The location (xCtb, yCtb) of the top-left sample of the luma coding tree block that contains the current coding block and the location (xCtr, yCtr) of the below-right center sample of the current luma coding block are derived as follows:

$$xCtb = (xCb \gg CtuLog2Size) \ll CtuLog2Size \quad (8-577)$$

$$yCtb = (yCb \gg CtuLog2Size) \ll CtuLog2Size \quad (8-578)$$

$$xCtr = xCb + (cbWidth / 2) \quad (8-579)$$

$$yCtr = yCb + (cbHeight / 2) \quad (8-580)$$
 2. The luma location (xColCtrCb, yColCtrCb) is set equal to the top-left sample of the collocated luma coding block covering the location given by (xCtr, yCtr) inside ColPic relative to the top-left luma sample of the collocated picture specified by ColPic.

3. The derivation process for subblock-based temporal merging base motion data as specified in clause 8.4.4.4 is invoked with the location (xCtb, yCtb), the location (xColCtCb, yColCtCb), the availability flags availableFlagA₀, availableFlagA₁, availableFlagB₀ and availableFlagB₁, and the prediction list utilization flags predFlagLXA₀, predFlagLXA₁, predFlagLXB₀ and predFlagLXB₁, and the reference indices refIdxLXA₀, refIdxLXA₁, refIdxLXB₀ and refIdxLXB₁, and the motion vectors mvLXA₀, mvLXA₁, mvLXB₀ and mvLXB₁, with X being 0 and 1 as inputs and the motion vectors ctrMvLX, the prediction list utilization flags ctrPredFlagLX and the reference indices ctrRefIdxLX of the collocated block, with X being 0 and 1, the bi-prediction weight index ctrgbiIdx, and the temporal motion vector tempMV as outputs.
4. The variable availableFlagSbCol is derived as follows:
 - If both ctrPredFlagL0 and ctrPredFlagL1 are equal to 0, availableFlagSbCol is set equal to 0.
 - Otherwise, availableFlagSbCol is set equal to 1.

When availableFlagSbCol is equal to 1, the following applies:

- The variables numSbX, numSbY, sbWidth, sbHeight and refIdxLXSbCol are derived as follows:

$$\text{numSbX} = \text{cbWidth} \gg 3 \quad (8-581)$$

$$\text{numSbY} = \text{cbHeight} \gg 3 \quad (8-582)$$

$$\text{sbWidth} = \text{cbWidth} / \text{numSbX} \quad (8-583)$$

$$\text{sbHeight} = \text{cbHeight} / \text{numSbY} \quad (8-584)$$

$$\text{refIdxLXSbCol} = 0 \quad (8-585)$$
- For xSbIdx = 0 .. numSbX – 1 and ySbIdx = 0 .. numSbY – 1, the motion vectors mvLXSbCol[xSbIdx][ySbIdx] and prediction list utilization flags predFlagLXSbCol[xSbIdx][ySbIdx] are derived as follows:
 - The luma location (xSb, ySb) specifying the top-left sample of the current coding subblock relative to the top-left luma sample of the current picture is derived as follows:

$$\text{xSb} = \text{xCb} + \text{xSbIdx} * \text{sbWidth} \quad (8-586)$$

$$\text{ySb} = \text{yCb} + \text{ySbIdx} * \text{sbHeight} \quad (8-587)$$
 - The location (xColSb, yColSb) of the collocated subblock inside ColPic is derived as follows.

$$\begin{aligned} \text{xColSb} = & \text{Clip3}(\text{xCtb}, \\ & \text{Min}(\text{CurPicWidthInSamplesY} - 1, \text{xCtb} + (1 \ll \text{CtbLog2SizeY}) + 3), \\ & \text{xSb} + (\text{tempMv}[0] \gg 4)) \end{aligned} \quad (8-588)$$

$$\begin{aligned} \text{yColSb} = & \text{Clip3}(\text{yCtb}, \\ & \text{Min}(\text{CurPicHeightInSamplesY} - 1, \text{yCtb} + (1 \ll \text{CtbLog2SizeY}) - 1), \\ & \text{ySb} + (\text{tempMv}[1] \gg 4)) \end{aligned} \quad (8-589)$$
 - The variable currCb specifies the luma coding block covering the current coding subblock inside the current picture.
 - The variable colCb specifies the luma coding block covering the modified location given by ((xColSb >> 3) << 3, (yColSb >> 3) << 3) inside the ColPic.
 - The luma location (xColCb, yColCb) is set equal to the top-left sample of the collocated luma coding block specified by colCb relative to the top-left luma sample of the collocated picture specified by ColPic.
 - The gbiIdxSbCol[xSbIdx][ySbIdx] is set equal to gbiIdxcolCb.

- The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with `currCb`, `colCb`, (`xColCb`, `yColCb`), `refIdxL0` set equal to 0 and `sbFlag` set equal to 1 as inputs and the output being assigned to the motion vector of the subblock `mvL0SbCol[ xSbIdx ][ ySbIdx ]` and `availableFlagL0SbCol`.
- The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with `currCb`, `colCb`, (`xColCb`, `yColCb`), `refIdxL1` set equal to 0 and `sbFlag` set equal to 1 as inputs and the output being assigned to the motion vector of the subblock `mvL1SbCol[ xSbIdx ][ ySbIdx ]` and `availableFlagL1SbCol`.
- When `availableFlagL0SbCol` and `availableFlagL1SbCol` are both equal to 0, the following applies for `X` being 0 and 1:

$$\text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] = \text{ctrMvLX} \quad (8-590)$$

$$\text{predFlagLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] = \text{ctrPredFlagLX} \quad (8-591)$$

$$\text{gbIdxSbCol}[\text{xSbIdx}][\text{ySbIdx}] = \text{ctrgbIdx} \quad (\text{x-xxx})$$

Bảng 10

8.4.4.4 Derivation process for subblock-based temporal merging base motion data

Inputs to this process are:

- the location (`xCtb`, `yCtb`) of the top-left sample of the luma coding tree block that contains the current coding block,
- the location (`xColCtb`, `yColCtb`) of the top-left sample of the collocated luma coding block that covers the below-right center sample.
- the availability flags `availableFlagA0`, `availableFlagA1`, `availableFlagB0`, and `availableFlagB1` of the neighbouring coding units,
- the reference indices `refIdxLXA0`, `refIdxLXA1`, `refIdxLXB0`, and `refIdxLXB1` of the neighbouring coding units,
- the prediction list utilization flags `predFlagLXA0`, `predFlagLXA1`, `predFlagLXB0`, and `predFlagLXB1` of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy `mvLXA0`, `mvLXA1`, `mvLXB0`, and `mvLXB1` of the neighbouring coding units.

Outputs of this process are:

- the motion vectors `ctrMvL0` and `ctrMvL1`,
- the prediction list utilization flags `ctrPredFlagL0` and `ctrPredFlagL1`,
- the reference indices `ctrRefIdxL0` and `ctrRefIdxL1`,
- the temporal motion vector `tempMV`,
- the bi-prediction weight index `ctrgbIdx`.

The variable `tempMv` is set as follows:

$$\text{tempMv}[0] = 0 \quad (8-592)$$

$$\text{tempMv}[1] = 0 \quad (8-593)$$

The variable `currPic` specifies the current picture.

The variable `availableFlagN` is set equal to FALSE, and the following applies:

- When `availableFlagA1` is equal to 1, the following applies:
 - `availableFlagN` is set equal to TRUE,

- refIdxLXN is set equal to refIdxLXA₀ and mvLXN is set equal to mvLXA₀, for X being replaced by 0 and 1.
- When availableFlagN is equal to FALSE and availableFlagLB₁ is equal to 1, the following applies:
 - availableFlagN is set equal to TRUE,
 - refIdxLXN is set equal to refIdxLXB₀ and mvLXN is set equal to mvLXB₀, for X being replaced by 0 and 1.
- When availableFlagN is equal to FALSE and availableFlagB₀ is equal to 1, the following applies:
 - availableFlagN is set equal to TRUE.
 - refIdxLXN is set equal to refIdxLXB₁ and mvLXN is set equal to mvLXB₁, for X being replaced by 0 and 1.
- When availableFlagN is equal to FALSE and availableFlagA₀ is equal to 1, the following applies:
 - availableFlagN is set equal to TRUE.
 - refIdxLXN is set equal to refIdxLXA₁ and mvLXN is set equal to mvLXA₁, for X being replaced by 0 and 1.

When availableFlagN is equal to TRUE, the following applies:

- If all of the following conditions are true, tempMV is set equal to mvLIN:
 - predFlagLIN is equal to 1,
 - DiffPicOrderCnt(ColPic, RefPicList1[refIdxLIN]) is equal to 0,
 - DiffPicOrderCnt(aPic, currPic) is less than or equal to 0 for every picture aPic in every reference picture list of the current tile group,
 - tile_group_type is equal to B,
 - collocated_from_l0_flag is equal to 0.

- Otherwise if all of the following conditions are true, tempMV is set equal to mvLON:
 - predFlagLON is equal to 1,
 - DiffPicOrderCnt(ColPic, RefPicList0[refIdxLON]) is equal to 0.

The location (xColCb, yColCb) of the collocated block inside ColPic is derived as follows.

```

xColCb = Clip3( xCtb,
                Min( CurPicWidthInSamplesY - 1, xCtb + ( 1 << CtbLog2SizeY ) + 3 ), (8-59
4)
                xColCtCb + ( tempMv[0] >> 4 ) )

yColCb = Clip3( yCtb,
                Min( CurPicHeightInSamplesY - 1, yCtb + ( 1 << CtbLog2SizeY ) - 1 ), (8-5
95)
                yColCtCb + ( tempMv[1] >> 4 ) )

```

The array colPredMode is set equal to the prediction mode array CuPredMode of the collocated picture specified by ColPic.

The motion vectors ctrMvL0 and ctrMvL1, the prediction list utilization flags ctrPredFlagL0 and ctrPredFlagL1, and the reference indices ctrRefIdxL0 and ctrRefIdxL1 are derived as follows:

- If colPredMode[xColCb][yColCb] is equal to MODE_INTER, the following applies:
 - The variable currCb specifies the luma coding block covering (xCtCb, yCtCb) inside the current picture.
 - The variable colCb specifies the luma coding block covering the modified location given by ((xColCb >> 3) << 3, (yColCb >> 3) << 3) inside the ColPic.

- The luma location (xColCb, yColCb) is set equal to the top-left sample of the collocated luma coding block specified by colCb relative to the top-left luma sample of the collocated picture specified by ColPic.
- The gbiIdxSbCol is set equal to ctrgbiIdx.
- The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with currCb, colCb, (xColCb, yColCb), centerRefIdxL0, and sbFlag set equal to 1 as inputs and the output being assigned to ctrMvL0 and ctrPredFlagL0.
- The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with currCb, colCb, (xColCb, yColCb), centerRefIdxL1, and sbFlag set equal to 1 as inputs and the output being assigned to ctrMvL1 and ctrPredFlagL1.
- Otherwise, the following applies:

$$\text{ctrPredFlagL0} = 0 \quad (8-596)$$

$$\text{ctrPredFlagL1} = 0 \quad (8-597)$$

Như được thể hiện trên Bảng 8, Bảng 9, và Bảng 10, gbiIdx có thể biểu thị chỉ số trọng số lưỡng dự đoán, và gbiIdxSbCol có thể biểu thị chỉ số trọng số lưỡng dự đoán dành cho ứng viên hợp nhất thời gian dựa trên khối con (ví dụ, ứng viên vector chuyển động theo thời gian trong danh sách ứng viên hợp nhất dựa trên khối con). Trong thủ tục (8.4.4.3) để trích xuất thông tin chuyển động cơ sở trên việc hợp nhất thời gian dựa trên khối con, thì gbiIdxSbCol có thể được trích xuất dưới dạng gbiIdxcolCb. Theo cách khác, trong thủ tục (8.4.4.3) để trích xuất thông tin chuyển động cơ sở trên việc hợp nhất thời gian dựa trên khối con theo điều kiện (ví dụ, khi cả availableFlagL0SbCol và availableFlagL1SbCol đều là 0), thì gbiIdxSbCol có thể được trích xuất dưới dạng ctrgbiIdx, và trong thủ tục (8.4.4.4) để trích xuất thông tin chuyển động cơ sở trên việc hợp nhất thời gian dựa trên khối con, thì ctrgbiIdx có thể được trích xuất dưới dạng gbiIdxSbCol. Tức là, chỉ số trọng số của ứng viên vector chuyển động theo thời gian dựa trên khối con có thể được trích xuất dưới dạng chỉ số trọng số theo các đơn vị là mỗi khối con, hoặc khi khối con không khả dụng, thì có thể được trích xuất dưới dạng chỉ số trọng số của khối trung tâm theo thời gian. Ví dụ, khối trung tâm theo thời gian có thể chỉ thị khối con hoặc mẫu được đặt tại tâm của khối cùng vị trí hoặc khối cùng vị trí, và cụ thể là, có thể chỉ thị khối con được đặt tại đáy phải của bốn khối con hoặc mẫu trung tâm của khối cùng vị trí hoặc mẫu.

Trong khi đó, theo phương án khác của sáng chế, khi xây dựng ứng viên vector

chuyển động dành cho chế độ hợp nhất, thì thông tin chỉ số trọng số trên ứng viên theo cặp có thể được trích xuất. Ví dụ, ứng viên theo cặp có thể được bao gồm trong danh sách ứng viên hợp nhất, và thông tin chỉ số trọng số trên trung bình được lấy trọng số của ứng viên theo cặp có thể được trích xuất. Ứng viên theo cặp có thể được trích xuất dựa trên các ứng viên hợp nhất khác trong danh sách ứng viên hợp nhất, và khi ứng viên theo cặp sử dụng việc lưỡng dự đoán, thì chỉ số trọng số dành cho trung bình được lấy trọng số có thể được trích xuất. Tức là, khi kiểu liên dự đoán là lưỡng dự đoán, thì thông tin chỉ số trọng số trên các ứng viên theo cặp trong danh sách ứng viên hợp nhất có thể được trích xuất.

Ứng viên theo cặp có thể được trích xuất dựa trên hai ứng viên hợp nhất khác (ví dụ, cand0 và cand1) trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất.

Ví dụ, thông tin chỉ số trọng số trên ứng viên theo cặp có thể được trích xuất dựa trên thông tin chỉ số trọng số trên bất kỳ trong số hai ứng viên hợp nhất đó (ví dụ, ứng viên hợp nhất cand0 hoặc ứng viên hợp nhất cand1). Ví dụ, thông tin chỉ số trọng số trên ứng viên theo cặp có thể được trích xuất dựa trên thông tin chỉ số trọng số trên ứng viên sử dụng việc lưỡng dự đoán trong số hai ứng viên hợp nhất đó.

Theo cách khác, khi thông tin chỉ số trọng số trên mỗi trong số hai ứng viên hợp nhất khác đó là giống như thông tin chỉ số trọng số thứ nhất, thì thông tin chỉ số trọng số trên ứng viên theo cặp có thể được trích xuất dựa trên thông tin chỉ số trọng số thứ nhất. Trong khi đó, khi thông tin chỉ số trọng số trên mỗi trong số hai ứng viên hợp nhất khác đó là không giống nhau, thì thông tin chỉ số trọng số trên ứng viên theo cặp có thể được trích xuất dựa trên thông tin chỉ số trọng số mặc định. Thông tin chỉ số trọng số mặc định có thể tương ứng với thông tin chỉ số trọng số trên việc gán trọng số giống nhau cho mỗi trong số L0 mẫu dự đoán và L1 mẫu dự đoán.

Theo cách khác, khi thông tin chỉ số trọng số trên mỗi trong số hai ứng viên hợp nhất khác đó là giống như thông tin chỉ số trọng số thứ nhất, thì thông tin chỉ số trọng

số trên ứng viên theo cặp có thể được trích xuất dựa trên thông tin chỉ số trọng số thứ nhất. Trong khi đó, khi thông tin chỉ số trọng số trên mỗi trong số hai ứng viên hợp nhất khác đó không giống nhau, thì thông tin chỉ số trọng số trên ứng viên theo cặp là bao gồm thông tin chỉ số trọng số mặc định trong số thông tin chỉ số trọng số trên mỗi trong số hai ứng viên khác đó. Thông tin chỉ số trọng số mặc định có thể tương ứng với thông tin chỉ số trọng số trên việc gán trọng số giống nhau cho mỗi trong số L0 mẫu dự đoán và L1 mẫu dự đoán.

