



(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)



1-0043301

(51)^{2020.01} H04N 19/31; H04N 19/70

(13) B

(21) 1-2020-05521

(22) 11/03/2020

(86) PCT/US2020/022018 11/03/2020

(87) WO2020/185853 A2 17/09/2020

(30) 62/816,521 11/03/2019 US; 62/850,985 21/05/2019 US; 62/883,195 06/08/2019 US;
62/904,744 24/09/2019 US

(45) 25/02/2025 443

(43) 27/12/2021 405

(73) DOLBY LABORATORIES LICENSING CORPORATION (US)
1275 Market Street San Francisco, California 94103

(72) ATKINS, Robin (CA); YIN, Peng (US); LU, Taoran (CN); PU, Fangjun (CN);
MCCARTHY, Sean Thomas (US); HUSAK, Walter J. (US); CHEN, Tao (US); SU,
Guan-Ming (TW).

(74) Công ty TNHH Quốc tế D & N (D&N INTERNATIONAL CO.,LTD.)

(54) PHƯƠNG TIỆN BẮT BIẾN ĐỌC ĐƯỢC BẰNG BỘ XỬ LÝ ĐỂ MÃ HÓA VIDEO
CÓ THỂ ĐỊNH TỶ LỆ TỐC ĐỘ KHUNG

(21) 1-2020-05521

(57) Sáng chế đề cập đến các phương pháp và hệ thống cho khả năng mở rộng tốc độ khung. Sự hỗ trợ được cung cấp cho các chuỗi video đầu vào và đầu ra với tốc độ khung thay đổi và góc màn trập thay đổi trên các cảnh, hoặc cho các chuỗi video đầu vào với tốc độ khung đầu vào và góc màn trập đầu vào cố định, nhưng cho phép bộ giải mã tạo ra đầu ra video tại tốc độ khung và góc màn trập đầu ra khác với các giá trị đầu vào tương ứng. Các kỹ thuật cho phép bộ giải mã giải mã một cách có hiệu quả tính toán hơn tốc độ khung và góc màn trập đích có khả năng tương thích ngược riêng trong số các kỹ thuật được phép cũng được trình bày. Cụ thể, sáng chế đề cập đến phương tiện bất biến đọc được bằng bộ xử lý để mã hóa video có thể định tỷ lệ tốc độ khung.

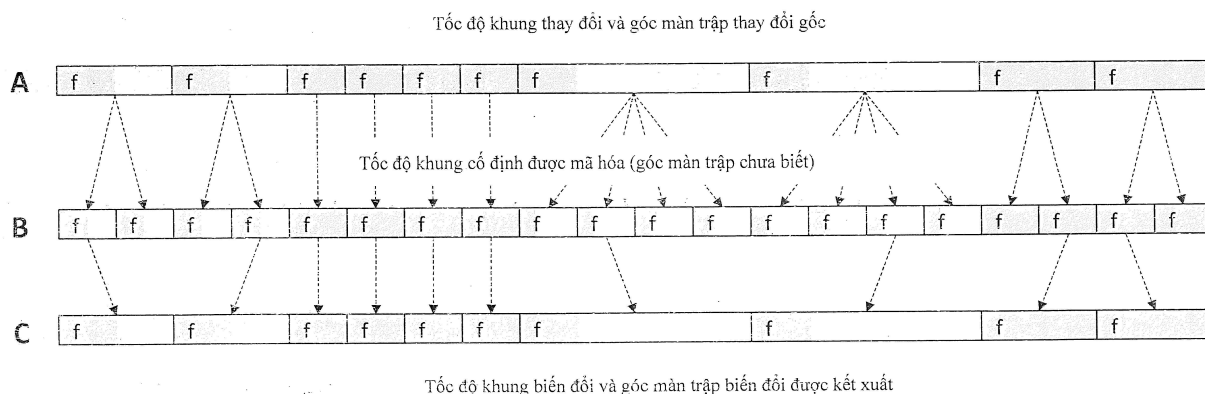


FIG. 3

Lĩnh vực kỹ thuật được đề cập

Nói chung, sáng chế đề cập đến các ảnh. Cụ thể hơn, phương án của sáng chế đề cập đến việc mã hóa video có thể định tỷ lệ tốc độ khung.

Tình trạng kỹ thuật của sáng chế

Như được dùng trong sáng chế này, thuật ngữ "dải động" (dynamic range - DR) có thể liên quan đến khả năng mà hệ thống thị giác của con người (human visual system - HVS) nhận thức được dải cường độ (ví dụ, độ chói, luma) trong ảnh, ví dụ, từ màu xám đậm nhất (vùng màu đen) đến màu trắng sáng nhất (vùng sáng nhất). Theo cách hiểu này, DR liên quan đến cường độ "dựa vào cảnh". DR cũng có thể liên quan đến khả năng mà thiết bị hiển thị có thể kết xuất một cách đầy đủ hoặc xấp xỉ dải cường độ của một độ rộng cụ thể. Theo cách hiểu này, DR liên quan đến cường độ "dựa vào hiển thị". Trừ khi một cách hiểu cụ thể được nêu rõ ràng để có ý nghĩa cụ thể tại đoạn nào đó trong phần mô tả này, nên hiểu rằng thuật ngữ này có thể được sử dụng theo một trong hai cách hiểu, ví dụ có thể hoán đổi cho nhau.

Như được dùng trong sáng chế này, thuật ngữ dải động cao (high dynamic range - HDR) liên quan đến độ rộng DR trải trên 14 - 15 bậc độ lớn của hệ thống thị giác của con người (HVS). Trong thực tế, DR mà con người có thể đồng thời cảm nhận được độ rộng lớn trong phạm vi cường độ có thể bị giảm bớt phần nào, so với HDR.

Trên thực tế, các ảnh bao gồm một hoặc nhiều thành phần màu (ví dụ, độ chói Y và độ màu Cb và Cr) trong đó mỗi thành phần màu được biểu diễn bởi độ chính xác n -bit trên mỗi điểm ảnh (ví dụ, $n = 8$). Bằng cách sử dụng kỹ thuật mã hóa độ chói tuyến tính, các ảnh trong đó $n \leq 8$ (ví dụ, các ảnh màu JPEG 24-bit) được xem là các ảnh có dải động chuẩn (standard dynamic range - SDR), trong khi các ảnh trong đó $n > 8$ có thể được xem là các ảnh có dải động tăng cường. Các ảnh HDR cũng có thể được lưu trữ và phân bố bằng cách sử dụng các định dạng dấu phẩy động có độ chính xác cao (ví dụ, 16-bit), như định dạng tập tin OpenEXR được phát triển bởi Industrial Light and Magic.

Hiện nay, việc phân bổ nội dung dải động cao của video, như Dolby Vision từ Dolby Laboratories hoặc HDR10 trong Blue-Ray, bị giới hạn ở độ phân giải 4K (ví dụ, 4096 x 2160 hoặc 3840 x 2160, và các độ phân giải tương tự) và 60 khung/giây (fps) bởi các tính năng của nhiều thiết bị phát lại. Trong các phiên bản tương lai, dự đoán rằng nội dung có độ phân giải lên đến 8K (ví dụ, 7680 x 4320) và 120 fps có thể có sẵn cho việc phân bổ và phát lại. Mong muốn rằng các loại nội dung trong tương lai sẽ tương thích với các thiết bị phát lại hiện có để đơn giản hóa hệ sinh thái nội dung phát lại HDR, như Dolby Vision. Lý tưởng nếu các nhà sản xuất nội dung có khả năng áp dụng và phân phối các công nghệ HDR tương lai mà không cần phải lấy và phân bổ các phiên bản nội dung đặc biệt mà tương thích với các thiết bị HDR hiện có (như HDR10 hoặc Dolby Vision). Theo đánh giá của các tác giả sáng chế ở đây, mong muốn có các công nghệ cải tiến để phân bổ theo cách có thể định tỷ lệ nội dung video, nhất là nội dung HDR.

Các phương pháp được mô tả trong phần này là các phương pháp có thể được theo đuổi, chứ không nhất thiết phải là các phương pháp đã được biết hoặc được theo đuổi trước đó. Do đó, trừ khi có quy định khác, không nên cho rằng phương pháp bất kỳ trong số các phương pháp được mô tả trong phần này đều thỏa mãn làm giải pháp kỹ thuật đã biết chỉ vì đưa chúng vào trong phần này. Tương tự, các vấn đề được xác định liên quan đến một hoặc nhiều phương pháp không nên được coi là đã được công nhận trong bất kỳ giải pháp kỹ thuật đã biết nào dựa vào phần này, trừ khi có quy định khác.

Bản chất kỹ thuật của sáng chế

Các phương án làm ví dụ được mô tả ở đây liên quan đến khả năng định tỷ lệ tốc độ khung trong mã hóa video. Theo một phương án, hệ thống có bộ xử lý thu dòng bit mã hóa bao gồm các khung video mã hóa, trong đó một hoặc nhiều khung mã hóa được mã hóa ở tốc độ khung thứ nhất và góc màn trập thứ nhất. Bộ xử lý thu cờ thứ nhất chỉ báo sự có mặt của nhóm các khung mã hóa cần giải mã ở tốc độ khung thứ hai và góc màn trập thứ hai, truy cập từ các giá trị dòng bit mã hóa của tốc độ khung thứ hai và góc màn trập thứ hai cho nhóm các khung mã hóa, và tạo ra các khung được giải mã ở tốc độ khung thứ hai và góc màn trập thứ hai dựa vào nhóm các khung mã hóa, tốc độ khung thứ nhất, góc màn trập thứ nhất, tốc độ khung thứ hai và góc màn trập thứ hai.

Theo phương án thứ hai, bộ giải mã có bộ xử lý:

thu dòng bit mã hóa bao gồm các nhóm khung video mã hóa, trong đó tất cả các khung video mã hóa trong dòng bit mã hóa được mã hóa ở tốc độ khung thứ nhất;

thu số lượng các khung kết hợp N ;

thu giá trị cho tốc độ khung đường cơ sở;

truy cập nhóm gồm N khung mã hóa liên tiếp, trong đó khung mã hóa thứ i trong nhóm gồm N khung mã hóa liên tiếp, trong đó $i = 1, 2, \dots, N$, biểu diễn số trung bình của tối đa i khung video đầu vào được mã hóa trong bộ mã hóa ở tốc độ khung đường cơ sở và góc màn trập thứ i dựa vào góc màn trập thứ nhất và tốc độ khung thứ nhất;

truy cập từ dòng bit mã hóa hoặc từ các giá trị đầu vào của người dùng cho tốc độ khung thứ hai và góc màn trập thứ hai, để giải mã nhóm gồm N khung mã hóa liên tiếp ở tốc độ khung thứ hai và góc màn trập thứ hai; và

tạo ra các khung được giải mã ở tốc độ khung thứ hai và góc màn trập thứ hai dựa vào nhóm gồm N khung mã hóa liên tiếp, tốc độ khung thứ nhất, góc màn trập thứ nhất, tốc độ khung thứ hai, và góc màn trập thứ hai.

Theo phương án thứ ba, cấu trúc dòng video mã hóa bao gồm:

phần hình ảnh mã hóa bao gồm việc mã hóa chuỗi hình ảnh video; và

phần báo hiệu bao gồm việc mã hóa:

tham số thang đo thời gian trong khoảng thời gian màn trập chỉ báo số lượng đơn vị thời gian trôi qua trong một giây;

tham số nhịp đồng hồ trong khoảng thời gian màn trập chỉ báo số lượng đơn vị thời gian của đồng hồ hoạt động ở tần số của tham số thang đo thời gian trong khoảng thời gian màn trập,

trong đó tham số nhịp đồng hồ trong khoảng thời gian màn trập chia cho tham số thang đo thời gian trong khoảng thời gian màn trập chỉ báo giá trị thời khoảng phơi sáng;

cờ thời khoảng trong khoảng thời gian màn trập chỉ báo liệu thông tin thời khoảng phơi sáng có cố định cho tất cả các lớp con thời gian trong phần hình ảnh mã hóa hay không; và

nếu cờ thời khoảng trong khoảng thời gian màn trập chỉ báo rằng thông tin thời khoảng phơi sáng là cố định, thì phiên bản giải mã của chuỗi hình ảnh video cho tất cả các lớp con thời gian trong phần hình ảnh mã hóa được giải mã bằng cách tính toán giá

trị thời khoảng phơi sáng dựa vào tham số thang đo thời gian trong khoảng thời gian màn trập và tham số nhịp đồng hồ trong khoảng thời gian màn trập, nếu không thì

phần báo hiệu bao gồm một hoặc nhiều mảng tham số lớp con, trong đó các giá trị trong một hoặc nhiều mảng tham số lớp con kết hợp với tham số thang đo thời gian trong khoảng thời gian màn trập được sử dụng để tính toán cho mỗi lớp con một giá trị thời khoảng phơi sáng lớp con tương ứng để hiển thị phiên bản giải mã của lớp con thời gian của chuỗi hình ảnh video.

Mô tả vắn tắt các hình vẽ

Phương án của sáng chế được minh họa bằng ví dụ, và không bị giới hạn ở ví dụ, trên các hình vẽ kèm theo và trong đó các số tham chiếu giống nhau chỉ các thành phần giống nhau và trong đó:

Fig. 1 mô tả quy trình làm ví dụ cho đường ống phân phối video;

Fig. 2 mô tả quy trình làm ví dụ để kết hợp các khung góc liên tiếp để kết xuất tốc độ khung đích ở góc màn trập đích theo một phương án của sáng chế;

Fig. 3 mô tả dạng biểu diễn làm ví dụ của chuỗi đầu vào với tốc độ khung đầu vào biến đổi và góc màn trập biến đổi trong vùng chứa với tốc độ khung cố định theo một phương án của sáng chế; và

Fig. 4 mô tả dạng biểu diễn làm ví dụ về khả năng định tỷ lệ theo thời gian ở các tốc độ khung và góc màn trập khác nhau với khả năng tương thích ngược theo một phương án của sáng chế.

Mô tả chi tiết sáng chế

Sáng chế đề cập đến các phương án làm ví dụ liên quan đến khả năng định tỷ lệ tốc độ khung cho mã hóa video. Trong phần mô tả sau đây, nhằm mục đích giải thích, nhiều chi tiết cụ thể được trình bày nhằm cung cấp sự hiểu biết cặn kẽ về các phương án khác nhau của sáng chế. Tuy nhiên, rõ ràng là các phương án khác nhau của sáng chế có thể được thực hiện mà không có các chi tiết cụ thể này. Trong các ví dụ khác, các cấu trúc và thiết bị đã biết không được mô tả quá chi tiết, để tránh làm cho các phương án của sáng chế rối, tối nghĩa và khó hiểu một cách không cần thiết.

Đường ống xử lý phân phối video ví dụ

Fig. 1 mô tả quy trình làm ví dụ của đường ống phân phối video thông thường (100) thể hiện các giai đoạn khác nhau từ giai đoạn ghi lại video đến giai đoạn hiển thị nội dung video. Chuỗi các khung video (102) được ghi lại hoặc được tạo ra bằng cách sử dụng khối tạo ảnh (105). Các khung video (102) có thể được ghi lại bằng kỹ thuật số (ví dụ, bằng máy ảnh kỹ thuật số) hoặc được tạo bằng máy tính (ví dụ, sử dụng hoạt hình trên máy tính) để cung cấp dữ liệu video (107). Ngoài ra, các khung video (102) có thể được ghi lại trên phim bằng máy quay phim. Phim được chuyển đổi sang định dạng kỹ thuật số để cung cấp dữ liệu video (107). Trong giai đoạn sản xuất (110), dữ liệu video (107) được chỉnh sửa để cung cấp dòng sản xuất video (112).