Trong khi đó, theo phương án khác của sáng chế, khi xây dựng ứng viên vector chuyển động cho chế độ hợp nhất theo các đơn vị là các khối con, thì thông tin chỉ số trọng số trên trung bình được lấy trọng số của các ứng viên vector chuyển động theo thời gian có thể được trích xuất. Ở đây, chế độ hợp nhất theo các đơn vị là các khối con có thể được gọi là chế độ hợp nhất afin (theo các đơn vị là các khối con). Ứng viên vector chuyển động theo thời gian có thể chỉ thị ứng viên vector chuyển động theo thời gian dựa trên khối con, và có thể được gọi là ứng viên SbTMVP (hoặc ATMVP). Thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối lân cận bên trái của khối hiện tại. Tức là, khi ứng viên được trích xuất bởi SbTMVP sử dụng việc lưỡng dự đoán, thì chỉ số trọng số của khối lân cận bên trái của khối hiện tại có thể được trích xuất làm chỉ số trọng số dành cho chế độ hợp nhất dựa trên khối con.

Ví dụ, vì ứng viên SbTMVP có thể trích xuất khối cùng vị trí dựa trên khối kề bên trái trong không gian (hoặc khối lân cận bên trái) của khối hiện tại, nên chỉ số trọng số của khối lân cận bên trái có thể được coi là tin cậy. Theo đó, chỉ số trọng số dành cho ứng viên SbTMVP có thể được trích xuất làm chỉ số trọng số của khối lân cận bên trái.

Trong khi đó, theo phương án khác của sáng chế, khi xây dựng ứng viên vector chuyển động dành cho chế độ hợp nhất afin, trong trường hợp mà ứng viên hợp nhất afin sử dụng việc lưỡng dự đoán, thì thông tin chỉ số trọng số trên trung bình được lấy

trọng số là có thể được trích xuất. Tức là, khi kiểu liên dự đoán là lưỡng dự đoán, thì thông tin chỉ số trọng số trên ứng viên trong danh sách ứng viên hợp nhất affin hoặc danh sách ứng viên hợp nhất khối con có thể được trích xuất.

Ví dụ, trong số các ứng viên hợp nhất affin, thì ứng viên hợp nhất affin được xây dựng có thể trích xuất các ứng viên CP0, CP1, CP2, hoặc CP3 dựa trên khối liên kề về không gian của khối hiện tại hoặc khối liên kề về thời gian để chỉ thị ứng viên để trích xuất MVF làm mô hình affin. Ví dụ, CP0 có thể chỉ thị điểm kiểm soát được đặt tại vị trí mẫu phía trên bên trái của khối hiện tại, CP1 có thể chỉ thị điểm kiểm soát được đặt tại vị trí mẫu phía trên bên phải của khối hiện tại, và CP2 có thể chỉ thị vị trí mẫu đáy trái của khối hiện tại. Ngoài ra, CP3 có thể chỉ thị điểm kiểm soát được đặt tại vị trí mẫu đáy phải của khối hiện tại.

Ví dụ, trong số các ứng viên hợp nhất affin, thì các ứng viên hợp nhất affin được xây dựng có thể được tạo ra dựa trên sự kết hợp của mỗi điểm kiểm soát như {CP0, CP1, CP2}, {CP0, CP1, CP3}, {CP0, CP2, CP3}, {CP1, CP2, CP3}, {CP0, CP1}, và {CP0, CP2} có thể được tạo ra dựa trên sự kết hợp của mỗi điểm kiểm soát của khối hiện tại. Ví dụ, các ứng viên hợp nhất affin có thể bao gồm ít nhất một trong số {CPMV0, CPMV1, CPMV2}, {CPMV0, CPMV1, CPMV3}, {CPMV0, CPMV2, CPMV3}, {CPMV1, CPMV2, CPMV3}, {CPMV0, CPMV1}, và {CPMV0, CPMV2}. CPMV0, CPMV1, CPMV2, và CPMV3 có thể lần lượt tương ứng với các vector chuyển động dành cho CP0, CP1, CP2, và CP3.

Theo một phương án, khi ứng viên hợp nhất affin bao gồm CPMV dành cho điểm kiểm soát 0 (CP0) được đặt tại đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất affin có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của CP0. Tức là, khi ứng viên hợp nhất affin bao gồm CPMV dành cho điểm kiểm soát 0 (CP0) được đặt tại đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất affin có thể được trích xuất dựa trên thông tin chỉ số trọng số thứ 0 trên CP0. Trong trường hợp này, thì khối cụ thể trong số

các khối lân cận của CP0 là tương ứng với khối được sử dụng để trích xuất CPMV cho CP0, và các khối lân cận của CP0 có thể bao gồm khối lân cận ở góc đỉnh trái của khối hiện tại, khối lân cận bên trái kề với đáy của khối lân cận ở góc đỉnh trái, và khối lân cận phía trên kề với bên phải của khối lân cận ở góc đỉnh trái.

Ngược lại, khi ứng viên hợp nhất affin không bao gồm CPMV dành cho CP0 được đặt tại đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất affin có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của điểm kiểm soát 1 (CP1) được đặt tại vị trí đỉnh phải của khối hiện tại. Tức là, khi ứng viên hợp nhất affin không bao gồm CPMV dành cho CP0 được đặt tại đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất affin có thể được trích xuất dựa trên thông tin chỉ số trọng số thứ nhất dành cho điểm kiểm soát 1 (CP1) được đặt tại đỉnh phải của khối hiện tại. Trong số các khối lân cận của CP1, thì khối cụ thể là tương ứng với khối được sử dụng để trích xuất CPMV dành cho CP1, và các khối lân cận CP1 có thể bao gồm khối lân cận ở góc đỉnh phải của khối hiện tại và khối lân cận trên đỉnh kề với bên trái của khối lân cận ở góc đỉnh phải.

Theo phương pháp nêu trên, thì thông tin chỉ số trọng số trên ứng viên hợp nhất affin có thể được trích xuất dựa trên thông tin chỉ số trọng số của khối được dùng để trích xuất lần lượt {CPMV0, CPMV1, CPMV2}, {CPMV01, CPMV1, CPMV3}, {CPMV0, CPMV2, CPMV3}, {CPMV1, CPMV2, CPMV3}, {CPMV0, CPMV1} và {CPMV0, CPMV2}.

Theo phương án khác để trích xuất thông tin chỉ số trọng số trên ứng viên hợp nhất affin, khi thông tin chỉ số trọng số dành cho CP0 được đặt tại đỉnh trái của khối hiện tại và thông tin chỉ số trọng số trên CP1 được đặt tại đỉnh phải của khối hiện tại là giống nhau, thì thông tin chỉ số trọng số trên ứng viên hợp nhất affin có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của CP0. Trong khi đó, khi thông tin chỉ số trọng số trên CP0 được đặt tại đỉnh trái của khối hiện tại và thông tin chỉ số trọng số trên CP1 được đặt tại đỉnh phải của khối hiện tại là không

giống nhau, thì thông tin chỉ số trọng số trên ứng viên hợp nhất affin có thể được trích xuất dựa trên thông tin chỉ số trọng số mặc định. Thông tin chỉ số trọng số mặc định có thể tương ứng với thông tin chỉ số trọng số trên việc cho trọng số giống nhau cho mỗi trong số L0 mẫu dự đoán và L1 mẫu dự đoán.

Theo phương án khác để trích xuất thông tin chỉ số trọng số trên ứng viên hợp nhất affin, thì thông tin chỉ số trọng số trên ứng viên hợp nhất affin có thể được trích xuất làm chỉ số trọng số của ứng viên có tần suất xuất hiện cao trong số các chỉ số trọng số của mỗi ứng viên. Ví dụ, chỉ số trọng số của khối ứng viên mà được xác định dưới dạng vectơ chuyển động ở CP0 trong số các khối ứng viên CP0, chỉ số trọng số của khối ứng viên mà được xác định dưới dạng vectơ chuyển động ở CP1 trong số các khối ứng viên CP1, chỉ số trọng số của khối ứng viên mà được xác định dưới dạng vectơ chuyển động ở CP2 trong số các khối ứng viên CP2, và/hoặc chỉ số trọng số chồng nhất trong số các chỉ số trọng số của các khối ứng viên mà được xác định dưới dạng vectơ chuyển động ở CP3 trong số các khối ứng viên CP3, là có thể được trích xuất làm chỉ số trọng số của ứng viên hợp nhất affin.

Ví dụ, CP0 và CP1 có thể được sử dụng làm điểm kiểm soát, CP0, CP1, và CP2 có thể được sử dụng, và CP3 có thể không được sử dụng. Tuy nhiên, ví dụ, khi cần sử dụng ứng viên CP3 của khối affin (khối được lập mã trong chế độ dự đoán affin), thì phương pháp để trích xuất chỉ số trọng số ở khối ứng viên thời gian được mô tả ở các phương án nêu trên có thể được sử dụng.

Fig.10 và Fig.11 là các hình vẽ thể hiện sơ đồ ví dụ của phương pháp giải mã video/ảnh và các thành phần liên quan theo (các) phương án thực hiện của sáng chế.

Phương pháp được bộc lộ trên Fig.10 có thể được thực hiện bởi thiết bị mã hoá được bộc lộ trên Fig.2 hoặc Fig.11. Cụ thể là, ví dụ, các bước S1000 đến S1030 trên Fig.10 có thể được thực hiện bởi bộ dự đoán 220 của thiết bị mã hoá 200 trên Fig.11, và bước S1040 trên Fig.10 có thể được thực hiện bởi bộ mã hoá entropy 240 của thiết bị mã hoá 200 trên Fig.11. Ngoài ra, tuy không được thể hiện trên Fig.10, nhưng trên

Fig.11, thì các mẫu dự đoán hoặc thông tin liên quan đến việc dự đoán có thể được trích xuất bởi bộ dự đoán 220 của thiết bị mã hoá 200, thông tin dư có thể được trích xuất từ các mẫu gốc hoặc các mẫu dự đoán bởi bộ xử lý phần dư 230 của thiết bị mã hoá 200, và luồng bit có thể được tạo ra từ thông tin dư hoặc thông tin liên quan đến việc dự đoán bởi bộ mã hoá entropy 240 của thiết bị mã hoá 200. Phương pháp được bộc lộ trên Fig.10 có thể bao gồm các phương án được mô tả trên đây trong bản mô tả này.

Như được thể hiện trên Fig.10, thiết bị mã hoá có thể xác định chế độ liên dự đoán của khối hiện tại và tạo ra thông tin về chế độ liên dự đoán chỉ thị chế độ liên dự đoán (S1000). Ví dụ, thiết bị mã hoá có thể xác định chế độ hợp nhất, chế độ (hợp nhất) afin, hoặc chế độ hợp nhất khối con làm chế độ liên dự đoán cần được áp dụng cho khối hiện tại, và có thể tạo ra thông tin về chế độ liên dự đoán chỉ thị chế độ hợp nhất, chế độ (hợp nhất) afin, hoặc chế độ hợp nhất khối con xác định được.

Thiết bị mã hoá có thể tạo ra danh sách ứng viên hợp nhất của khối hiện tại dựa trên chế độ liên dự đoán (S1010). Ví dụ, thiết bị mã hoá có thể tạo ra danh sách ứng viên hợp nhất theo chế độ liên dự đoán xác định được. Ở đây, khi chế độ liên dự đoán xác định được là chế độ hợp nhất afin hoặc chế độ hợp nhất khối con, thì danh sách ứng viên hợp nhất có thể được gọi là danh sách ứng viên hợp nhất afin hoặc danh sách ứng viên hợp nhất khối con, nhưng cũng có thể được gọi đơn giản là danh sách ứng viên hợp nhất.

Ví dụ, các ứng viên có thể được chèn vào danh sách ứng viên hợp nhất cho đến khi số lượng ứng viên trong danh sách ứng viên hợp nhất trở thành số lượng ứng viên tối đa. Ở đây, ứng viên có thể chỉ thị ứng viên hoặc khối ứng viên để trích xuất thông tin chuyển động (hoặc vector chuyển động) của khối hiện tại. Ví dụ, khối ứng viên có thể được trích xuất thông qua việc tìm kiếm các khối lân cận của khối hiện tại. Ví dụ, khối lân cận có thể bao gồm khối lân cận về không gian và/hoặc khối lân cận về thời gian của khối hiện tại, khối lân cận về không gian có thể được ưu tiên tìm kiếm để trích xuất ứng viên (hợp nhất không gian), và sau đó khối lân cận về thời gian có thể được

tìm kiếm để trích xuất ứng viên (hợp nhất thời gian), và các ứng viên trích xuất được có thể được chèn vào danh sách ứng viên hợp nhất. Ví dụ, khi số lượng ứng viên trong danh sách ứng viên hợp nhất là nhỏ hơn số lượng ứng viên tối đa trong danh sách ứng viên hợp nhất ngay cả sau khi các ứng viên đã được chèn vào, thì các ứng viên bổ sung có thể được chèn vào. Ví dụ, các ứng viên bổ sung bao gồm ít nhất một trong số (các) ứng viên hợp nhất dựa trên lịch sử, (các) ứng viên hợp nhất trung bình theo cặp, ATMVP, và các ứng viên hợp nhất lưỡng dự đoán kết hợp (khi loại lát cắt/loại nhóm ô của lát cắt/nhóm ô hiện tại là loại B) và/hoặc ứng viên hợp nhất vector không.

Theo cách khác, ví dụ, các ứng viên có thể được chèn vào danh sách ứng viên hợp nhất afin cho đến khi số lượng ứng viên trong danh sách ứng viên hợp nhất afin trở thành số lượng ứng viên tối đa. Ở đây, ứng viên có thể bao gồm vector chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV) của khối hiện tại. Theo cách khác, ứng viên có thể chỉ thị ứng viên hoặc khối ứng viên để trích xuất CPMV. CPMV có thể chỉ thị vector chuyển động tại điểm kiểm soát (Control Point - CP) của khối hiện tại. Ví dụ, số lượng CP có thể là 2, 3, hoặc 4, CP có thể được đặt tại ít nhất một phần của đỉnh trái (hoặc góc đỉnh trái), đỉnh phải (hoặc góc đỉnh phải), đáy trái (hoặc góc đáy trái), hoặc đáy phải (hoặc góc đáy phải) của khối hiện tại, và chỉ một CP có thể tồn tại tại mỗi vị trí.

Ví dụ, ứng viên có thể được trích xuất thông qua việc tìm kiếm khối lân cận (hoặc khối lân cận của CP của khối hiện tại) của khối hiện tại. Ví dụ, danh sách ứng viên hợp nhất afin có thể bao gồm ít nhất một trong số ứng viên hợp nhất afin được kế thừa, ứng viên hợp nhất afin được xây dựng, và ứng viên vector chuyển động không. Ví dụ, trong danh sách ứng viên hợp nhất afin, thì ứng viên hợp nhất afin được kế thừa có thể được chèn vào trước, và sau đó ứng viên hợp nhất afin được xây dựng có thể được chèn vào. Ngoài ra, cho dù các ứng viên hợp nhất afin được xây dựng trong danh sách ứng viên hợp nhất afin được chèn vào, khi số lượng ứng viên trong danh sách ứng viên hợp nhất afin nhỏ hơn số lượng ứng viên tối đa, thì phần còn lại có thể được điền vào

bằng các ứng viên vector chuyển động không. Ở đây, ứng viên vector chuyển động không có thể được gọi là vector không. Ví dụ, danh sách ứng viên hợp nhất affin có thể là danh sách theo chế độ hợp nhất affin mà trong đó vector chuyển động được trích xuất theo các đơn vị là các mẫu, hoặc có thể là danh sách theo chế độ hợp nhất affin mà trong đó vector chuyển động được trích xuất theo các đơn vị là các khối con. Trong trường hợp này, danh sách ứng viên hợp nhất affin có thể được gọi là danh sách ứng viên hợp nhất khối con, và danh sách ứng viên hợp nhất khối con cũng có thể bao gồm các ứng viên (hoặc các ứng viên SbTMVP) được trích xuất từ SbTMVP. Ví dụ, khi ứng viên SbTMVP được bao gồm trong danh sách ứng viên hợp nhất khối con, thì nó có thể được đặt trước ứng viên hợp nhất affin được kế thừa và ứng viên hợp nhất affin được xây dựng trong danh sách ứng viên hợp nhất khối con.