Dữ liệu video của dòng sản xuất video (112) sau đó được cung cấp cho bộ xử lý ở khối (115) để chỉnh sửa hậu kỳ. Việc chỉnh sửa hậu kỳ ở khối (115) có thể bao gồm điều chỉnh hoặc sửa đổi màu sắc hoặc độ sáng ở các khu vực cụ thể của ảnh để nâng cao chất lượng ảnh hoặc đạt được sự thể hiện cụ thể cho ảnh theo mục đích sáng tạo của người tạo video. Việc này đôi khi được gọi là “căn chỉnh màu sắc” hoặc “chỉnh màu sắc”. Việc chỉnh sửa khác (ví dụ, chọn và tạo chuỗi cảnh, cắt ảnh, thêm hiệu ứng đặc biệt trực quan do máy tính tạo ra, điều khiển rung lắc hoặc làm mờ, điều khiển tốc độ khung, v.v.) có thể được thực hiện ở khối (115) để tạo ra phiên bản cuối cùng (117) của sản phẩm sản xuất để phân phối. Trong quá trình chỉnh sửa hậu kỳ (115), các ảnh video được xem trên màn hình chuẩn (125). Sau khi chỉnh sửa hậu kỳ (115), dữ liệu video của quy trình sản xuất cuối cùng (117) có thể được phân phối đến khối mã hóa (120) để phân phối xuôi dòng đến các thiết bị giải mã và phát lại như máy truyền hình, thiết bị giải mã tín hiệu truyền hình, rạp chiếu phim, và tương tự. Theo một số phương án, khối mã hóa (120) có thể bao gồm các bộ mã hóa âm thanh và video, chẳng hạn như các bộ mã hóa được xác định bởi định dạng ATSC, DVB, DVD, Blu-Ray và các định dạng phân phối khác, để tạo dòng bit mã hóa (122). Trong bộ thu, dòng bit mã hóa (122) được giải mã bằng đơn vị giải mã (130) để tạo ra tín hiệu giải mã (132) biểu diễn tín hiệu giống hoặc gần giống với tín hiệu (117). Bộ thu có thể được gắn với màn hình đích (140) mà có thể có các đặc điểm hoàn toàn khác so với màn hình chuẩn (125). Trong trường hợp đó, khối quản lý màn hình (135) có thể được dùng để ánh xạ dải động của tín hiệu giải mã (132) đến các đặc tính của màn hình đích (140) bằng cách tạo ra tín hiệu ánh xạ của màn hình (137).

MÃ HÓA CÓ THỂ ĐỊNH TỶ LỆ

Mã hóa có thể định tỷ lệ đã là một phần của một số chuẩn mã hóa video như, MPEG-2, AVC, và HEVC. Theo một số phương án của sáng chế, mã hóa có thể định tỷ lệ được mở rộng để cải thiện hiệu suất và độ linh hoạt, đặc biệt là khi nó liên quan đến nội dung HDR có độ phân giải rất cao.

Như được sử dụng ở đây, thuật ngữ “góc màn trập” thể hiện cài đặt màn trập có thể điều chỉnh được để điều khiển các tỷ lệ thời gian phim tiếp xúc với ánh sáng trong mỗi khoảng thời gian khung. Ví dụ, theo phương án

$$\frac{\text{Góc màn trập}}{360} = \frac{\text{thời gian phơi sáng}}{\text{khoảng thời gian khung}} \quad (1)$$

Thuật ngữ này đến từ các màn trập kế thừa, cơ học, xoay; tuy nhiên, các máy ảnh kỹ thuật số hiện đại cũng có thể điều chỉnh màn trập của chúng theo kiểu điện tử. Các nhà nhiếp ảnh có thể sử dụng góc màn trập để điều khiển lượng nhòe hoặc rung do chuyển động mà được ghi trong mỗi khung. Lưu ý là thay vì sử dụng “thời gian phơi sáng” thì có thể còn sử dụng các thuật ngữ thay thế như “thời khoảng phơi sáng,” “khoảng thời gian màn trập,” và “tốc độ màn trập.” Tương tự, thay vì sử dụng “khoảng thời gian khung” thì có thể sử dụng thuật ngữ “thời khoảng khung.” Ngoài ra, có thể thay thế “khoảng thời gian khung” bằng “1/tốc độ khung.” Giá trị thời gian phơi sáng thường nhỏ hơn hoặc bằng thời khoảng của khung. Ví dụ, góc màn trập 180 độ chỉ báo thời gian phơi sáng bằng một nửa thời khoảng của khung. Trong một số trường hợp, thời gian phơi sáng có thể lớn hơn thời khoảng của khung của video mã hóa, ví dụ, khi tốc độ khung mã hóa là 120 fps và tốc độ khung của nội dung video liên kết trước khi mã hóa và hiển thị là 60 fps.

Xem xét, mà không giới hạn ở, phương án mà nội dung gốc được chụp (hoặc tạo ra) ở tốc độ khung gốc (ví dụ, 120 fps) với góc màn trập 360 độ. Tiếp theo, trong thiết bị thu, có thể kết xuất đầu ra video ở nhiều tốc độ khung bằng hoặc thấp hơn tốc độ khung gốc bằng sự kết hợp công bằng các khung gốc, ví dụ, bằng cách tính trung bình hoặc các cách khác được biết đến trong lĩnh vực kỹ thuật.

Quy trình kết hợp có thể được thực hiện với các tín hiệu được mã hóa phi tuyến tính, (ví dụ, sử dụng gama, PQ hoặc HLG), nhưng chất lượng ảnh tốt nhất thu được thông qua kết hợp các khung trong miền ánh sáng tuyến tính bằng cách trước tiên chuyển đổi các tín hiệu mã hóa phi tuyến tính thành các dạng biểu diễn ánh sáng tuyến tính, tiếp theo kết hợp các khung được chuyển đổi, và cuối cùng mã hóa lại đầu ra bằng chức năng truyền phi tuyến

tính. Quy trình này cung cấp sự mô phỏng chính xác hơn về sự phơi sáng của máy ảnh vật lý so với sự kết hợp trong miền phi tuyến tính.

Nói chung, quy trình kết hợp khung có thể được biểu diễn xét về tốc độ khung gốc, tốc độ khung đích, góc màn trập đích, và số lượng khung cần kết hợp là:

$$n_frames = (target_shutter_angle/360)*(original_frame_rate/target_frame_rate), \quad (2)$$

tương đương với

$$target_shutter_angle = 360*n_frames*(target_frame_rate/original_frame_rate), \quad (3)$$

trong đó n_frames là số lượng các khung kết hợp, $original_frame_rate$ là tốc độ khung của nội dung gốc, $target_frame_rate$ là tốc độ khung cần kết xuất (trong đó, $target_frame_rate \leq original_frame_rate$), và $target_shutter_angle$ chỉ báo lượng nhòe chuyển động mong muốn. Theo ví dụ này, giá trị cực đại của $target_shutter_angle$ là 360 độ và tương ứng với độ nhòe chuyển động cực đại. Giá trị cực tiểu của $target_shutter_angle$ có thể được biểu diễn là $360*(target_frame_rate/original_frame_rate)$ và tương ứng với độ nhòe chuyển động cực tiểu. Giá trị cực đại của n_frames có thể được biểu diễn là $(original_frame_rate/target_frame_rate)$. Các giá trị của $target_frame_rate$ và $target_shutter_angle$ nên được chọn sao cho giá trị của n_frames là số nguyên khác không.

Trong trường hợp đặc biệt mà tốc độ khung gốc là 120 fps, phương trình (2) có thể được viết lại là

$$n_frames = target_shutter_angle/(3*target_frame_rate),$$

(4)

tương đương với

$$target_shutter_angle = 3*n_frames*target_frame_rate. \quad (5)$$

Mối quan hệ giữa các giá trị của $target_frame_rate$, n_frames , và $target_shutter_angle$ được thể hiện trong Bảng 1 cho trường hợp $original_frame_rate = 120$ fps. Trong Bảng 1, “NA” chỉ báo rằng sự kết hợp tương ứng của tốc độ khung đích và số lượng các khung kết hợp là không được phép.

Bảng 1: Mối quan hệ giữa tốc độ khung đích, số lượng khung được kết hợp, và góc màn trập đích, với tốc độ khung gốc bằng 120 fps.

Tốc độ khung đích (fps)	Số lượng các khung được kết hợp				
	5	4	3	2	1
	Góc màn trập đích (độ)				
24	360	288	216	144	72
30	NA	360	270	180	90
40	NA	NA	360	240	120
60	NA	NA	NA	360	180

Fig. 2 mô tả quy trình làm ví dụ để kết hợp các khung gốc liên tiếp để kết xuất tốc độ khung đích ở góc màn trập đích theo một phương án. Cho trước chuỗi đầu vào (205) ở tốc độ 120 fps và góc màn trập 360 độ, quy trình tạo ra chuỗi video đầu ra (210) ở tốc độ 24 fps và góc màn trập 216 độ bằng cách kết hợp ba trong số các khung đầu vào trong tập hợp năm khung liên tiếp (ví dụ, ba khung liên tiếp thứ nhất), và bỏ hai khung còn lại. Lưu ý rằng theo một số phương án, khung đầu ra-01 của (210) có thể được tạo ra bằng cách kết hợp các khung đầu vào thay thế (205), như các khung 1, 3, và 5, hoặc các khung 2, 4, và 5, và các khung tương tự; tuy nhiên, mong đợi rằng việc kết hợp các khung liên tiếp sẽ tạo ra đầu ra video có chất lượng tốt hơn.

Mong muốn hỗ trợ nội dung gốc có tốc độ khung biến đổi, ví dụ, để quản lý hiệu ứng nghệ thuật và phong cách. Cũng mong muốn tốc độ khung đầu vào biến đổi của nội dung gốc được đóng gói trong “vùng chứa” mà có tốc độ khung cố định để làm đơn giản việc sản xuất, trao đổi, phân phối nội dung. Ví dụ, ba phương án về cách biểu diễn dữ liệu video có tốc độ khung biến đổi trong vùng chứa tốc độ khung cố định được trình bày. Nhằm mục đích rõ ràng và không giới hạn, các phần mô tả sau đây sử dụng vùng chứa 120 fps cố định, nhưng các phương pháp tiếp cận có thể được mở rộng dễ dàng đến vùng chứa có tốc độ khung khác.

Phương án thứ nhất (tốc độ khung biến đổi)

Phương án thứ nhất là phần mô tả rõ ràng về nội dung gốc có tốc độ khung biến đổi (không bất biến) được đóng gói trong vùng chứa có tốc độ khung bất biến. Ví dụ, nội dung gốc có tốc độ các khung khác nhau, giả sử, ở tốc độ 24, 30, 40, 60, hoặc 120 fps, cho các cảnh khác nhau, có thể được đóng gói trong vùng chứa có tốc độ khung bất biến 120 fps. Với ví

dụ này, mỗi khung đầu vào có thể được trùng lặp 5x, 4x, 3x, 2x, hoặc 1x lần để đóng gói nó vào vùng chứa 120 fps thông thường.

Fig. 3 mô tả ví dụ về chuỗi video đầu vào A có tốc độ khung biến đổi và góc màn trập biến đổi được biểu diễn trong dòng bit mã hóa B có tốc độ khung cố định. Tiếp theo, trong bộ giải mã, bộ giải mã tái tạo chuỗi video đầu ra C ở tốc độ khung và góc màn trập mong muốn, có thể thay đổi từ cảnh này sang cảnh khác. Ví dụ, như được minh họa trên Fig. 3, để tạo ra chuỗi B, một số khung đầu vào được trùng lặp, một số được mã hóa như nguyên trạng (không có sự trùng lặp), và một số được sao chép bốn lần. Tiếp theo, để tạo ra chuỗi C, một khung bất kỳ từ mỗi tập hợp khung trùng lặp được chọn để tạo ra các khung đầu ra, so khớp tốc độ khung gốc và góc màn trập.

Theo phương án này, siêu dữ liệu được chèn trong dòng bit để chỉ báo tốc độ khung và góc màn trập (cơ sở) gốc. Siêu dữ liệu có thể được báo hiệu bằng cách sử dụng cú pháp mức cao như Tập hợp tham số chuỗi (SPS), Tập hợp tham số hình ảnh (PPS), phần đầu nhóm lát hoặc ô, và tương tự. Sự có mặt của siêu dữ liệu cho phép các bộ mã hóa và bộ giải mã thực hiện các chức năng có lợi, như:

- a) Bộ mã hóa có thể bỏ qua các khung trùng lặp, nhờ đó làm tăng tốc độ mã hóa và đơn giản hóa việc xử lý. Ví dụ, tất cả các đơn vị cây mã hóa (coding tree units - CTUs) trong các khung trùng lặp có thể được mã hóa bằng cách sử dụng chế độ BỎ QUA (SKIP) và chỉ số tham chiếu 0 trong DANH SÁCH (LIST) 0 của các khung tham chiếu, chỉ ra khung hình được giải mã mà các khung trùng lặp được sao chép từ đó.
- b) Bộ giải mã có thể bỏ qua việc giải mã các khung trùng lặp nhờ đó đơn giản hóa việc xử lý. Ví dụ, siêu dữ liệu trong dòng bit có thể chỉ báo khung là bản sao của khung giải mã trước đó mà bộ giải mã có thể tái tạo bằng cách sao chép và mà không giải mã khung mới.
- c) Thiết bị phát lại có thể tối ưu việc xử lý xuôi dòng bằng cách chỉ báo tốc độ khung cơ sở, ví dụ bằng cách điều chỉnh sự chuyển đổi tốc độ khung hoặc các thuật toán làm giảm nhiễu.

Phương án này cho phép người dùng cuối thấy nội dung kết xuất ở các tốc độ khung được dự định bởi những người sáng tạo nội dung. Phương án này không cung cấp khả năng

trương thích ngược với các thiết bị mà không hỗ trợ tốc độ khung của vùng chứa là, ví dụ, 120 fps.

Các bảng 2 và 3 thể hiện cú pháp làm ví dụ của tải tin chuỗi byte thô (RBSB) cho tập hợp tham số chuỗi và phần đầu nhóm ô, trong đó các phần tử cú pháp mới được đề xuất được thể hiện bằng phông chữ được làm nổi bật. Cú pháp còn lại theo sau cú pháp trong bản đặc tả được đề xuất của bộ mã hóa-giải mã đa năng video (VVC) (Ref.[2]).

Ví dụ, trong SPS (xem Bảng 2), có thể bổ sung cờ để cho phép tốc độ khung biến đổi.

sps_vfr_enabled_flag bằng 1 chỉ ra chuỗi video mã hóa (CVS) có thể chứa tốc độ khung biến đổi content. **sps_vfr_enabled_flag** bằng 0 chỉ ra CVS chứa nội dung tốc độ khung cố định.