Thiết bị mã hoá có thể tạo ra thông tin chọn chỉ thị một trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất (S1020). Ví dụ, danh sách ứng viên hợp nhất có thể bao gồm ít nhất một số trong số ứng viên hợp nhất không gian, ứng viên hợp nhất thời gian, ứng viên theo cặp, hoặc ứng viên vector không, và một trong số các ứng viên này có thể được chọn cho việc liên dự đoán khối hiện tại. Theo cách khác, ví dụ, danh sách ứng viên hợp nhất khối con có thể bao gồm ít nhất một số trong số ứng viên hợp nhất affin được kế thừa, ứng viên hợp nhất affin được xây dựng, ứng viên SbTMVP, hoặc ứng viên vector không, và một ứng viên trong số các ứng viên này, để liên dự đoán khối hiện tại, có thể được chọn.

Ví dụ, thông tin chọn có thể bao gồm thông tin chỉ số chỉ thị ứng viên được chọn trong danh sách ứng viên hợp nhất. Ví dụ, thông tin chọn có thể được gọi là thông tin chỉ số hợp nhất hoặc thông tin chỉ số hợp nhất khối con.

Thiết bị mã hoá có thể tạo ra thông tin về kiểu liên dự đoán chỉ thị kiểu liên dự đoán của khối hiện tại là lưỡng dự đoán (S1030). Ví dụ, kiểu liên dự đoán của khối hiện tại có thể được xác định là lưỡng dự đoán trong số dự đoán L0, dự đoán L1, hoặc lưỡng dự đoán, và thông tin về kiểu liên dự đoán chỉ thị việc này có thể được tạo ra. Ở đây,

việc dự đoán L0 có thể chỉ thị việc dự đoán dựa trên danh sách ảnh tham chiếu 0, việc dự đoán L1 có thể chỉ thị việc dự đoán dựa trên danh sách ảnh tham chiếu 1, và việc lưỡng dự đoán có thể chỉ thị việc dự đoán dựa trên danh sách ảnh tham chiếu 0 và danh sách ảnh tham chiếu 1. Ví dụ, thiết bị mã hoá có thể tạo ra thông tin về kiểu liên dự đoán dựa trên kiểu liên dự đoán. Ví dụ, thông tin về kiểu liên dự đoán có thể bao gồm phần tử cú pháp `inter_pred_idc`.

Thiết bị mã hoá có thể mã hoá thông tin hình ảnh bao gồm thông tin về chế độ liên dự đoán, thông tin chọn, và thông tin về kiểu liên dự đoán (S1040). Ví dụ, thông tin hình ảnh có thể được gọi là thông tin video. Thông tin hình ảnh có thể bao gồm các thông tin khác nhau theo (các) phương án nêu trên của sáng chế. Ví dụ, thông tin hình ảnh có thể bao gồm ít nhất một phần của thông tin liên quan đến việc dự đoán hoặc thông tin liên quan đến phần dư. Ví dụ, thông tin liên quan đến việc dự đoán có thể bao gồm ít nhất một phần của thông tin về chế độ liên dự đoán, thông tin chọn, và thông tin về kiểu liên dự đoán. Ví dụ, thiết bị mã hoá có thể tạo ra luồng bit hoặc thông tin được mã hoá bằng cách mã hoá thông tin hình ảnh bao gồm tất cả hoặc một phần của các thông tin (hoặc các phần tử cú pháp) được mô tả trên đây. Theo cách khác, nó có thể được xuất ra dưới dạng luồng bit. Ngoài ra, luồng bit hoặc thông tin được mã hoá này có thể được truyền đến thiết bị giải mã qua mạng hoặc phương tiện lưu trữ.

Tuy không được thể hiện trên Fig.10, nhưng, ví dụ, thiết bị mã hoá có thể tạo ra các mẫu dự đoán của khối hiện tại. Theo cách khác, ví dụ, thiết bị mã hoá có thể tạo ra các mẫu dự đoán của khối hiện tại dựa trên ứng viên được chọn. Theo cách khác, ví dụ, thiết bị mã hoá có thể trích xuất thông tin chuyển động dựa trên ứng viên được chọn, và có thể tạo ra các mẫu dự đoán của khối hiện tại dựa trên thông tin chuyển động. Ví dụ, thiết bị mã hoá có thể tạo ra L0 mẫu dự đoán và L1 mẫu dự đoán theo việc lưỡng dự đoán, và có thể tạo ra các mẫu dự đoán của khối hiện tại dựa trên L0 mẫu dự đoán và L1 mẫu dự đoán này. Trong trường hợp này, các mẫu dự đoán của khối hiện tại có thể được tạo ra từ L0 mẫu dự đoán và L1 mẫu dự đoán nhờ sử dụng thông tin chỉ số trọng

số (hoặc thông tin trọng số) dành cho việc lưỡng dự đoán. Ở đây, thông tin trọng số có thể được hiển thị dựa trên thông tin chỉ số trọng số.

Nói cách khác, ví dụ, thiết bị mã hoá có thể tạo ra L0 mẫu dự đoán và L1 mẫu dự đoán của khối hiện tại dựa trên ứng viên được chọn. Ví dụ, khi kiểu liên dự đoán của khối hiện tại được xác định là lưỡng dự đoán, thì danh sách ảnh tham chiếu 0 và danh sách ảnh tham chiếu 1 có thể được sử dụng cho việc dự đoán khối hiện tại. Ví dụ, L0 mẫu dự đoán có thể biểu thị các mẫu dự đoán của khối hiện tại mà được trích xuất dựa trên danh sách ảnh tham chiếu 0, và L1 mẫu dự đoán có thể biểu thị các mẫu dự đoán của khối hiện tại mà được trích xuất dựa trên danh sách ảnh tham chiếu 1.

Ví dụ, các ứng viên có thể bao gồm ứng viên hợp nhất không gian. Ví dụ, khi ứng viên được chọn là ứng viên hợp nhất không gian, thì thông tin chuyển động L0 và thông tin chuyển động L1 có thể được trích xuất dựa trên ứng viên hợp nhất không gian này, và L0 mẫu dự đoán và L1 mẫu dự đoán được tạo ra dựa trên đó.

Ví dụ, các ứng viên có thể bao gồm ứng viên hợp nhất thời gian. Ví dụ, khi ứng viên được chọn là ứng viên hợp nhất thời gian, thì thông tin chuyển động L0 và thông tin chuyển động L1 có thể được trích xuất dựa trên ứng viên hợp nhất thời gian này, và L0 mẫu dự đoán và L1 mẫu dự đoán được tạo ra dựa trên đó.

Ví dụ, các ứng viên có thể bao gồm các ứng viên theo cặp. Ví dụ, khi ứng viên được chọn là ứng viên theo cặp, thì thông tin chuyển động L0 và thông tin chuyển động L1 có thể được trích xuất dựa trên ứng viên theo cặp này, và L0 mẫu dự đoán và L1 mẫu dự đoán có thể được tạo ra dựa trên đó. Ví dụ, ứng viên theo cặp có thể được trích xuất dựa trên hai ứng viên khác trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất.

Theo cách khác, ví dụ, danh sách ứng viên hợp nhất có thể là danh sách ứng viên hợp nhất khối con, và ứng viên hợp nhất afin, ứng viên hợp nhất khối con, hoặc ứng viên SbTMVP, có thể được chọn. Ở đây, ứng viên hợp nhất afin theo các đơn vị là các khối con có thể được gọi là ứng viên hợp nhất khối con.

Ví dụ, các ứng viên có thể bao gồm ứng viên hợp nhất khối con. Ví dụ, khi ứng viên được chọn là ứng viên hợp nhất khối con, thì thông tin chuyển động L0 và thông tin chuyển động L1 có thể được trích xuất dựa trên ứng viên hợp nhất khối con này, và L0 mẫu dự đoán và L1 mẫu dự đoán được tạo ra dựa trên đó. Ví dụ, ứng viên hợp nhất khối con có thể bao gồm các vector chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV), và L0 mẫu dự đoán và L1 mẫu dự đoán có thể được tạo ra bằng cách thực hiện việc dự đoán theo các đơn vị là khối con dựa trên các CPMV này.

Ở đây, CPMV có thể được chỉ thị dựa trên một khối trong số các khối lân cận của điểm kiểm soát (Control Point - CP) của khối hiện tại. Ví dụ, số lượng CP có thể là 2, 3, hoặc 4, CP có thể được đặt tại ít nhất một phần của đỉnh trái (hoặc góc đỉnh trái), đỉnh phải (hoặc góc đỉnh phải), đáy trái (hoặc góc đáy trái), hoặc đáy phải (hoặc góc đáy phải) của khối hiện tại, và chỉ một CP có thể tồn tại tại mỗi vị trí.

Ví dụ, CP có thể là CP0 được đặt tại đỉnh trái của khối hiện tại. Trong trường hợp này, các khối lân cận có thể bao gồm khối lân cận ở góc đỉnh trái của khối hiện tại, khối lân cận đáy trái kề với đáy của khối lân cận ở góc đỉnh trái, và khối lân cận trên đỉnh kề với bên phải của khối lân cận ở góc đỉnh trái. Theo cách khác, các khối lân cận có thể bao gồm khối A₂, khối B₂, hoặc khối B₃ trên Fig.8.

Theo cách khác, ví dụ, CP có thể là CP1 được đặt tại đỉnh phải của khối hiện tại. Trong trường hợp này, các khối lân cận có thể bao gồm khối lân cận ở góc đỉnh phải của khối hiện tại và khối lân cận trên đỉnh kề với bên trái của khối lân cận ở góc đỉnh phải. Theo cách khác, các khối lân cận có thể bao gồm khối B₀ hoặc khối B₁ trên Fig.8.

Theo cách khác, ví dụ, CP có thể là CP2 được đặt tại đáy trái của khối hiện tại. Trong trường hợp này, các khối lân cận có thể bao gồm khối lân cận ở góc đáy trái của khối hiện tại và khối lân cận bên trái kề với đỉnh của khối lân cận ở góc đáy trái. Theo cách khác, các khối lân cận có thể bao gồm khối A₀ hoặc khối A₁ trên Fig.8.

Theo cách khác, ví dụ, CP có thể là CP3 được đặt tại đáy phải của khối hiện tại. Ở đây, CP3 cũng có thể được gọi là RB. Trong trường hợp này, các khối lân cận có thể

bao gồm khối cùng vị trí của khối hiện tại hoặc khối lân cận ở góc đáy phải của khối được đặt cùng vị trí. Ở đây, khối được đặt cùng vị trí có thể bao gồm khối tại cùng vị trí với khối hiện tại trong ảnh tham chiếu khác với ảnh hiện tại mà khối hiện tại được đặt trong đó. Theo cách khác, khối lân cận có thể bao gồm khối T trên Fig.8.

Theo cách khác, ví dụ, các ứng viên có thể bao gồm các ứng viên SbTMVP. Ví dụ, khi ứng viên được chọn là ứng viên SbTMVP, thì thông tin chuyển động L0 và thông tin chuyển động L1 có thể được trích xuất dựa trên khối lân cận bên trái của khối hiện tại, và dựa trên đó, L0 mẫu dự đoán và L1 mẫu dự đoán có thể được tạo ra. Ví dụ, L0 mẫu dự đoán và L1 mẫu dự đoán có thể được tạo ra bằng cách thực hiện việc dự đoán theo các đơn vị là các khối con.

Ví dụ, thông tin chuyển động L0 có thể bao gồm chỉ số ảnh tham chiếu L0, vector chuyển động L0, v.v., và thông tin chuyển động L1 có thể bao gồm chỉ số ảnh tham chiếu L1, vector chuyển động L1, v.v.. Chỉ số ảnh tham chiếu L0 có thể bao gồm thông tin chỉ thị ảnh tham chiếu trong danh sách ảnh tham chiếu 0, và chỉ số ảnh tham chiếu L1 có thể bao gồm thông tin chỉ thị ảnh tham chiếu trong danh sách ảnh tham chiếu 1.

Ví dụ, thiết bị mã hoá có thể tạo ra các mẫu dự đoán của khối hiện tại dựa trên L0 mẫu dự đoán, L1 mẫu dự đoán, và thông tin trọng số. Ví dụ, thông tin trọng số có thể được hiển thị dựa trên thông tin chỉ số trọng số. Thông tin chỉ số trọng số có thể chỉ thị thông tin chỉ số trọng số trên việc lưỡng dự đoán. Ví dụ, thông tin trọng số có thể bao gồm thông tin về trung bình được lấy trọng số của L0 mẫu dự đoán hoặc L1 mẫu dự đoán. Tức là, thông tin chỉ số trọng số có thể chỉ thị thông tin chỉ số trên trọng số được sử dụng cho trung bình được lấy trọng số, và có thể tạo ra thông tin chỉ số trọng số trong thủ tục tạo ra các mẫu dự đoán dựa trên trung bình được lấy trọng số này. Ví dụ, thông tin chỉ số trọng số có thể bao gồm thông tin chỉ thị bất kỳ trong số ba hoặc năm trọng số. Ví dụ, trung bình được lấy trọng số có thể biểu thị trung bình được lấy trọng số trong việc lưỡng dự đoán với trọng số mức độ CU (Bi-prediction with CU-

level Weight - BCW) hoặc việc lưỡng dự đoán với trung bình được lấy trọng số (Bi-prediction with Weighted Average - BWA).

Ví dụ, các ứng viên có thể bao gồm ứng viên hợp nhất thời gian, và thông tin chỉ số trọng số trên ứng viên hợp nhất thời gian có thể được biểu diễn bởi 0. Tức là, thông tin chỉ số trọng số trên ứng viên hợp nhất thời gian có thể được biểu diễn bởi 0. Ở đây, thông tin chỉ số trọng số là 0 có thể có nghĩa là việc các trọng số của mỗi chiều tham chiếu (tức là chiều dự đoán L0 và chiều dự đoán L1 trong quá trình lưỡng dự đoán) là giống nhau. Theo cách khác, ví dụ, các ứng viên có thể bao gồm ứng viên hợp nhất thời gian, và thông tin chỉ số trọng số có thể được chỉ thị dựa trên thông tin chỉ số trọng số trên khối cùng vị trí. Tức là, thông tin chỉ số trọng số trên ứng viên hợp nhất thời gian có thể được chỉ thị dựa trên thông tin chỉ số trọng số trên khối cùng vị trí. Ở đây, khối được đặt cùng vị trí có thể bao gồm khối tại cùng vị trí với khối hiện tại trong ảnh tham chiếu khác với ảnh hiện tại mà khối hiện tại được đặt trong đó.

Ví dụ, các ứng viên có thể bao gồm ứng viên theo cặp, và thông tin chỉ số trọng số có thể được chỉ thị dựa trên thông tin chỉ số trọng số trên một trong số hai ứng viên khác đó trong danh sách ứng viên hợp nhất mà được dùng để trích xuất ứng viên theo cặp. Tức là, thông tin chỉ số trọng số trên ứng viên theo cặp có thể được chỉ thị dựa trên thông tin chỉ số trọng số trên một trong số hai ứng viên khác đó trong danh sách ứng viên hợp nhất mà được dùng để trích xuất ứng viên theo cặp.

Ví dụ, các ứng viên có thể bao gồm ứng viên theo cặp, và ứng viên theo cặp có thể được chỉ thị dựa trên hai ứng viên khác trong số các ứng viên. Khi thông tin chỉ số trọng số trên mỗi trong số hai ứng viên khác đó là giống như thông tin chỉ số trọng số thứ nhất, thì thông tin chỉ số trọng số trên ứng viên theo cặp có thể được chỉ thị dựa trên thông tin chỉ số trọng số thứ nhất, khi thông tin chỉ số trọng số trên mỗi trong số hai ứng viên khác đó là không giống nhau, thì thông tin chỉ số trọng số trên ứng viên theo cặp có thể được chỉ thị dựa trên thông tin chỉ số trọng số mặc định, và trong trường hợp này, thông tin chỉ số trọng số mặc định có thể tương ứng với thông tin chỉ số trọng số trên

việc cho trọng số giống nhau cho mỗi trong số L0 mẫu dự đoán và L1 mẫu dự đoán.