Trong `tile_group_header()` (xem Bảng 3),

tile_group_vrf_info_present_flag bằng 1 chỉ ra các phần tử cú pháp `tile_group_true_fr` và `tile_group_shutterangle` có mặt trong cú pháp. **tile_group_vrf_info_present_flag** bằng 0 chỉ ra các phần tử cú pháp `tile_group_true_fr` và `tile_group_shutterangle` không có mặt trong cú pháp. Khi **tile_group_vrf_info_present_flag** không có mặt, thì nó được suy ra là 0.

tile_group_true_fr chỉ báo tốc độ khung đúng của dữ liệu video được mang trong dòng bit này.

tile_group_shutterangle chỉ báo góc màn trập tương ứng với tốc độ khung đúng của dữ liệu video được mang trong dòng bit này.

tile_group_skip_flag bằng 1 chỉ ra nhóm ô hiện thời được sao chép từ một nhóm ô khác. **tile_group_skip_flag** bằng 0 chỉ ra nhóm ô hiện thời không được sao chép từ một nhóm ô khác.

tile_group_copy_pic_order_cnt_lsb chỉ ra modul đếm thứ tự hình ảnh `MaxPicOrderCntLsb` cho hình ảnh giải mã trước đó mà từ đó hình ảnh hiện thời sao chép khi **tile_group_skip_flag** được đặt bằng 1.

Bảng 2: Ví dụ về cú pháp RBSP của tập hợp tham số cho nội dung với tốc độ khung biến đổi

<code>seq_parameter_set_rbsp() {</code>	Mô tả
<code>sps_max_sub_layers_minus1</code>	u(3)
<code>sps_reserved_zero_5bits</code>	u(5)

profile_tier_level(sps_max_sub_layers_minus1)	
sps_seq_parameter_set_id	ue(v)
...	
sps_vfr_enabled_flag	u(1)
...	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbp_trailing_bits()	
}	

Bảng 3: Ví dụ về cú pháp phần đầu nhóm ô với sự hỗ trợ cho nội dung có tốc độ khung biến đổi

tile_group_header() {	Mô tả
tile_group_pic_parameter_set_id	ue(v)
if(NumTilesInPic > 1) {	
tile_group_address	ue(v)
num_tiles_in_tile_group_minus1	ue(v)
}	
tile_group_type	ue(v)
tile_group_pic_order_cnt_lsb	ue(v)
if(sps_vfr_enabled_flag) {	
tile_group_vfr_info_present_flag	u(1)
if(tile_group_vfr_info_present_flag) {	
tile_group_true_fr	u(9)
tile_group_shutterangle	u(9)
}	
tile_group_skip_flag	u(1)

}	
if(tile_group_skip_flag)	
tile_group_copy_pic_order_cnt_lsb	ue(v)
else{	
TẤT CẢ TILE_GROUP_SYNTAX KHÁC	
}	
if(num_tiles_in_tile_group_minus1 > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_tiles_in_tile_group_minus1; i++)	
entry_point_offset_minus1[i]	ue(v)
}	
byte_alignment()	
}	

Phương án thứ hai – Vùng chứa tốc độ khung cố định

Phương án thứ hai cho phép các trường hợp sử dụng mà nội dung gốc có tốc độ khung cố định và góc màn trập có thể được kết xuất bởi bộ giải mã ở tốc độ khung khác và góc màn trập mô phỏng biến đổi, như được minh họa trên Fig. 2. Ví dụ, trong trường hợp mà nội dung gốc có tốc độ khung 120 fps và góc màn trập 360 độ (nghĩa là góc màn trập mở 1/120 giây), bộ giải mã có thể kết xuất nhiều tốc độ khung mà nhỏ hơn hoặc bằng 120 fps. Ví dụ, như được mô tả trong Bảng 1, để giải mã 24 fps với góc màn trập mô phỏng 216 độ, bộ giải mã có thể kết hợp ba khung được giải mã và hiển thị ở 24 fps. Bảng 4 mở rộng dựa trên Bảng 1 và minh họa cách kết hợp số lượng các khung mã hóa khác nhau để kết xuất ở tốc độ khung đích đầu ra và các góc màn trập đích mong muốn. Việc kết hợp các khung có thể được thực hiện bằng cách tính trung bình điểm ảnh đơn thuần, bằng cách tính trung bình điểm ảnh có trọng số, trong đó các điểm ảnh từ một khung nhất định có thể được gán trọng số nhiều hơn các điểm ảnh của các khung khác và tổng của tất cả các trọng số cộng lại bằng một, hoặc bằng các sơ đồ nội suy bằng bộ lọc khác được biết đến trong lĩnh vực này. Trong Bảng 4, hàm Ce(a,b) biểu thị sự kết hợp của các khung hình mã hóa a đến b, trong đó sự kết hợp có thể được thực hiện bằng cách tính trung bình, tính trung bình có trọng số, lọc, và tương tự.

Bảng 4: Ví dụ về kết hợp các khung đầu vào ở 120 fps để tạo ra các khung đầu ra ở các giá trị fps đích và góc màn trập

Đầu vào	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10
Enc.	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10
Dec. 120fps										
@360	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10
Dec. 60fps										
@360	Ce(1,2)		Ce(3,4)		Ce(5,6)		Ce(7,8)		Ce(9,10)	
@180	e1		e3		e5		e7		e9	
Dec. 40fps										
@360	Ce(1,3)			Ce(4,6)			Ce(7,9)			Ce(10,12)
@240	Ce(1,2)			Ce(4,5)			Ce(7,8)			Ce(10,11)
@120	e1			e4			e7			e10
Dec. 30fps										
@360	Ce(1,4)				Ce(5,8)				Ce(9,12)	
@270	C(1,3)				Ce(5,7)				Ce(9,11)	
@180	Ce(1,2)				Ce(5,6)				Ce(9,10)	
@90	e1				e5				e9	
Dec. 24fps										
@360	Ce(1,5)					Ce(6,10)				

@288	Ce(1,4)	Ce(6,9)
@216	Ce(1,3)	Ce(6,8)
@144	Ce(1,2)	Ce(6,7)
@72	e1	e6

Khi giá trị của góc màn trập đích nhỏ hơn 360 độ, thì bộ giải mã có thể kết hợp các tập hợp khung giải mã khác nhau. Ví dụ, từ Bảng 1, cho trước dòng gốc 120 fps @ 360-độ, để tạo ra dòng ở 40 fps và góc màn trập 240 độ, bộ giải mã cần kết hợp hai khung trong số ba khung có thể có. Do đó, bộ giải mã có thể kết hợp hoặc khung thứ nhất và khung thứ hai hoặc khung thứ hai và khung thứ ba. Việc lựa chọn các khung nào để kết hợp có thể được mô tả liên quan đến thuật ngữ về “pha giải mã” được biểu diễn là:

$$\text{decode_phase} = \text{decode_phase_idx} * (360/n_frames), \quad (6)$$

trong đó decode_phase_idx chỉ báo chỉ số độ lệch trong tập hợp các khung tuần tự có các giá trị chỉ số trong $[0, n_frames_max-1]$, trong đó n_frames được cho bởi phương trình (2), và

$$n_frames_max = \text{orig_frame_rate}/\text{target_frame_rate}. \quad (7)$$

Nói chung, decode_phase_idx nằm trong khoảng từ $[0, n_frames_max-n_frames]$. Ví dụ, đối với chuỗi gốc ở 120 fps và góc màn trập 360 độ, đối với tốc độ khung đích 40 fps ở góc màn trập 240 độ, $n_frames_max = 120/40 = 3$. Từ phương trình (2), $n_frames = 2$, do vậy decode_phase_idx nằm trong khoảng từ $[0, 1]$. Do vậy, $\text{decode_phase_idx} = 0$ chỉ báo chọn các khung với chỉ số 0 và 1, và $\text{decode_phase_idx} = 1$ chỉ báo chọn các khung với chỉ số 1 và 2.

Theo phương án này, tốc độ khung biến đổi được kết xuất theo dự tính của người sáng tạo nội dung có thể được báo hiệu là siêu dữ liệu, như bản tin thông tin tăng cường bổ sung (SEI) hoặc như thông tin khả năng sử dụng video (VUI). Tùy ý, tốc độ khung được kết xuất có thể được điều khiển bởi bộ thu hoặc người dùng. Ví dụ về bản tin SEI chuyển đổi tốc độ khung chỉ ra tốc độ khung và góc màn trập thích hợp của người sáng tạo nội dung được thể hiện trong Bảng 5. Bản tin SEI có thể còn chỉ báo liệu việc kết hợp các khung được thực hiện trong miền tín hiệu mã hóa (ví dụ, gamma, PQ, v.v.) hay miền ánh sáng tuyến tính. Lưu ý rằng xử lý hậu kỳ yêu cầu bộ đệm khung bổ sung cho bộ đệm hình ảnh giải mã (decoder

picture buffer - DPB). Bản tin SEI có thể chỉ báo cần thêm bao nhiêu bộ đệm khung, hoặc một phương pháp khác nào đó để kết hợp các khung. Ví dụ, để làm giảm sự phức tạp, các khung có thể được kết hợp lại tại độ phân giải không gian giảm.

Như được thể hiện trong Bảng 4, tại các kết hợp nhất định của tốc độ khung và góc màn trập (ví dụ, ở 30 fps và 360 độ hoặc ở 24 fps và 288 hoặc 360 độ) bộ giải mã có thể cần kết hợp nhiều hơn ba khung giải mã, làm tăng số lượng không gian bộ đệm được yêu cầu bởi bộ giải mã. Để làm giảm gánh nặng về thêm không gian bộ đệm trong bộ giải mã, theo một số phương án, các kết hợp nhất định của tốc độ khung và góc màn trập có thể nằm ngoài các giới hạn ở tập hợp tham số giải mã được phép (ví dụ, bằng cách cài đặt các biên dạng và mức mã hóa thích hợp).

Xem xét lại, ví dụ, trường hợp phát lại ở 24 fps, bộ giải mã có thể quyết định hiển thị cùng một khung năm lần để hiển thị ở tốc độ khung đầu ra 120 fps. Điều này giống chính xác như thể hiện khung một lần ở tốc độ khung đầu ra 24 fps. Ưu điểm của việc giữ tốc độ khung đầu ra bất biến là ở chỗ màn hình có thể chạy ở tốc độ xung nhịp bất biến, khiến toàn bộ phần cứng trở nên đơn giản hơn nhiều. Nếu màn hình có thể thay đổi động tốc độ xung nhịp thì điều đó có thể hợp lý hơn khi chỉ thể hiện khung một lần (cho thứ 1/24 của giây) thay vì lặp lại cùng khung năm lần (mỗi thứ 1/120 của giây). Cách tiếp cận trước có thể dẫn đến chất lượng hình ảnh cao hơn một chút, hiệu suất quang học tốt hơn, hoặc hiệu suất năng lượng tốt hơn. Các cân nhắc tương tự cũng áp dụng được cho các tốc độ khung khác.

Bảng 5 mô tả ví dụ về cú pháp bản tin SEI chuyển đổi tốc độ khung theo một phương án.

Bảng 5: Ví dụ về cú pháp bản tin SEI cho phép chuyển đổi tốc độ khung

framerate_conversion(payloadSize) {	Mô tả
framerate_conversion_cancel_flag	u(1)
if(!frame_conversion_cancel_flag) {	
base_frame_rate	u(9)
base_shutter_angle	u(9)
decode_phase_idx_present_flag	u(1)
if(decode_phase_idx_present_flag) {	
decode_phase_idx	u(3)
}	
conversion_domain_idc	u(1)
num_frame_buffer	u(3)
framerate_conversion_persistence_flag	u(1)
}	
}	

framerate_conversion_cancel_flag bằng 1 chỉ báo bản tin SEI hủy bỏ tính liên tục của bản tin SEI chuyển đổi tốc độ khung bất kỳ trước đó theo thứ tự đầu ra. framerate_conversion_cancel_flag bằng 0 chỉ báo thông tin chuyển đổi tốc độ khung như sau. **base_frame_rate** chỉ ra tốc độ khung mong muốn.

base_shutter_angle chỉ ra góc màn trập mong muốn.

decode_phase_idx_present_flag bằng 1 chỉ ra thông tin pha giải mã là có mặt. decode_phase_idx_present_flag bằng 0 chỉ ra thông tin pha giải mã là không có mặt.

decode_phase_idx chỉ báo chỉ số độ lệch trong tập hợp các khung tuần tự có các giá trị chỉ số 0..(n_frames_max-1) trong đó $n_frames_max = 120/base_frame_rate$. Giá trị của decode_phase_idx sẽ nằm trong khoảng 0..(n_frames_max-n_frames), trong đó $n_frames = base_shutter_angle/(3*base_frame_rate)$. Khi decode_phase_idx là không có mặt, nó được suy là 0.

conversion_domain_idc bằng 0 chỉ ra sự kết hợp khung được thực hiện trong miền tuyến tính. **conversion_domain_idc** bằng 1 chỉ ra sự kết hợp khung được thực hiện trong miền phi tuyến tính.

num_frame_buffers chỉ ra số lượng bộ đệm khung bổ sung (không tính DPB).

framerate_conversion_persistence_flag chỉ ra tính liên tục của bản tin SEI chuyển đổi tốc độ khung cho lớp hiện thời. **framerate_conversion_persistence_flag** bằng 0 chỉ ra bản tin SEI chuyển đổi tốc độ khung chỉ áp dụng cho hình ảnh giải mã hiện thời. Đặt **picA** là hình ảnh hiện thời. **framerate_conversion_persistence_flag** bằng 1 chỉ ra bản tin SEI chuyển đổi tốc độ khung vẫn tồn tại cho lớp hiện thời theo thứ tự đầu ra đến khi một hoặc nhiều các điều kiện sau là đúng:

- Chuỗi video theo lớp được mã hóa (coded layer-wise video sequence - CLVS) mới của lớp hiện thời bắt đầu.
- Dòng bit kết thúc.
- Hình ảnh **picB** trong lớp hiện thời trong đơn vị truy cập chứa bản tin SEI chuyển đổi tốc độ khung hình mà có thể áp dụng cho lớp hiện thời được xuất ra cho **PicOrderCnt(picB)** nào lớn hơn **PicOrderCnt(picA)**, trong đó **PicOrderCnt(picB)** và **PicOrderCnt(picA)** lần lượt là các giá trị **PicOrderCntVal** của **picB** và **picA**, ngay sau khi dẫn ra quy trình giải mã để đếm thứ tự hình ảnh cho **picB**.

Phương án thứ ba – Đầu vào được mã hóa ở nhiều góc màn trập

Phương án thứ ba là sơ đồ mã hóa cho phép trích xuất tốc độ khung con từ dòng bit, do đó hỗ trợ khả năng tương thích ngược. Trong HEVC, điều này đạt được bằng khả năng định tỷ lệ theo thời gian. Khả năng định tỷ lệ lớp thời gian được cho phép bằng cách gán các giá trị khác nhau cho phần tử cú pháp **temporal_id** cho các khung giải mã. Do đó, dòng bit có thể được trích xuất một cách đơn giản trên cơ sở các giá trị **temporal_id**. Tuy nhiên, cách tiếp cận kiểu HEVC đối với khả năng định tỷ lệ theo thời gian không cho phép kết xuất tốc độ khung đầu ra với góc màn trập khác nhau. Ví dụ, tốc độ khung cơ sở 60 fps được trích xuất từ tốc độ khung gốc 120 fps sẽ luôn có góc màn trập 180 độ.