Ví dụ, các ứng viên có thể bao gồm ứng viên theo cặp, và ứng viên theo cặp có thể được chỉ thị dựa trên hai ứng viên khác trong số các ứng viên. Khi thông tin chỉ số trọng số trên mỗi trong số hai ứng viên khác đó là giống như thông tin chỉ số trọng số thứ nhất, thì thông tin chỉ số trọng số trên ứng viên theo cặp có thể được chỉ thị dựa trên thông tin chỉ số trọng số thứ nhất. Khi thông tin chỉ số trọng số trên mỗi trong số hai ứng viên khác đó không giống nhau, thì thông tin chỉ số trọng số có thể được chỉ thị dựa trên thông tin chỉ số trọng số thay vì thông tin chỉ số trọng số mặc định trong số các thông tin chỉ số trọng số của mỗi trong số hai ứng viên khác đó. Thông tin chỉ số trọng số mặc định có thể tương ứng với thông tin chỉ số trọng số trên việc cho trọng số giống nhau cho mỗi trong số L0 mẫu dự đoán và L1 mẫu dự đoán.

Ví dụ, danh sách ứng viên hợp nhất có thể là danh sách ứng viên hợp nhất khối con, và ứng viên hợp nhất afin, ứng viên hợp nhất khối con, hoặc ứng viên SbTMVP, là có thể được chọn. Ở đây, ứng viên hợp nhất afin theo các đơn vị là các khối con có thể được gọi là ứng viên hợp nhất khối con.

Ví dụ, các ứng viên bao gồm ứng viên hợp nhất afin, và ứng viên hợp nhất afin có thể bao gồm các vector chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV).

Ví dụ, khi ứng viên hợp nhất afin bao gồm CPMV dành cho điểm kiểm soát 0 (CP0) được đặt ở đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất afin có thể được chỉ thị dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của CP0. Khi ứng viên hợp nhất afin không bao gồm CPMV dành cho CP0 được đặt tại đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất afin có thể được chỉ thị dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của điểm kiểm soát 1 (CP1) được đặt ở vị trí đỉnh phải của khối hiện tại.

Khối cụ thể trong số các khối lân cận của CP0 là tương ứng với khối được sử

dụng để trích xuất CPMV cho CP0, và các khối lân cận của CP0 có thể bao gồm khối lân cận ở góc đỉnh trái của khối hiện tại, khối lân cận bên trái kề với đáy của khối lân cận ở góc đỉnh trái, và khối lân cận trên đỉnh kề với bên phải của khối lân cận ở góc đỉnh trái.

Trong số các khối lân cận của CP1, thì khối cụ thể là tương ứng với khối được sử dụng để trích xuất CPMV dành cho CP1, và các khối lân cận CP1 có thể bao gồm khối lân cận ở góc đỉnh phải của khối hiện tại và khối lân cận trên đỉnh kề với bên trái của khối lân cận ở góc đỉnh phải.

Theo cách khác, ví dụ, các ứng viên có thể bao gồm ứng viên SbTMVP, và thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được chỉ thị dựa trên thông tin chỉ số trọng số trên khối lân cận bên trái của khối hiện tại. Tức là, thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được chỉ thị dựa trên thông tin chỉ số trọng số trên khối lân cận bên trái.

Theo cách khác, ví dụ, các ứng viên có thể bao gồm ứng viên SbTMVP, và thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được biểu diễn bởi 0. Tức là, thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được biểu diễn bởi 0. Ở đây, thông tin chỉ số trọng số là 0 có thể có nghĩa là việc các trọng số của mỗi chiều tham chiếu (tức là chiều dự đoán L0 và chiều dự đoán L1 trong quá trình lưỡng dự đoán) là giống nhau.

Theo cách khác, ví dụ, các ứng viên có thể bao gồm ứng viên SbTMVP, và thông tin chỉ số trọng số có thể được chỉ thị dựa trên thông tin chỉ số trọng số trên khối trung tâm ở khối cùng vị trí. Tức là, thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được chỉ thị dựa trên thông tin chỉ số trọng số trên khối trung tâm ở khối cùng vị trí. Ở đây, khối được đặt cùng vị trí có thể bao gồm khối tại cùng vị trí với khối hiện tại trong ảnh tham chiếu khác với ảnh hiện tại mà khối hiện tại được đặt trong đó, và khối trung tâm có thể bao gồm khối con ở đáy phải trong số bốn khối con được đặt ở tâm của khối được đặt cùng vị trí.

Theo cách khác, ví dụ, các ứng viên có thể bao gồm ứng viên SbTMVP, và thông tin chỉ số trọng số có thể được chỉ thị dựa trên thông tin chỉ số trọng số trên mỗi trong số các khối con ở khối cùng vị trí. Tức là, thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được chỉ thị dựa trên thông tin chỉ số trọng số trên mỗi trong số các khối con của khối cùng vị trí.

Theo cách khác, mặc dù không được thể hiện trên Fig.10, nhưng, ví dụ, thiết bị mã hoá này có thể trích xuất các mẫu dư dựa trên các mẫu dự đoán và các mẫu gốc. Trong trường hợp này, thông tin liên quan đến phần dư có thể được trích xuất dựa trên các mẫu dư. Các mẫu dư có thể được trích xuất dựa trên thông tin liên quan đến phần dư. Các mẫu được tái tạo có thể được tạo ra dựa trên các mẫu dư và các mẫu dự đoán. Khối được tái tạo và ảnh được tái tạo có thể được trích xuất dựa trên các mẫu được tái tạo. Theo cách khác, ví dụ, thiết bị mã hoá này có thể mã hoá thông tin hình ảnh bao gồm thông tin liên quan đến phần dư hoặc thông tin liên quan đến việc dự đoán.

Ví dụ, thiết bị mã hoá có thể tạo ra luồng bit hoặc thông tin được mã hoá bằng cách mã hoá thông tin hình ảnh bao gồm tất cả hoặc một phần của các thông tin (hoặc các phần tử cú pháp) được mô tả trên đây. Theo cách khác, nó có thể được xuất ra dưới dạng luồng bit. Ngoài ra, luồng bit hoặc thông tin được mã hoá này có thể được truyền đến thiết bị giải mã qua mạng hoặc phương tiện lưu trữ. Theo cách khác, luồng bit hoặc thông tin được mã hoá này có thể được lưu giữ trên phương tiện lưu trữ đọc được bằng máy tính, và luồng bit hoặc thông tin được mã hoá này có thể được tạo ra bởi phương pháp mã hoá ảnh được mô tả trên đây.

Fig.12 và Fig.13 là các hình vẽ thể hiện sơ đồ ví dụ của phương pháp giải mã video/ảnh và các thành phần liên quan theo (các) phương án thực hiện của sáng chế.

Phương pháp được bộc lộ trên Fig.12 có thể được thực hiện bởi thiết bị giải mã được bộc lộ trên Fig.3 hoặc Fig.13. Cụ thể là, ví dụ, bước S1200 trên Fig.12 có thể được thực hiện bởi bộ giải mã entropy 310 của thiết bị giải mã 300 trên Fig.13, và bước S1210 đến S1260 trên Fig.12 là được thực hiện bởi bộ dự đoán 330 của thiết bị giải mã 300

trên Fig.13. Ngoài ra, mặc dù không được thể hiện trên Fig.12, nhưng trên Fig.13, thì thông tin liên quan đến việc dự đoán hoặc thông tin dư là có thể được trích xuất từ luồng bit bởi bộ giải mã entropy 310 của thiết bị giải mã 300, các mẫu dư có thể được trích xuất từ thông tin dư bởi bộ xử lý phần dư 320 của thiết bị giải mã 300, các mẫu dự đoán có thể được trích xuất từ thông tin liên quan đến việc dự đoán bởi bộ dự đoán 330 của thiết bị giải mã 300, và khối được tái tạo hoặc ảnh được tái tạo là có thể được trích xuất từ các mẫu dư hoặc các mẫu dự đoán bởi bộ cộng 340 của thiết bị giải mã 300. Phương pháp được bộc lộ trên Fig.12 có thể bao gồm các phương án được mô tả trên đây trong bản mô tả này.

Như được thể hiện trên Fig.12, thiết bị giải mã này có thể nhận thông tin hình ảnh bao gồm thông tin về chế độ liên dự đoán và thông tin về kiểu liên dự đoán thông qua luồng bit (S1200). Ví dụ, thông tin hình ảnh có thể được gọi là thông tin video. Thông tin hình ảnh có thể bao gồm các thông tin khác nhau theo (các) phương án nêu trên của sáng chế. Ví dụ, thông tin hình ảnh có thể bao gồm ít nhất một phần của thông tin liên quan đến việc dự đoán hoặc thông tin liên quan đến phần dư.

Ví dụ, thông tin liên quan đến việc dự đoán có thể bao gồm thông tin về chế độ liên dự đoán hoặc thông tin về kiểu liên dự đoán. Ví dụ, thông tin về chế độ liên dự đoán có thể bao gồm thông tin chỉ thị ít nhất một số trong số các chế độ liên dự đoán khác nhau. Ví dụ, các chế độ khác nhau chẳng hạn như chế độ hợp nhất, chế độ bỏ cóc, chế độ dự đoán vector chuyển động (Motion Vector Prediction - MVP), chế độ afin, chế độ hợp nhất khối con, hoặc chế độ hợp nhất với MVD (Merge with MVD - MMVD), là có thể được sử dụng. Ngoài ra, chế độ tinh chỉnh vector chuyển động phía bộ giải mã (Decoder side Motion Vector Refinement - DMVR), chế độ phân giải vector chuyển động thích ứng (Adaptive Motion Vector Resolution - AMVR), lưỡng dự đoán với trọng số mức độ CU, hoặc luồng quang học hai chiều (Bi-Directional Optical Flow - BDOF), v.v., là có thể được sử dụng bổ sung hoặc thay thế dưới dạng các chế độ phụ trợ. Ví dụ, thông tin về kiểu liên dự đoán có thể bao gồm phần tử cú pháp `inter_pred_idc`. Theo

cách khác, thông tin về kiểu liên dự đoán có thể bao gồm thông tin chỉ thị bất kỳ trong số việc dự đoán L0, việc dự đoán L1, và việc lưỡng dự đoán.

Thiết bị giải mã này có thể tạo ra danh sách ứng viên hợp nhất của khối hiện tại dựa trên thông tin về chế độ liên dự đoán (S1210). Ví dụ, thiết bị giải mã này có thể xác định chế độ liên dự đoán của khối hiện tại là chế độ hợp nhất, chế độ (hợp nhất) afin, hoặc chế độ hợp nhất khối con dựa trên thông tin về chế độ liên dự đoán, và tạo ra danh sách ứng viên hợp nhất theo chế độ liên dự đoán xác định được. Ở đây, khi chế độ liên dự đoán là chế độ hợp nhất afin hoặc chế độ hợp nhất khối con, thì danh sách ứng viên hợp nhất có thể được gọi là danh sách ứng viên hợp nhất afin hoặc danh sách ứng viên hợp nhất khối con, nhưng cũng có thể được gọi đơn giản là danh sách ứng viên hợp nhất.

Ví dụ, các ứng viên có thể được chèn vào danh sách ứng viên hợp nhất cho đến khi số lượng ứng viên trong danh sách ứng viên hợp nhất trở thành số lượng ứng viên tối đa. Ở đây, ứng viên có thể chỉ thị ứng viên hoặc khối ứng viên để trích xuất thông tin chuyển động (hoặc vector chuyển động) của khối hiện tại. Ví dụ, khối ứng viên có thể được trích xuất thông qua việc tìm kiếm các khối lân cận của khối hiện tại. Ví dụ, khối lân cận có thể bao gồm khối lân cận về không gian và/hoặc khối lân cận về thời gian của khối hiện tại, khối lân cận về không gian có thể được ưu tiên tìm kiếm để trích xuất ứng viên (hợp nhất không gian), và sau đó khối lân cận về thời gian có thể được tìm kiếm để trích xuất ứng viên (hợp nhất thời gian), và các ứng viên trích xuất được có thể được chèn vào danh sách ứng viên hợp nhất. Ví dụ, khi số lượng ứng viên trong danh sách ứng viên hợp nhất là nhỏ hơn số lượng ứng viên tối đa trong danh sách ứng viên hợp nhất ngay cả sau khi các ứng viên đã được chèn vào, thì các ứng viên bổ sung có thể được chèn vào. Ví dụ, các ứng viên bổ sung bao gồm ít nhất một trong số (các) ứng viên hợp nhất dựa trên lịch sử, (các) ứng viên hợp nhất trung bình theo cặp, ATMVP, và các ứng viên hợp nhất lưỡng dự đoán kết hợp (khi loại lát cắt/loại nhóm ô của lát cắt/nhóm ô hiện tại là loại B) và/hoặc ứng viên hợp nhất vector không.

Theo cách khác, ví dụ, các ứng viên có thể được chèn vào danh sách ứng viên hợp nhất afin cho đến khi số lượng ứng viên trong danh sách ứng viên hợp nhất afin trở thành số lượng ứng viên tối đa. Ở đây, ứng viên có thể bao gồm vector chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV) của khối hiện tại. Theo cách khác, ứng viên có thể chỉ thị ứng viên hoặc khối ứng viên để trích xuất CPMV. CPMV có thể chỉ thị vector chuyển động tại điểm kiểm soát (Control Point - CP) của khối hiện tại. Ví dụ, số lượng CP có thể là 2, 3, hoặc 4, CP có thể được đặt tại ít nhất một phần của đỉnh trái (hoặc góc đỉnh trái), đỉnh phải (hoặc góc đỉnh phải), đáy trái (hoặc góc đáy trái), hoặc đáy phải (hoặc góc đáy phải) của khối hiện tại, và chỉ một CP có thể tồn tại tại mỗi vị trí.

Ví dụ, khối ứng viên có thể được trích xuất thông qua việc tìm kiếm khối lân cận (hoặc khối lân cận của CP của khối hiện tại) của khối hiện tại. Ví dụ, danh sách ứng viên hợp nhất afin có thể bao gồm ít nhất một trong số ứng viên hợp nhất afin được kế thừa, ứng viên hợp nhất afin được xây dựng, và ứng viên vector chuyển động không. Ví dụ, trong danh sách ứng viên hợp nhất afin, thì ứng viên hợp nhất afin được kế thừa có thể được chèn vào trước, và sau đó ứng viên hợp nhất afin được xây dựng có thể được chèn vào. Ngoài ra, cho dù các ứng viên hợp nhất afin được xây dựng trong danh sách ứng viên hợp nhất afin được chèn vào, khi số lượng ứng viên trong danh sách ứng viên hợp nhất afin nhỏ hơn số lượng ứng viên tối đa, thì phần còn lại có thể được điền vào bằng các ứng viên vector chuyển động không. Ở đây, ứng viên vector chuyển động không có thể được gọi là vector không. Ví dụ, danh sách ứng viên hợp nhất afin có thể là danh sách theo chế độ hợp nhất afin mà trong đó vector chuyển động được trích xuất theo các đơn vị là các mẫu, hoặc có thể là danh sách theo chế độ hợp nhất afin mà trong đó vector chuyển động được trích xuất theo các đơn vị là các khối con. Trong trường hợp này, danh sách ứng viên hợp nhất afin có thể được gọi là danh sách ứng viên hợp nhất khối con, và danh sách ứng viên hợp nhất khối con cũng có thể bao gồm các ứng viên (hoặc các ứng viên SbTMVP) được trích xuất từ SbTMVP. Ví dụ, khi ứng viên SbTMVP

được bao gồm trong danh sách ứng viên hợp nhất khối con, thì nó có thể được đặt trước ứng viên hợp nhất afin được kế thừa và ứng viên hợp nhất afin được xây dựng trong danh sách ứng viên hợp nhất khối con.

Thiết bị giải mã này có thể tạo ra thông tin chọn chỉ thị một trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất (S1220). Ví dụ, danh sách ứng viên hợp nhất có thể bao gồm ít nhất một số trong số ứng viên hợp nhất không gian, ứng viên hợp nhất thời gian, ứng viên theo cặp, hoặc ứng viên vector không, và một trong số các ứng viên này có thể được chọn cho việc liên dự đoán khối hiện tại. Theo cách khác, ví dụ, danh sách ứng viên hợp nhất khối con có thể bao gồm ít nhất một số trong số ứng viên hợp nhất afin được kế thừa, ứng viên hợp nhất afin được xây dựng, ứng viên SbTMVP, hoặc ứng viên vector không, và một ứng viên trong số các ứng viên này, để liên dự đoán khối hiện tại, có thể được chọn. Ví dụ, ứng viên được chọn có thể được chọn từ danh sách ứng viên hợp nhất dựa trên thông tin chọn. Ví dụ, thông tin chọn có thể bao gồm thông tin chỉ số chỉ thị ứng viên được chọn trong danh sách ứng viên hợp nhất. Ví dụ, thông tin chọn có thể được gọi là thông tin chỉ số hợp nhất hoặc thông tin chỉ số hợp nhất khối con. Ví dụ, thông tin chọn có thể được bao gồm trong thông tin hình ảnh. Theo cách khác, thông tin chọn có thể được bao gồm trong thông tin về chế độ liên dự đoán.

Thiết bị giải mã này có thể trích xuất kiểu liên dự đoán của khối hiện tại là lưỡng dự đoán dựa trên thông tin về kiểu liên dự đoán (S1230). Ví dụ, kiểu liên dự đoán của khối hiện tại có thể được trích xuất là lưỡng dự đoán trong số dự đoán L0, dự đoán L1, hoặc lưỡng dự đoán dựa trên thông tin về kiểu liên dự đoán. Ở đây, việc dự đoán L0 có thể chỉ thị việc dự đoán dựa trên danh sách ảnh tham chiếu 0, việc dự đoán L1 có thể chỉ thị việc dự đoán dựa trên danh sách ảnh tham chiếu 1, và việc lưỡng dự đoán có thể chỉ thị việc dự đoán dựa trên danh sách ảnh tham chiếu 0 và danh sách ảnh tham chiếu 1. Ví dụ, thông tin về kiểu liên dự đoán có thể bao gồm phần tử cú pháp `inter_pred_idc`.

Thiết bị giải mã này có thể trích xuất thông tin chuyển động trên khối hiện tại

dựa trên ứng viên được chọn (S1240). Ví dụ, thiết bị giải mã này có thể trích xuất thông tin chuyển động L0 và thông tin chuyển động L1 dựa trên ứng viên được chọn khi kiểu liên dự đoán được trích xuất là lưỡng dự đoán. Ví dụ, thông tin chuyển động L0 có thể bao gồm chỉ số ảnh tham chiếu L0, vector chuyển động L0, v.v., và thông tin chuyển động L1 có thể bao gồm chỉ số ảnh tham chiếu L1, vector chuyển động L1, v.v.. Chỉ số ảnh tham chiếu L0 có thể bao gồm thông tin chỉ thị ảnh tham chiếu trong danh sách ảnh tham chiếu 0, và chỉ số ảnh tham chiếu L1 có thể bao gồm thông tin chỉ thị ảnh tham chiếu trong danh sách ảnh tham chiếu 1.

Thiết bị giải mã này có thể tạo ra L0 mẫu dự đoán và L1 mẫu dự đoán của khối hiện tại dựa trên thông tin chuyển động này (S1250). Ví dụ, khi kiểu liên dự đoán của khối hiện tại được trích xuất là lưỡng dự đoán, thì danh sách ảnh tham chiếu 0 và danh sách ảnh tham chiếu 1 có thể được sử dụng cho việc dự đoán khối hiện tại. Ví dụ, L0 mẫu dự đoán có thể biểu thị các mẫu dự đoán của khối hiện tại mà được trích xuất dựa trên danh sách ảnh tham chiếu 0, và L1 mẫu dự đoán có thể biểu thị các mẫu dự đoán của khối hiện tại mà được trích xuất dựa trên danh sách ảnh tham chiếu 1.

Ví dụ, các ứng viên có thể bao gồm ứng viên hợp nhất không gian. Ví dụ, khi ứng viên được chọn là ứng viên hợp nhất không gian, thì thông tin chuyển động L0 và thông tin chuyển động L1 có thể được trích xuất dựa trên ứng viên hợp nhất không gian này, và L0 mẫu dự đoán và L1 mẫu dự đoán được tạo ra dựa trên đó.

Ví dụ, các ứng viên có thể bao gồm ứng viên hợp nhất thời gian. Ví dụ, khi ứng viên được chọn là ứng viên hợp nhất thời gian, thì thông tin chuyển động L0 và thông tin chuyển động L1 có thể được trích xuất dựa trên ứng viên hợp nhất thời gian này, và L0 mẫu dự đoán và L1 mẫu dự đoán được tạo ra dựa trên đó.

Ví dụ, các ứng viên có thể bao gồm các ứng viên theo cặp. Ví dụ, khi ứng viên được chọn là ứng viên theo cặp, thì thông tin chuyển động L0 và thông tin chuyển động L1 có thể được trích xuất dựa trên ứng viên theo cặp này, và L0 mẫu dự đoán và L1 mẫu dự đoán có thể được tạo ra dựa trên đó. Ví dụ, ứng viên theo cặp có thể được trích

xuất dựa trên hai ứng viên khác trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất.

Theo cách khác, ví dụ, danh sách ứng viên hợp nhất có thể là danh sách ứng viên hợp nhất khối con, và ứng viên hợp nhất afin, ứng viên hợp nhất khối con, hoặc ứng viên SbTMVP, có thể được chọn. Ở đây, ứng viên hợp nhất afin theo các đơn vị là các khối con có thể được gọi là ứng viên hợp nhất khối con.

Ví dụ, các ứng viên có thể bao gồm ứng viên hợp nhất afin. Ví dụ, khi ứng viên được chọn là ứng viên hợp nhất afin, thì thông tin chuyển động L0 và thông tin chuyển động L1 có thể được trích xuất dựa trên ứng viên hợp nhất afin này, và L0 mẫu dự đoán và L1 mẫu dự đoán được tạo ra dựa trên đó. Ví dụ, ứng viên hợp nhất afin này có thể bao gồm các vectơ chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV), và L0 mẫu dự đoán và L1 mẫu dự đoán có thể được tạo ra bằng cách thực hiện việc dự đoán theo các đơn vị là khối con dựa trên các CPMV này.

Ở đây, CPMV có thể được trích xuất dựa trên một khối trong số các khối lân cận của điểm kiểm soát (Control Point - CP) của khối hiện tại. Ví dụ, số lượng CP có thể là 2, 3, hoặc 4, CP có thể được đặt tại ít nhất một phần của đỉnh trái (hoặc góc đỉnh trái), đỉnh phải (hoặc góc đỉnh phải), đáy trái (hoặc góc đáy trái), hoặc đáy phải (hoặc góc đáy phải) của khối hiện tại, và chỉ một CP có thể tồn tại tại mỗi vị trí.

Ví dụ, CP có thể là CP0 được đặt tại đỉnh trái của khối hiện tại. Trong trường hợp này, các khối lân cận có thể bao gồm khối lân cận ở góc đỉnh trái của khối hiện tại, khối lân cận đáy trái kề với đáy của khối lân cận ở góc đỉnh trái, và khối lân cận trên đỉnh kề với bên phải của khối lân cận ở góc đỉnh trái. Theo cách khác, các khối lân cận có thể bao gồm khối A2, khối B2, hoặc khối B3 trên Fig.8.

Theo cách khác, ví dụ, CP có thể là CP1 được đặt tại đỉnh phải của khối hiện tại. Trong trường hợp này, các khối lân cận có thể bao gồm khối lân cận ở góc đỉnh phải của khối hiện tại và khối lân cận trên đỉnh kề với bên trái của khối lân cận ở góc đỉnh phải. Theo cách khác, các khối lân cận có thể bao gồm khối B0 hoặc khối B1 trên Fig.8.

Theo cách khác, ví dụ, CP có thể là CP2 được đặt tại đáy trái của khối hiện tại. Trong trường hợp này, các khối lân cận có thể bao gồm khối lân cận ở góc đáy trái của khối hiện tại và khối lân cận bên trái kề với đỉnh của khối lân cận ở góc đáy trái. Theo cách khác, các khối lân cận có thể bao gồm khối A0 hoặc khối A1 trên Fig.8.

Theo cách khác, ví dụ, CP có thể là CP3 được đặt tại đáy phải của khối hiện tại. Ở đây, CP3 cũng có thể được gọi là RB. Trong trường hợp này, các khối lân cận có thể bao gồm khối cùng vị trí của khối hiện tại hoặc khối lân cận ở góc đáy phải của khối được đặt cùng vị trí. Ở đây, khối được đặt cùng vị trí có thể bao gồm khối tại cùng vị trí với khối hiện tại trong ảnh tham chiếu khác với ảnh hiện tại mà khối hiện tại được đặt trong đó. Theo cách khác, khối lân cận có thể bao gồm khối T trên Fig.8.

Theo cách khác, ví dụ, các ứng viên có thể bao gồm các ứng viên SbTMVP. Ví dụ, khi ứng viên được chọn là ứng viên SbTMVP, thì thông tin chuyển động L0 và thông tin chuyển động L1 có thể được trích xuất dựa trên khối lân cận bên trái của khối hiện tại, và dựa trên đó, L0 mẫu dự đoán và L1 mẫu dự đoán có thể được tạo ra. Ví dụ, L0 mẫu dự đoán và L1 mẫu dự đoán có thể được tạo ra bằng cách thực hiện việc dự đoán theo các đơn vị là các khối con.

Ví dụ, thiết bị giải mã này có thể tạo ra các mẫu dự đoán của khối hiện tại dựa trên L0 mẫu dự đoán, L1 mẫu dự đoán, và thông tin trọng số (S1260). Ví dụ, thông tin trọng số có thể được trích xuất dựa trên thông tin chỉ số trọng số trên ứng viên được chọn từ trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất. Ví dụ, thông tin trọng số có thể bao gồm thông tin về trung bình được lấy trọng số của L0 mẫu dự đoán hoặc L1 mẫu dự đoán. Tức là, thông tin chỉ số trọng số có thể chỉ thị thông tin chỉ số trên các trọng số được sử dụng cho trung bình được lấy trọng số, và trung bình được lấy trọng số này có thể được thực hiện dựa trên thông tin chỉ số trọng số. Ví dụ, thông tin chỉ số trọng số có thể bao gồm thông tin chỉ thị bất kỳ trong số ba hoặc năm trọng số. Ví dụ, trung bình được lấy trọng số có thể biểu thị trung bình được lấy trọng số trong việc lưỡng dự đoán với trọng số mức độ CU (Bi-prediction with CU-level

Weight - BCW) hoặc việc lưỡng dự đoán với trung bình được lấy trọng số (Bi-prediction with Weighted Average - BWA).

Ví dụ, các ứng viên có thể bao gồm ứng viên hợp nhất thời gian, và thông tin chỉ số trọng số trên ứng viên hợp nhất thời gian có thể được trích xuất là 0. Tức là, thông tin chỉ số trọng số trên ứng viên hợp nhất thời gian có thể được trích xuất là 0. Ở đây, thông tin chỉ số trọng số là 0 có thể có nghĩa là việc các trọng số của mỗi chiều tham chiếu (tức là chiều dự đoán L0 và chiều dự đoán L1 trong quá trình lưỡng dự đoán) là giống nhau.

Ví dụ, các ứng viên có thể bao gồm ứng viên hợp nhất thời gian, và thông tin chỉ số trọng số trên ứng viên hợp nhất thời gian có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối cùng vị trí. Tức là, thông tin chỉ số trọng số trên ứng viên hợp nhất thời gian có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối cùng vị trí. Ở đây, khối được đặt cùng vị trí có thể bao gồm khối tại cùng vị trí với khối hiện tại trong ảnh tham chiếu khác với ảnh hiện tại mà khối hiện tại được đặt trong đó.

Ví dụ, các ứng viên có thể bao gồm ứng viên theo cặp, và thông tin chỉ số trọng số có thể được trích xuất là thông tin chỉ số trọng số trên một trong số hai ứng viên khác đó trong danh sách ứng viên hợp nhất mà được dùng để trích xuất ứng viên theo cặp. Tức là, thông tin chỉ số trọng số trên ứng viên theo cặp có thể được trích xuất là thông tin chỉ số trọng số trên một trong số hai ứng viên khác đó trong danh sách ứng viên hợp nhất mà được dùng để trích xuất ứng viên theo cặp.

Ví dụ, các ứng viên có thể bao gồm ứng viên theo cặp, và ứng viên theo cặp có thể được trích xuất dựa trên hai ứng viên khác trong số các ứng viên. Khi thông tin chỉ số trọng số của mỗi trên hai ứng viên khác đó là giống như thông tin chỉ số trọng số thứ nhất, thì thông tin chỉ số trọng số dành cho ứng viên theo cặp có thể được trích xuất dựa trên thông tin chỉ số trọng số thứ nhất. Khi thông tin chỉ số trọng số trên mỗi trong số hai ứng viên khác đó là không giống nhau, thì thông tin chỉ số trọng số trên ứng viên theo cặp có thể được trích xuất dựa trên thông tin chỉ số trọng số mặc định, và trong

trường hợp này, thông tin chỉ số trọng số mặc định có thể tương ứng với thông tin chỉ số trọng số trên việc cho trọng số giống nhau cho mỗi trong số L0 mẫu dự đoán và L1 mẫu dự đoán.

Ví dụ, các ứng viên có thể bao gồm ứng viên theo cặp, và ứng viên theo cặp có thể được trích xuất dựa trên hai ứng viên khác trong số các ứng viên. Khi thông tin chỉ số trọng số của mỗi trên hai ứng viên khác đó là giống như thông tin chỉ số trọng số thứ nhất, thì thông tin chỉ số trọng số dành cho ứng viên theo cặp có thể được trích xuất dựa trên thông tin chỉ số trọng số thứ nhất. Khi thông tin chỉ số trọng số trên mỗi trong số hai ứng viên khác đó không giống nhau, thì thông tin chỉ số trọng số có thể được trích xuất dựa trên thông tin chỉ số trọng số thay vì thông tin chỉ số trọng số mặc định trong số các thông tin chỉ số trọng số của mỗi trong số hai ứng viên khác đó. Thông tin chỉ số trọng số mặc định có thể tương ứng với thông tin chỉ số trọng số trên việc cho trọng số giống nhau cho mỗi trong số L0 mẫu dự đoán và L1 mẫu dự đoán.

Ví dụ, danh sách ứng viên hợp nhất có thể là danh sách ứng viên hợp nhất khối con, và ứng viên hợp nhất afin, ứng viên hợp nhất khối con, hoặc ứng viên SbTMVP, là có thể được chọn. Ở đây, ứng viên hợp nhất afin theo các đơn vị là các khối con có thể được gọi là ứng viên hợp nhất khối con.

Ví dụ, các ứng viên bao gồm ứng viên hợp nhất afin, và ứng viên hợp nhất afin có thể bao gồm các vector chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV).

Ví dụ, khi ứng viên hợp nhất afin bao gồm CPMV dành cho điểm kiểm soát 0 (CP0) được đặt ở đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất afin có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của CP0. Khi ứng viên hợp nhất afin không bao gồm CPMV dành cho CP0 được đặt tại đỉnh trái của khối hiện tại, thì thông tin chỉ số trọng số trên ứng viên hợp nhất afin có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối cụ thể trong số các khối lân cận của điểm kiểm soát 1 (CP1) được đặt ở vị trí đỉnh phải

của khối hiện tại.

Khối cụ thể trong số các khối lân cận của CP0 là tương ứng với khối được sử dụng để trích xuất CPMV cho CP0, và các khối lân cận của CP0 có thể bao gồm khối lân cận ở góc đỉnh trái của khối hiện tại, khối lân cận bên trái kề với đáy của khối lân cận ở góc đỉnh trái, và khối lân cận trên đỉnh kề với bên phải của khối lân cận ở góc đỉnh trái.

Trong số các khối lân cận của CP1, thì khối cụ thể là tương ứng với khối được sử dụng để trích xuất CPMV dành cho CP1, và các khối lân cận CP1 có thể bao gồm khối lân cận ở góc đỉnh phải của khối hiện tại và khối lân cận trên đỉnh kề với bên trái của khối lân cận ở góc đỉnh phải.

Theo cách khác, ví dụ, các ứng viên có thể bao gồm ứng viên SbTMVP, và thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối lân cận bên trái của khối hiện tại. Tức là, thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối lân cận bên trái.

Theo cách khác, ví dụ, các ứng viên có thể bao gồm ứng viên SbTMVP, và thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được trích xuất là 0. Tức là, thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được trích xuất là 0. Ở đây, thông tin chỉ số trọng số là 0 có thể có nghĩa là việc các trọng số của mỗi chiều tham chiếu (tức là chiều dự đoán L0 và chiều dự đoán L1 trong quá trình lưỡng dự đoán) là giống nhau.