Trong ATSC 3.0, một phương pháp khác được mô tả trong đó các khung tại 60 fps có góc màn trập 360 độ được mô phỏng là trung bình có trọng số của hai khung 120 fps. Các khung 60 fps được mô phỏng được gán giá trị **temporal_id** bằng 0 và được kết hợp với các

khung 120 fps gốc đan xen được gán giá trị `temporal_id` bằng 1. Khi 60 fps được yêu cầu, bộ giải mã chỉ yêu cầu giải mã các khung với `temporal_id` 0. Khi cần 120 fps, thì bộ giải mã có thể trừ mỗi khung `temporal_id` = 1 (tức là, khung 120 fps) từ phiên bản được định tỷ lệ của mỗi khung `temporal_id` = 0 tương ứng (tức là, khung 60 fps được mô phỏng) để khôi phục khung 120 fps gốc tương ứng mà không được truyền rõ ràng, do đó tạo lại tất cả các khung 120 fps gốc.

Theo một số phương án, sáng chế mô tả thuật toán mới hỗ trợ nhiều tốc độ khung đích và góc màn trập đích theo cách tương thích ngược (backward compatible - BC). Đề xuất là xử lý trước nội dung 120 fps gốc ở tốc độ khung cơ sở ở một số góc màn trập. Sau đó, tại bộ giải mã, các tốc độ khung khác ở nhiều góc màn trập khác nhau có thể được suy ra một cách đơn giản. Cách tiếp cận ATSC 3.0 có thể được coi là một trường hợp đặc biệt của sơ đồ được đề xuất, trong đó các khung có `temporal_id`=0 mang các khung ở góc màn trập 60fps@360 và các khung có `temporal_id`=1 mang các khung ở góc màn trập 60fps@180.

Theo ví dụ thứ nhất, như được thể hiện trên Fig. 4, xem xét chuỗi đầu vào ở 120 fps và góc màn trập 360 mà được sử dụng để mã hóa chuỗi với tốc độ khung lớp cơ sở 40 fps và góc màn trập ở 120, 240, và 360 độ. Theo sơ đồ này bộ giải mã tính toán các khung mới bằng cách kết hợp nhiều nhất ba trong số các khung đầu vào gốc. Ví dụ, khung hình mã hóa 2 (En-2) biểu diễn đầu vào ở 40 fps và 240 độ được tạo ra bằng cách kết hợp các khung đầu vào 1 và 2, và khung hình mã hóa 3 (En-3) biểu diễn đầu vào ở 40 fps và 360 độ được tạo ra bằng cách kết hợp khung En-2 với khung đầu vào 3. Trong bộ giải mã, để tái tạo chuỗi đầu vào, khung giải mã 2 (Dec-2) được tạo ra bằng cách trừ khung En-1 từ khung En-2, và khung giải mã 3 (Dec-3) được tạo ra bằng cách trừ khung En-2 từ khung En-3. Ba khung giải mã biểu diễn đầu ra ở tốc độ khung cơ sở 120 fps và góc màn trập 360 độ. Tốc độ khung và góc màn trập bổ sung có thể được ngoại suy bằng cách sử dụng các khung giải mã như được thể hiện trong Bảng 6. Trong Bảng 6, hàm số $Cs(a,b)$ biểu thị việc kết hợp các khung đầu vào a đến b, trong đó việc kết hợp có thể được thực hiện bằng cách tính trung bình, tính trung bình có trọng số, lọc, và tương tự.

Bảng 6: Ví dụ về việc kết hợp khung với đường cơ sở 40 fps

Các khung đầu vào 120fps @360	s1	s2	s3	s4	s5	s6	s7	s8	s9
Mã hóa khung 120fps	e1= s1	e2= Cs(1, 2)	e3 = Cs(1, 3)	e4=s4	e5= Cs(4, 5)	e6= Cs(4, 6)	e7=s7	e8 = Cs(7, 8)	e9 = Cs(7, 9)
Giải mã 120fps									
@360	e1= s1	e2-e1 = s2	e3- e2=s3	e4 = s4	e5- e4=s5	e6-e4 =s6	e7 = s7	e8- e7=s8	e9-e8 =s9
Giải mã 60fps									
@360	e2	e3-e2+e4 =Cs(3,4)		e6-e4=Cs(5,6)		e8 = Cs(7,8)		e9- e8+e1 0	
@180	e1	e3-e2=s3		e5-e4=s5		e7		e9-e8	
Giải mã 40fps									
@360	e3 =Cs(1,3)			e6			e9		
@240	e2=Cs(1,2)			e5			e8		
@120	e1=s1			e4			e7		

Giải mã 30fps			
@360	$e3+e4 = Cs(1,4)$	$e6-e4+e8 = Cs(5,8)$	$e9-e8+e1$ 2
@270	$e3 = Cs(1,3)$	$e6-e5+e7 = Cs(5,7)$	$e9-e8+e1$ 1
@180	$e2 = Cs(1,2)$	$e6-e4=Cs(5,6)$	$e9-e8+e1$ 0
@90	$e1$	$e5-e4 = s5$	$e9-e8$
Giải mã 24fps			
@360	$e3+e5 = Cs(1,5)$	$e6-e5+e9+e10 = Cs(6,10)$	
@288	$e3+e4 = Cs(1,4)$	$e6-e5+e9 = Cs(6,9)$	
@216	$e3 = Cs(1,3)$	$e6-e5+e8 = Cs(6,8)$	
@144	$e2 = Cs(1,2)$	$e6-e5+e7=Cs(6,7)$	
@72	$e1 = s1$	$e6-e5 = s6$	

Ưu điểm của cách tiếp cận này là ở chỗ, như thể hiện trong Bảng 6, tất cả các phiên bản 40 fps có thể được giải mã mà không cần xử lý thêm. Một ưu điểm khác là ở chỗ các tốc độ khung khác có thể được suy ra ở các góc màn trập khác nhau. Ví dụ, xem xét bộ giải mã giải mã ở 30 fps và góc màn trập 360. Từ Bảng 4, đầu ra tương ứng với chuỗi các khung được tạo ra bởi $Ce(1,4) = Cs(1,4)$, $Cs(5,8)$, $Cs(9,12)$, và tương tự, mà cũng khớp với chuỗi giải mã được thể hiện trong Bảng 6, tuy nhiên, trong Bảng 6, $Cs(5,8) = e6-e4+e8$. Theo một phương án, các bảng tra cứu (LUT) có thể được sử dụng để xác định cách các khung giải mã cần được kết hợp để tạo ra chuỗi đầu ra ở tốc độ khung đầu ra được chỉ ra và góc màn trập được mô phỏng.

Trong một ví dụ khác, đề xuất kết hợp nhiều nhất năm khung trong bộ mã hóa để đơn giản hóa việc trích xuất lớp cơ sở 24 fps ở góc màn trập 72, 144, 216, 288, và 360 độ, như được thể hiện dưới đây. Đây là mong muốn đối với nội dung phim ảnh mà được trình diễn tốt nhất ở 24fps trên các tivi kế thừa.