Theo cách khác, ví dụ, các ứng viên có thể bao gồm ứng viên SbTMVP, và thông tin chỉ số trọng số có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối trung tâm ở khối cùng vị trí. Tức là, thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được trích xuất dựa trên thông tin chỉ số trọng số trên khối trung tâm ở khối cùng vị trí. Ở đây, khối được đặt cùng vị trí có thể bao gồm khối tại cùng vị trí với khối hiện tại trong ảnh tham chiếu khác với ảnh hiện tại mà khối hiện tại được đặt

trong đó, và khối trung tâm có thể bao gồm khối con ở đáy phải trong số bốn khối con được đặt ở tâm của khối được đặt cùng vị trí.

Theo cách khác, ví dụ, các ứng viên có thể bao gồm ứng viên SbTMVP, và thông tin chỉ số trọng số có thể được trích xuất dựa trên thông tin chỉ số trọng số trên mỗi trong số các khối con ở khối cùng vị trí. Tức là, thông tin chỉ số trọng số trên ứng viên SbTMVP có thể được trích xuất dựa trên thông tin chỉ số trọng số trên mỗi trong số các khối con của khối cùng vị trí.

Mặc dù không được thể hiện trên Fig.12, nhưng, ví dụ, thiết bị giải mã này có thể trích xuất các mẫu dư dựa trên thông tin liên quan đến phần dư được bao gồm trong thông tin hình ảnh. Ngoài ra, thiết bị giải mã này có thể tạo ra các mẫu được tái tạo dựa trên các mẫu dự đoán và các mẫu dư. Khối được tái tạo và ảnh được tái tạo có thể được trích xuất dựa trên các mẫu được tái tạo.

Ví dụ, thiết bị giải mã này có thể thu thập thông tin hình ảnh bao gồm tất cả hoặc các bộ phận của các phần thông tin (hoặc các phần tử cú pháp) nêu trên bằng cách giải mã luồng bit hoặc thông tin được mã hoá. Ngoài ra, luồng bit hoặc thông tin được mã hoá này có thể được lưu giữ ở phương tiện lưu trữ đọc được bằng máy tính, và có thể khiến phương pháp giải mã nêu trên được thực hiện.

Tuy các phương pháp đã được mô tả trên cơ sở lưu đồ mà trong đó các bước hoặc các khối được liệt kê theo trình tự ở các phương án nêu trên, nhưng các bước của sáng chế là không chỉ giới hạn ở thứ tự nhất định, và bước nhất định có thể được thực hiện ở bước khác hoặc theo thứ tự khác hoặc đồng thời với các bước đã mô tả trên đây. Ngoài ra, người có hiểu biết trung bình trong lĩnh vực sẽ hiểu rằng các bước của các lưu đồ là không có tính loại trừ, và bước khác có thể được bao gồm vào trong đó, hoặc một hoặc nhiều bước ở lưu đồ đó có thể được xoá mà không ảnh hưởng đến phạm vi của sáng chế.

Phương pháp nêu trên theo sáng chế là có thể có dạng phần mềm, và thiết bị mã hoá và/hoặc thiết bị giải mã theo sáng chế có thể được bao gồm trong thiết bị để thực

hiện việc xử lý ảnh, ví dụ, ti vi (TV), máy tính, điện thoại thông minh, hộp thu tín hiệu truyền hình, thiết bị hiển thị, v.v..

Khi các phương án của sáng chế được thực hiện bằng phần mềm, thì phương pháp nêu trên có thể được thực hiện bởi môđun (tiền trình hoặc hàm) mà thực hiện chức năng nêu trên. Môđun này có thể được lưu giữ trong bộ nhớ và được thực thi bởi bộ xử lý. Bộ nhớ này có thể được lắp đặt bên trong hoặc bên ngoài bộ xử lý và có thể được nối đến bộ xử lý thông qua các phương tiện khác nhau đã được biết rõ. Bộ xử lý có thể bao gồm mạch tích hợp chuyên dụng (Application-Specific Integrated Circuit - ASIC), các con chip khác, mạch logic, và/hoặc thiết bị xử lý dữ liệu. Bộ nhớ này có thể bao gồm bộ nhớ chỉ đọc (Read-Only Memory - ROM), bộ nhớ truy cập ngẫu nhiên (Random Access Memory - RAM), bộ nhớ flash (bộ nhớ chớp nhoáng), thẻ nhớ, phương tiện lưu trữ, và/hoặc thiết bị lưu trữ khác. Nói cách khác, các phương án theo sáng chế có thể được thực hiện và được thực thi trên bộ xử lý, bộ vi xử lý, bộ điều khiển, hoặc chip. Ví dụ, các đơn vị chức năng được thể hiện trên các hình vẽ tương ứng có thể được thực hiện và được thực thi trên máy tính, bộ xử lý, bộ vi xử lý, bộ điều khiển, hoặc chip. Trong trường hợp này, thì thông tin về việc thực hiện (ví dụ, thông tin về các chỉ dẫn) hoặc các thuật toán là có thể được lưu giữ ở phương tiện lưu trữ kỹ thuật số.

Ngoài ra, thiết bị giải mã và thiết bị mã hoá mà (các) phương án của sáng chế được áp dụng vào đó là có thể được bao gồm trong bộ thu phát quảng bá đa phương tiện, thiết bị đầu cuối truyền thông di động, thiết bị video rạp chiếu phim gia đình, thiết bị video rạp chiếu phim kỹ thuật số, camera giám sát, thiết bị chat video, và thiết bị truyền thông thời gian thực chẳng hạn như truyền thông video, thiết bị tạo luồng di động, phương tiện lưu trữ, máy ghi hình cầm tay, nhà cung cấp dịch vụ video theo yêu cầu (Video on Demand - VoD), thiết bị video OTT (Over The Top - dịch vụ cung cấp nội dung trên nền mạng viễn thông), nhà cung cấp dịch vụ tạo luồng Internet, thiết bị video 3D (3 Dimensional - 3 chiều), thiết bị thực tế ảo (Virtual Reality - VR), thiết bị thực tế tăng cường (Augment Reality - AR), thiết bị video điện thoại hình ảnh, thiết bị đầu cuối

trên xe (ví dụ, thiết bị đầu cuối trên xe (bao gồm xe tự hành), thiết bị đầu cuối trên máy bay, hoặc thiết bị đầu cuối trên tàu thủy), và thiết bị video y tế; và có thể được sử dụng để xử lý tín hiệu ảnh hoặc dữ liệu ảnh. Ví dụ, thiết bị video OTT có thể bao gồm máy chơi game chuyên dụng, máy phát đĩa Bluray, TV có kết nối Internet, hệ thống rạp hát gia đình, điện thoại thông minh, máy tính bảng, và đầu ghi video kỹ thuật số (Digital Video Recorder - DVR).

Ngoài ra, phương pháp xử lý mà (các) phương án của sáng chế được áp dụng vào đó có thể được tạo ra dưới dạng chương trình được thực thi bởi máy tính và có thể được lưu giữ ở phương tiện ghi đọc được bằng máy tính. Dữ liệu đa phương tiện có cấu trúc dữ liệu theo (các) phương án của sáng chế cũng có thể được lưu giữ ở phương tiện ghi đọc được bằng máy tính. Phương tiện ghi đọc được bằng máy tính này bao gồm tất cả các loại thiết bị lưu trữ và các thiết bị lưu trữ phân tán mà dữ liệu đọc được bằng máy tính được lưu giữ trong đó. Phương tiện ghi đọc được bằng máy tính này có thể bao gồm, ví dụ, đĩa Bluray (Bluray Disc - BD), thiết bị buýt nối tiếp vạn năng (Universal Serial Bus - USB), ROM (Read Only Memory - bộ nhớ chỉ đọc), PROM (Programmable ROM - ROM lập trình được), EPROM (Erasable Programmable ROM - ROM lập trình được và xoá được), EEPROM (Electrically Erasable Programmable ROM - ROM lập trình được và xoá được bằng điện), RAM (Random Access Memory - bộ nhớ truy cập ngẫu nhiên), đĩa CD-ROM, băng từ, đĩa mềm, và thiết bị lưu trữ dữ liệu quang học. Phương tiện ghi đọc được bằng máy tính này cũng bao gồm các phương tiện được thực hiện dưới dạng sóng mang (ví dụ, truyền qua mạng Internet). Ngoài ra, luồng bit mà được tạo ra bởi phương pháp mã hoá này là có thể được lưu giữ trên phương tiện ghi đọc được bằng máy tính hoặc được truyền qua mạng truyền thông dùng dây hoặc không dây.

Ngoài ra, (các) phương án của sáng chế là có thể được thực hiện dưới dạng sản phẩm chương trình máy tính dựa trên mã chương trình, và mã chương trình này có thể được thực thi trên máy tính theo (các) phương án của sáng chế. Mã chương trình này

có thể được lưu giữ trên bộ mang đọc được bằng máy tính.

Fig.14 là hình vẽ thể hiện một ví dụ của hệ thống tạo luồng nội dung mà phương án này của sáng chế có thể được áp dụng vào đó.

Như được thể hiện trên Fig.14, hệ thống tạo luồng nội dung mà các phương án của sáng chế được áp dụng vào đó này có thể nói chung là bao gồm máy chủ mã hoá, máy chủ tạo luồng, máy chủ web, bộ lưu trữ phương tiện, thiết bị người dùng, và thiết bị đầu vào đa phương tiện.

Máy chủ mã hoá có chức năng nén các nội dung được nhập vào từ các thiết bị đầu vào đa phương tiện, chẳng hạn điện thoại thông minh, camera, máy ghi hình cầm tay, v.v., thành dữ liệu kỹ thuật số để tạo ra luồng bit, và truyền nó đến máy chủ tạo luồng. Theo ví dụ khác, trong trường hợp mà thiết bị đầu vào đa phương tiện, chẳng hạn điện thoại thông minh, camera, máy ghi hình cầm tay, v.v., trực tiếp tạo ra luồng bit, thì máy chủ mã hoá có thể được lược bỏ.

Luồng bit này có thể được tạo ra bằng phương pháp mã hoá hoặc phương pháp tạo luồng bit mà các phương án của sáng chế được áp dụng vào đó. Và máy chủ tạo luồng có thể tạm thời lưu giữ luồng bit này trong tiến trình truyền hoặc nhận luồng bit này.

Máy chủ tạo luồng truyền dữ liệu đa phương tiện đến thiết bị người dùng trên cơ sở yêu cầu của người dùng thông qua máy chủ web, mà có chức năng như trang thiết bị để báo cho người dùng biết có dịch vụ gì. Khi người dùng yêu cầu dịch vụ mà người dùng cần, thì máy chủ web truyền yêu cầu đó đến máy chủ tạo luồng, và máy chủ tạo luồng truyền dữ liệu đa phương tiện đến người dùng. Về mặt này, thì hệ thống tạo luồng nội dung có thể bao gồm máy chủ điều khiển riêng biệt, và trong trường hợp này, thì máy chủ điều khiển này có chức năng điều khiển các lệnh/các đáp ứng giữa trang thiết bị tương ứng trong hệ thống tạo luồng nội dung này.

Máy chủ tạo luồng có thể nhận các nội dung từ bộ lưu trữ phương tiện và/hoặc máy chủ mã hoá. Ví dụ, trong trường hợp các nội dung này được nhận từ máy chủ mã

hoá, thì các nội dung này có thể nhận được trong thời gian thực. Trong trường hợp này, máy chủ tạo luồng có thể lưu giữ luồng bit này trong một khoảng thời gian định trước để cung cấp dịch vụ tạo luồng một cách mượt mà.

Ví dụ, thiết bị người dùng có thể bao gồm điện thoại di động, điện thoại thông minh, laptop (máy tính xách tay), thiết bị đầu cuối phát sóng số, máy trợ lý cá nhân kỹ thuật số (Personal Digital Assistant - PDA), máy chơi đa phương tiện cơ động (Portable Multimedia Player - PMP), thiết bị dẫn đường, máy tính bảng lai laptop, máy tính bảng, máy tính xách tay siêu mỏng nhẹ, thiết bị đeo (ví dụ, thiết bị đầu cuối kiểu đồng hồ (đồng hồ thông minh), thiết bị đầu cuối kiểu kính mắt (kính thông minh), thiết bị hiển thị gắn trên đầu (Head Mounted Display - HMD)), ti vi kỹ thuật số, máy tính để bàn, biển chỉ dẫn kỹ thuật số, v.v..

Mỗi trong số các máy chủ trong hệ thống tạo luồng nội dung này có thể được vận hành dưới dạng máy chủ phân tán, và trong trường hợp này, thì dữ liệu mà mỗi máy chủ nhận được là có thể được xử lý theo cách phân tán.

Các điểm yêu cầu bảo hộ trong bản mô tả này có thể được kết hợp theo những cách khác nhau. Ví dụ, các dấu hiệu kỹ thuật ở các điểm yêu cầu bảo hộ về phương pháp của bản mô tả này có thể được kết hợp để được thực hiện ở thiết bị, và các dấu hiệu kỹ thuật ở các điểm yêu cầu bảo hộ về thiết bị có thể được kết hợp để được thực hiện ở phương pháp. Ngoài ra, các dấu hiệu kỹ thuật ở (các) điểm yêu cầu bảo hộ về phương pháp và (các) điểm yêu cầu bảo hộ về thiết bị có thể được kết hợp để được thực hiện ở thiết bị. Ngoài ra, các dấu hiệu kỹ thuật ở (các) điểm yêu cầu bảo hộ về phương pháp và (các) điểm yêu cầu bảo hộ về thiết bị có thể được kết hợp để được thực hiện ở phương pháp.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã ảnh được thực hiện bởi thiết bị giải mã, trong đó phương pháp giải mã ảnh này bao gồm các bước:

nhận thông tin hình ảnh bao gồm thông tin về chế độ liên dự đoán thông qua luồng bit;

tạo ra danh sách ứng viên hợp nhất của khối hiện tại dựa trên thông tin về chế độ liên dự đoán;

chọn một ứng viên từ trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất này;

trích xuất thông tin chuyển động trên khối hiện tại dựa trên ứng viên được chọn;

tạo ra L0 mẫu dự đoán và L1 mẫu dự đoán của khối hiện tại dựa trên thông tin chuyển động này; và

tạo ra các mẫu dự đoán của khối hiện tại dựa trên L0 mẫu dự đoán, L1 mẫu dự đoán này, và thông tin trọng số, trong đó thông tin trọng số là thông tin trọng số dành cho việc lưỡng dự đoán và được trích xuất dựa trên chỉ số trọng số dành cho ứng viên được chọn,

trong đó các ứng viên bao gồm ứng viên hợp nhất afin được xây dựng, và ứng viên hợp nhất afin được xây dựng này bao gồm các vector chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV),

trong đó dựa trên trường hợp mà ứng viên hợp nhất afin được xây dựng là được tạo ra dựa trên $\{CP0, CP1, CP3\}$, thì chỉ số trọng số dành cho ứng viên hợp nhất afin được xây dựng là được thiết đặt bằng chỉ số trọng số dành cho CP0, trong đó CP0 liên quan đến góc đỉnh trái của khối hiện tại, CP1 liên quan đến góc đỉnh phải của khối hiện tại và CP3 liên quan đến góc đáy phải của khối hiện tại.

2. Phương pháp mã hoá ảnh được thực hiện bởi thiết bị mã hoá, trong đó phương pháp mã hoá ảnh này bao gồm các bước:

xác định chế độ liên dự đoán của khối hiện tại và tạo ra thông tin về chế độ liên dự đoán chỉ thị chế độ liên dự đoán này;

tạo ra danh sách ứng viên hợp nhất của khối hiện tại dựa trên thông tin về chế độ liên dự đoán;

tạo ra thông tin chọn chỉ thị một trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất này; và

mã hoá thông tin hình ảnh bao gồm thông tin về chế độ liên dự đoán và thông tin chọn,

trong đó thông tin chuyển động của ứng viên được chỉ thị bởi thông tin chọn là để trích xuất L0 mẫu dự đoán và L1 mẫu dự đoán của khối hiện tại,

trong đó thông tin trọng số là để tạo ra các mẫu dự đoán của khối hiện tại dựa trên L0 mẫu dự đoán và L1 mẫu dự đoán, trong đó thông tin trọng số là dành cho việc lưỡng dự đoán và được trích xuất dựa trên chỉ số trọng số dành cho ứng viên được chỉ thị bởi thông tin chọn,

trong đó các ứng viên bao gồm ứng viên hợp nhất afin được xây dựng, và ứng viên hợp nhất afin được xây dựng này bao gồm các vector chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV),

trong đó, dựa trên trường hợp mà ứng viên hợp nhất afin được xây dựng là được tạo ra dựa trên $\{CP0, CP1, CP3\}$, thì chỉ số trọng số dành cho ứng viên hợp nhất afin được xây dựng là được thiết đặt bằng chỉ số trọng số dành cho CP0, và

trong đó CP0 liên quan đến góc đỉnh trái của khối hiện tại, CP1 liên quan đến góc đỉnh phải của khối hiện tại và CP3 liên quan đến góc đáy phải của khối hiện tại.

3. Phương tiện lưu trữ kỹ thuật số phi nhất thời đọc được bằng máy tính có lưu giữ luồng bit được tạo ra bởi phương pháp mã hoá ảnh, trong đó phương pháp này bao gồm các bước:

xác định chế độ liên dự đoán của khối hiện tại và tạo ra thông tin về chế độ liên dự đoán chỉ thị chế độ liên dự đoán này;

tạo ra danh sách ứng viên hợp nhất của khối hiện tại dựa trên thông tin về chế độ liên dự đoán;

tạo ra thông tin chọn chỉ thị một trong số các ứng viên được bao gồm trong danh sách ứng viên hợp nhất này; và

mã hoá thông tin hình ảnh để tạo ra luồng bit, trong đó thông tin hình ảnh này bao gồm thông tin về chế độ liên dự đoán và thông tin chọn,

trong đó thông tin chuyển động của ứng viên được chỉ thị bởi thông tin chọn là để trích xuất L0 mẫu dự đoán và L1 mẫu dự đoán của khối hiện tại,

trong đó thông tin trọng số là để tạo ra các mẫu dự đoán của khối hiện tại dựa trên L0 mẫu dự đoán và L1 mẫu dự đoán, trong đó thông tin trọng số là dành cho việc lưỡng dự đoán và được trích xuất dựa trên chỉ số trọng số dành cho ứng viên được chỉ thị bởi thông tin chọn,

trong đó các ứng viên bao gồm ứng viên hợp nhất afin được xây dựng, và ứng viên hợp nhất afin được xây dựng này bao gồm các vectơ chuyển động của điểm kiểm soát (Control Point Motion Vector - CPMV),

trong đó, dựa trên trường hợp mà ứng viên hợp nhất afin được xây dựng là được tạo ra dựa trên $\{CP0, CP1, CP3\}$, thì chỉ số trọng số dành cho ứng viên hợp nhất afin được xây dựng là được thiết đặt bằng chỉ số trọng số dành cho CP0, và

trong đó CP0 liên quan đến góc đỉnh trái của khối hiện tại, CP1 liên quan đến góc đỉnh phải của khối hiện tại và CP3 liên quan đến góc đáy phải của khối hiện tại.

Fig.1

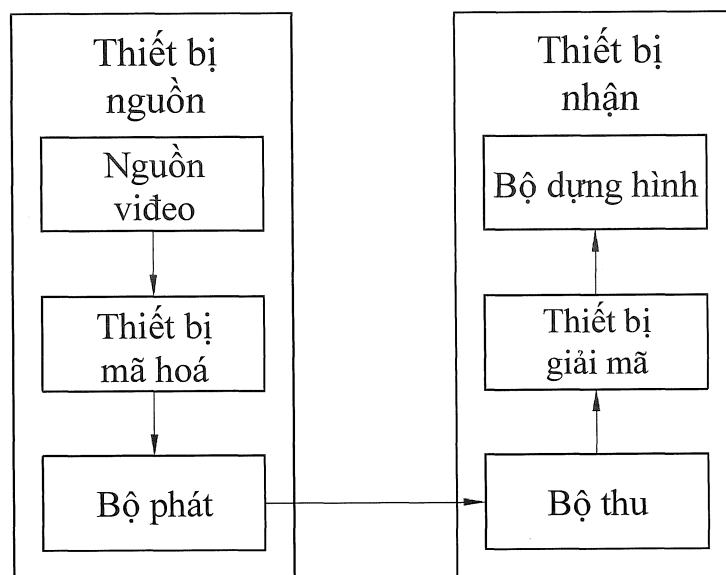


Fig.3

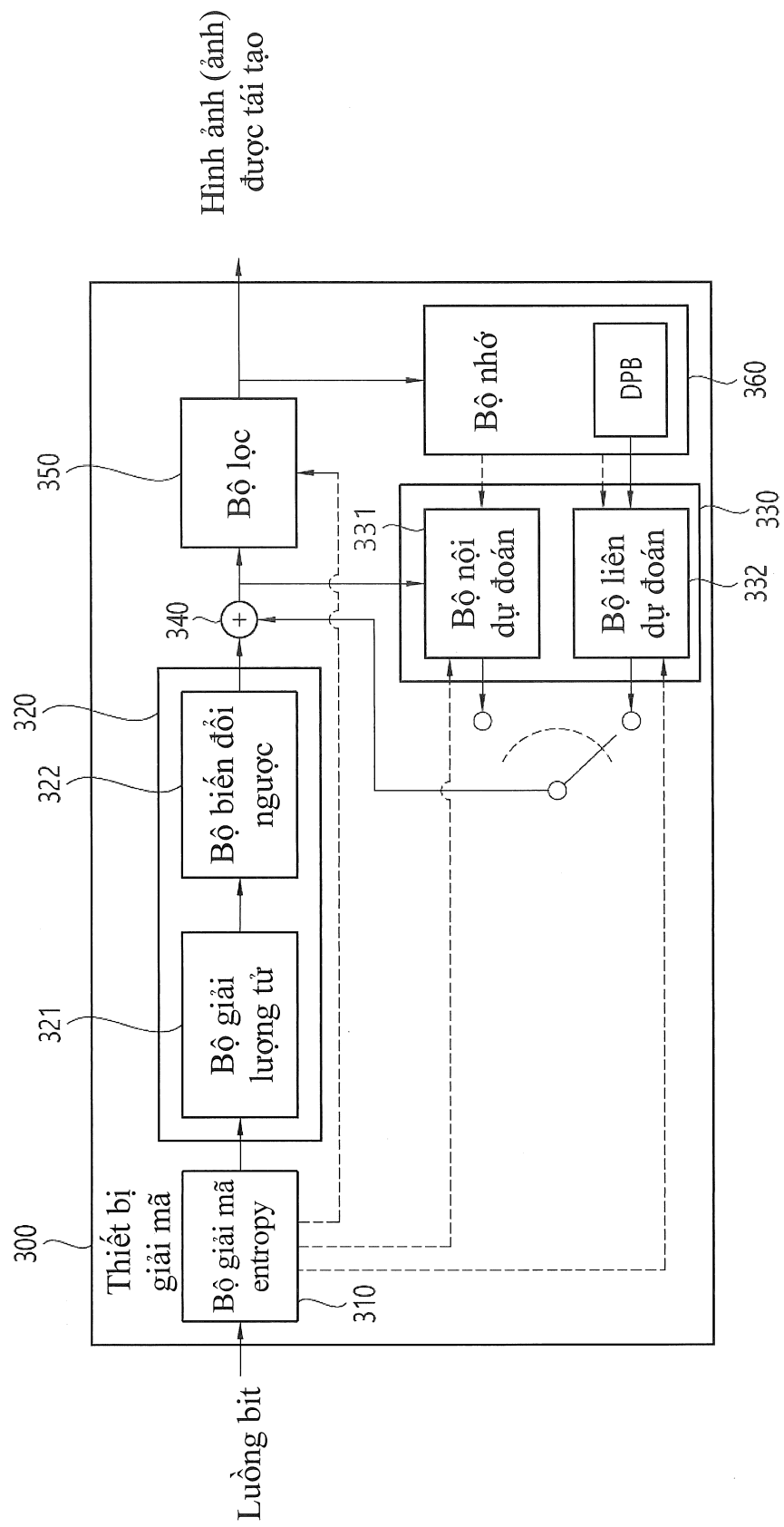


Fig.4

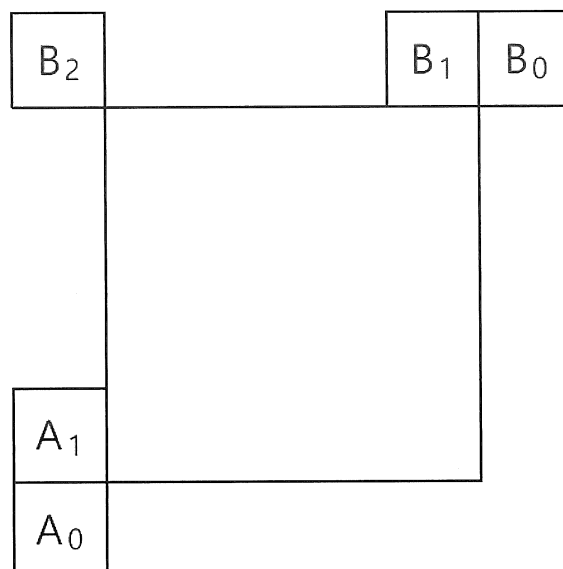


Fig.5A

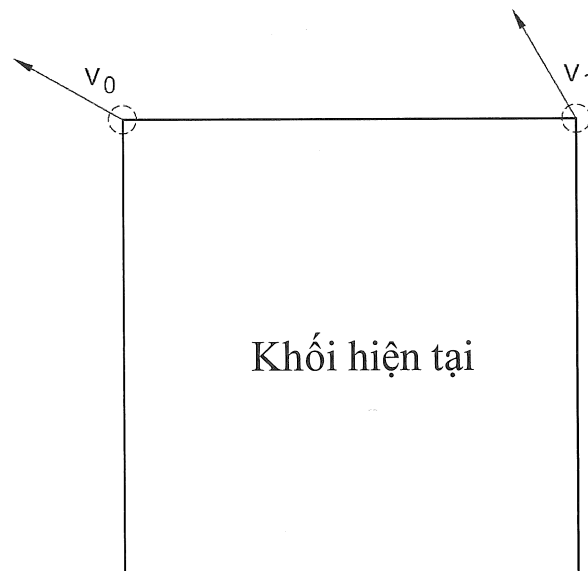


Fig.5B

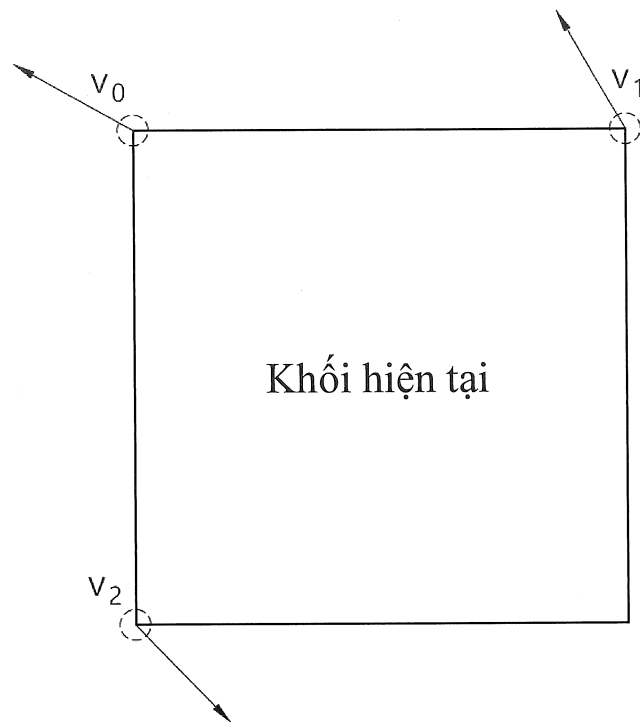


Fig.6

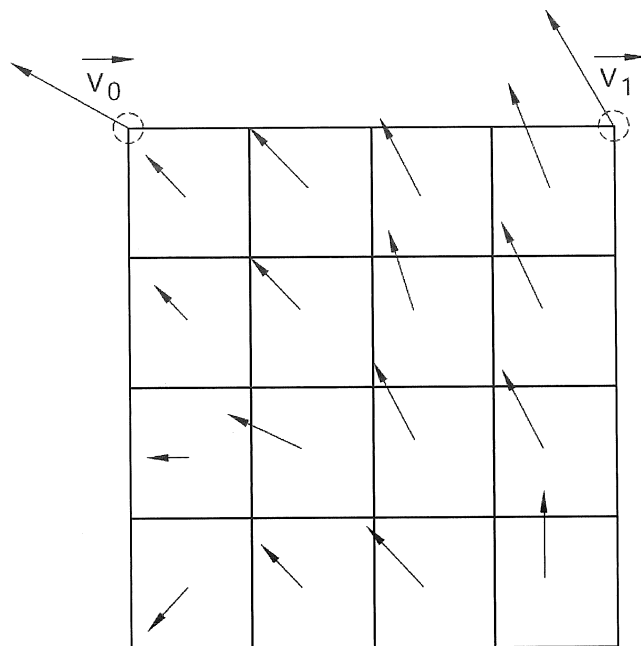


Fig.7

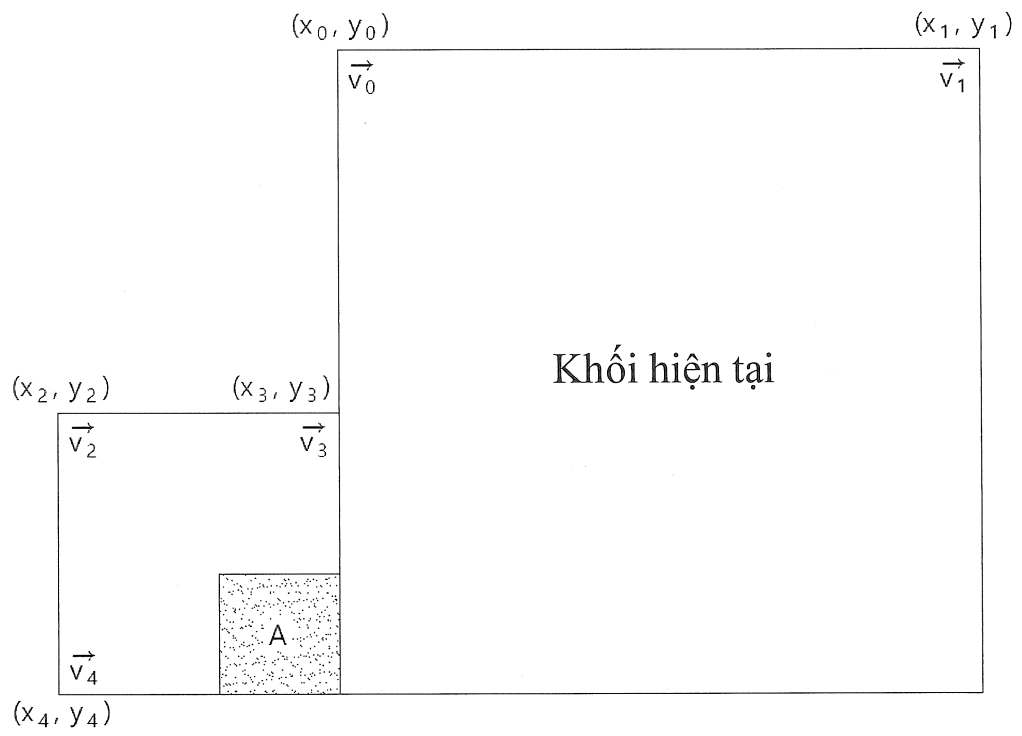


Fig.8

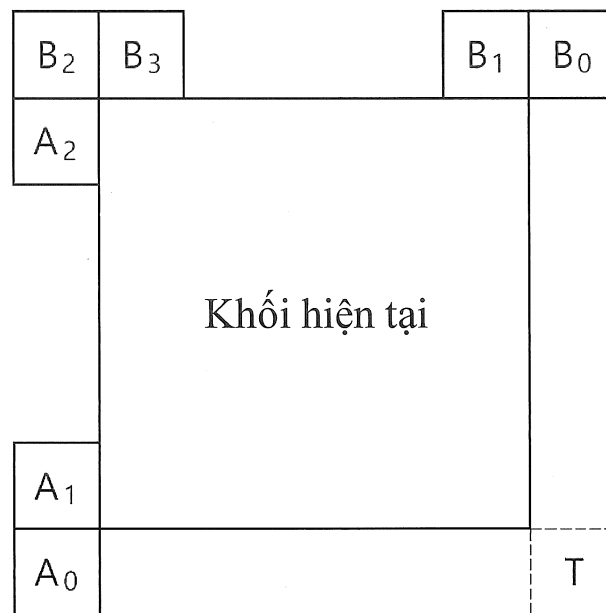


Fig.9

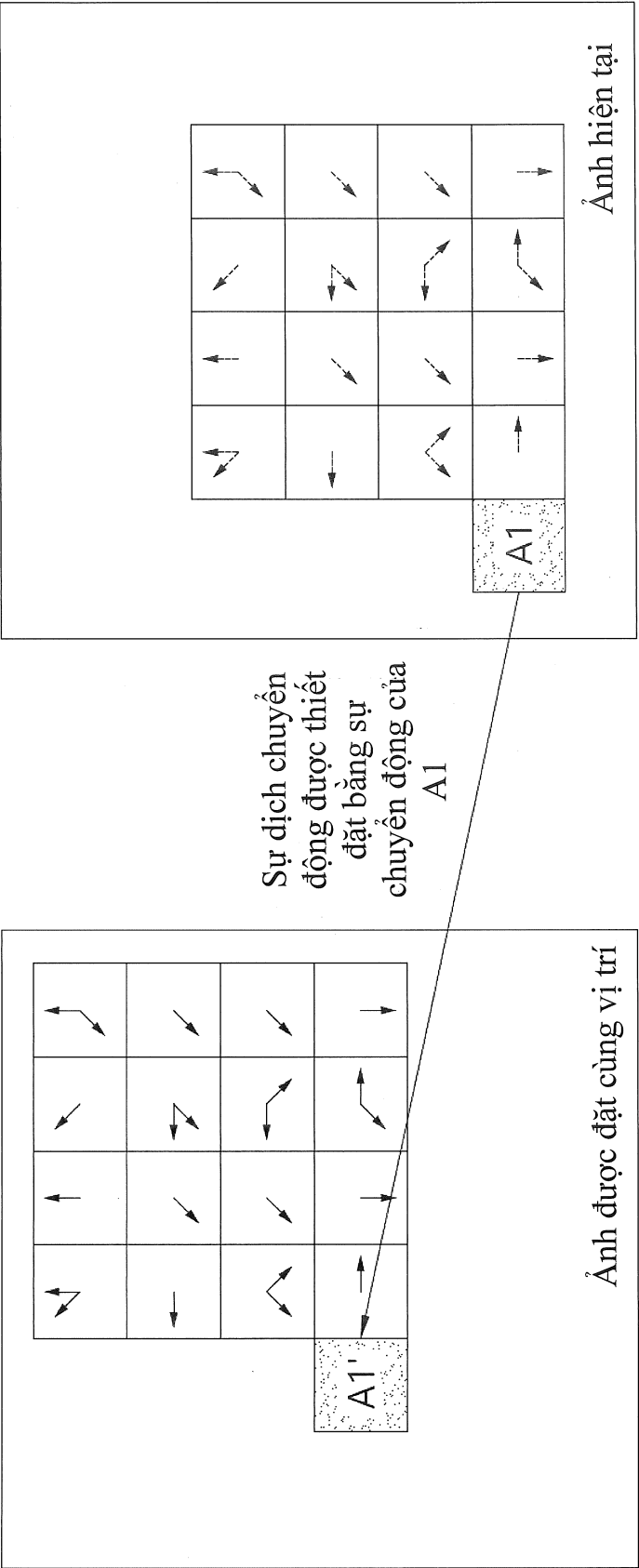


Fig.10

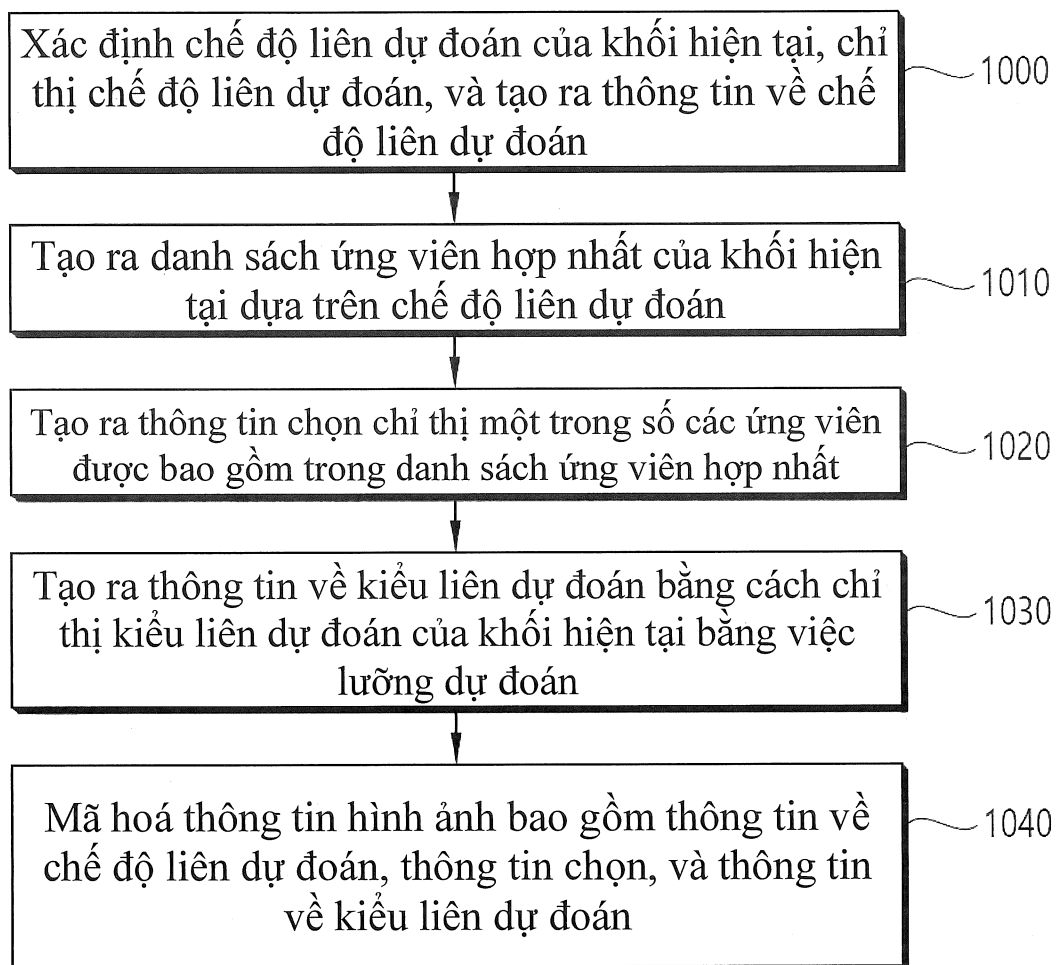
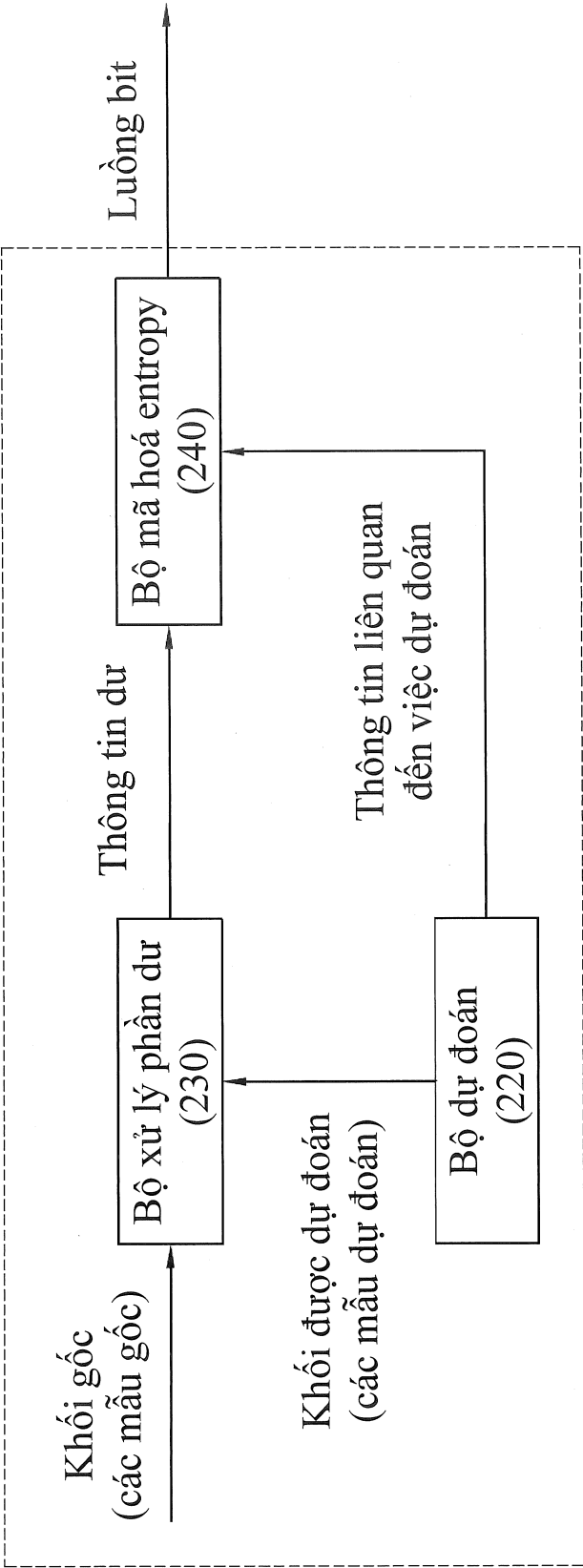


Fig.11



Thiết bị mã hoá (200)

Fig.12

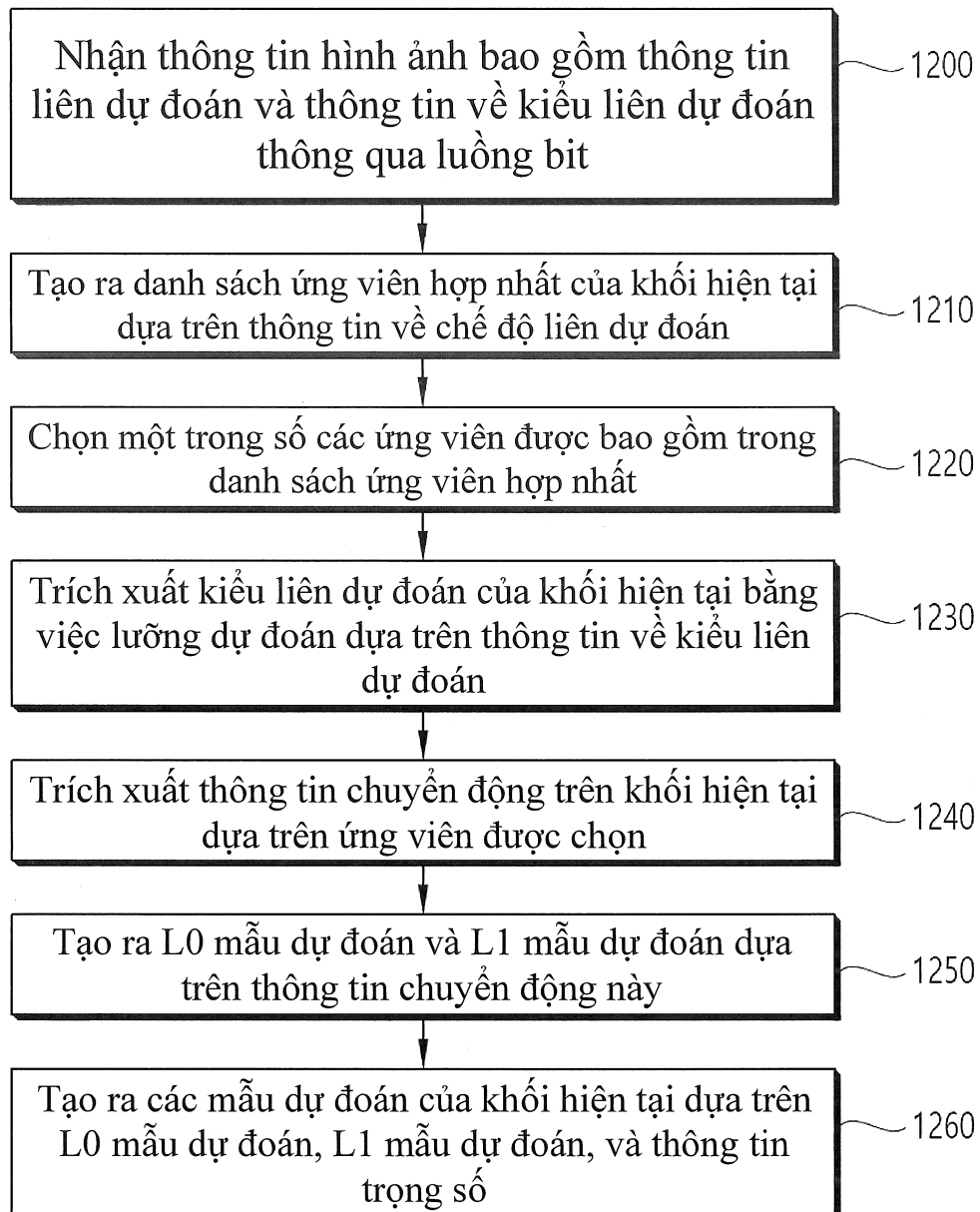
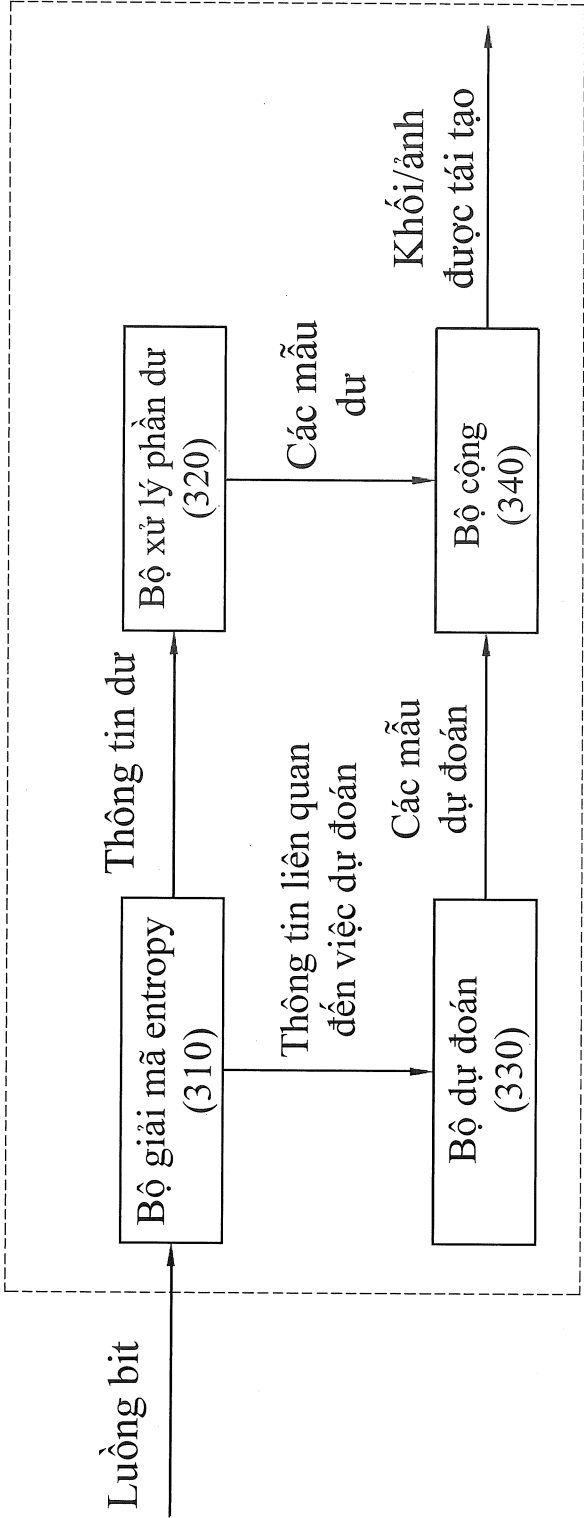


Fig.13



Thiết bị giải mã (300)

Fig.14