Bảng 7: Ví dụ về việc kết hợp khung với đường cơ sở 24 fps

Đầu vào khung 120fps @360	s1	s2	s3	s4	s5	s6	s7	s8	s9
Enc. khung	e1 = s1	e2 = Cs(1,2)	e3 = Cs(1,3)	e4 = Cs(1,4)	e5 = Cs(1,5)	e6 = s6	e7 = Cs(6,7)	e8 = Cs(6,8)	e9 = Cs(6,9)
Giải mã 120fps @360	e1	e2-e1	e3-e2	e4-e3	e5-e4	e6	e7-e6	e8-e7	e9-e8
Giải mã 60fps @360	e2		e4-e2		e5-e4+e6		e8-e6		e10-e8
@180	e1		e3-e2		e5-e4		e7-e6		e9-e8
Giải mã 40fps @360	e3			e5-e3+e6			e9-e6		

@240	e2			e5-e3			e8-e6		
@120	e1			e4-e3			e7-e6		
Giải mã 30fps									
@360	e4			e5-e4+e8			e10-e8+e1 2		
@270	e3			e5-e4+e7			e10-e8+e1 1		
@180	e2			e5-e4+e6			e10-e8		
@90	e1			e5-e4			e9-e8		
Decode 24fps									
@360	e5					e10			
@288	e4					e9			
@216	e3					e8			
@144	e2					e7			
@72	e1					e6			

Như được thể hiện trong Bảng 7, nếu tốc độ khung giải mã khớp với tốc độ khung đường cơ sở (24 fps), thì, trong mỗi nhóm năm khung (ví dụ, e1 đến e5) bộ giải mã có thể đơn giản lựa chọn một khung ở góc màn trập mong muốn (ví dụ, e2 cho góc màn trập ở 144 độ). Để giải mã ở tốc độ khung khác nhau và góc màn trập cụ thể, bộ giải mã sẽ cần xác định

cách kết hợp chính xác (tức là, bằng cách cộng hoặc trừ) các khung giải mã. Ví dụ, để giải mã tại 30 fps và góc màn trập 180 độ, các bước sau có thể là như sau:

a) Bộ giải mã có thể xem xét bộ mã hóa giả định truyền ở 120 fps và 360 độ mà không cần xem xét đến khả năng tương thích ngược, sau đó, từ Bảng 1, bộ giải mã cần kết hợp 2 trong số 4 khung để tạo ra chuỗi đầu ra ở tốc độ khung và góc màn trập mong muốn. Ví dụ, như được thể hiện trong Bảng 4, chuỗi bao gồm, $Ce(1,2) = Avg(s1, s2)$, $Ce(5,6) = Avg(s5, s6)$, và tương tự, trong đó $Avg(s1,s2)$ có thể biểu thị trung bình của các khung $s1$ và $s2$.

b) Cho trước rằng theo định nghĩa các khung hình mã hóa có thể được biểu diễn là $e1 = s1$, $e2 = Avg(s1, s2)$, $e3 = Avg(s1, s3)$, và tương tự, có thể dễ dàng suy ra rằng chuỗi các khung trong bước a) có thể còn được biểu diễn dưới dạng:

- $Ce(1,2) = Avg(s1,s2) = e2$
- $Ce(5,6) = Avg(s5,s6) = Avg(s1,s5) - Avg(s1,s4) + s6 = e5 - e4 + e6$
- v.v.

Như nêu trên đây, sự kết hợp chính xác của các khung giải mã có thể được tính toán trước và là khả dụng như LUT.

Ưu điểm của phương pháp đề xuất là ở chỗ nó cung cấp các lựa chọn cho cả người sáng tạo và người dùng nội dung; tức là, nó cho phép lựa chọn định hướng/chỉnh sửa và lựa chọn người dùng. Ví dụ, việc xử lý trước nội dung trong bộ giải mã cho phép tốc độ khung cơ sở được sáng tạo với các góc màn trập khác nhau. Mỗi góc màn trập có thể được gán giá trị `temporal_id` trong khoảng $[0, (n_frames - 1)]$, trong đó `n_frames` có giá trị bằng 120 chia cho tốc độ khung cơ sở. (Ví dụ, với tốc độ khung cơ sở 24 fps, `temporal_id` nằm trong khoảng $[0,4]$.) Sự lựa chọn có thể được thực hiện để tối ưu hóa hiệu suất nén, hoặc vì các lý do thẩm mỹ. Trong một số trường hợp sử dụng, giả sử, để vượt qua dòng trên cùng, nhiều dòng bit với các lớp cơ sở khác nhau có thể được mã hóa và lưu trữ và được đưa ra để người dùng lựa chọn.

Trong ví dụ thứ hai về các phương pháp được bộc lộ, nhiều tốc độ khung tương thích ngược có thể được hỗ trợ. Lý tưởng nếu người có hiểu biết trung bình trong lĩnh vực này có thể muốn có khả năng giải mã ở 24 khung/giây để có được lớp cơ sở 24 fps, ở 30 khung/giây để có được chuỗi 30 fps, ở 60 khung/giây để có được chuỗi 60 fps, và tương tự. Nếu góc màn trập đích không được xác định, thì góc màn trập đích mặc định, trong số các

góc màn trập phù hợp với các tốc độ khung nguồn và đích, gần nhất có thể với 180 độ được đề xuất. Ví dụ, với các giá trị thể hiện trong Bảng 7, góc màn trập đích được ưu tiên cho fps tại 120, 60, 40, 30, và 24 là 360, 180, 120, 180, và 216 độ.

Từ các ví dụ trên đây, có thể thấy rằng việc chọn cách mã hóa nội dung có thể ảnh hưởng đến độ phức tạp trong việc giải mã các tốc độ khung lớp cơ sở cụ thể. Một phương án của sáng chế là chọn thích ứng sơ đồ mã hóa dựa vào tốc độ khung lớp cơ sở mong muốn. Đối với nội dung phim ảnh tốc độ này có thể là 24 fps, ví dụ, trong khi đó đối với thể thao nó có thể là 60 fps.

Cú pháp ví dụ cho phương án BC của sáng chế được thể hiện dưới đây và trong các Bảng 8 và 9.

Trong SPS (Bảng 8), hai phần tử cú pháp được bổ sung: `sps_hfr_BC_enabled_flag`, và `sps_base_framerate` (nếu `sps_hfr_BC_enabled_flag` được đặt bằng 1).

`sps_hfr_BC_enabled_flag` bằng 1 chỉ ra tốc độ khung cao với khả năng tương thích ngược được phép trong chuỗi video mã hóa (CVS). `sps_hfr_BC_enabled_flag` bằng 0 chỉ ra tốc độ khung cao với khả năng tương thích ngược không được phép trong CVS.

`sps_base_framerate` chỉ ra tốc độ khung cơ sở cho CVS hiện thời.

Trong phần đầu nhóm ô, nếu `sps_hfr_BC_enabled_flag` được đặt bằng 1, cú pháp `number_avg_frames` được gửi đi trong dòng bit.

`number_avg_frames` chỉ ra số lượng khung ở tốc độ khung cao nhất (ví dụ, 120 fps) mà được kết hợp để tạo ra hình ảnh hiện thời ở tốc độ khung cơ sở.

Bảng 8: Ví dụ về Cú pháp RBSP cho đầu vào ở các góc màn trập khác nhau

<code>seq_parameter_set_rbsp() {</code>	Mô tả
<code>sps_max_sub_layers_minus1</code>	u(3)
<code>sps_reserved_zero_5bits</code>	u(5)
<code>profile_tier_level(sps_max_sub_layers_minus1)</code>	
<code>sps_seq_parameter_set_id</code>	ue(v)
...	
<code>sps_hfr_BC_enabled_flag</code>	u (1)
<code>if(sps_hfr_BC_enabled_flag) {</code>	u (1)

sps_base_frame_rate	u(9)
}	
...	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

Bảng 9: Ví dụ về cú pháp RBSB của tập hợp tham số hình ảnh cho đầu vào ở các góc màn trập khác nhau

pic_parameter_set_rbsp() {	Mô tả
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
...	
if(sps_hfr_BC_enabled_flag)	
number_avg_frames	
...	se(v)
rbsp_trailing_bits()	
}	

Các thay đổi đối với phương án thứ hai (tốc độ khung cố định)

Chuẩn mã hóa HEVC (H.265) (Ref.[1]) và chuẩn mã hóa video đa năng đang phát triển (thường được gọi là VVC, xem Ref.[2]), xác định phần tử cú pháp, **pic_struct**, chỉ báo liệu hình ảnh nên được hiển thị là khung hay là một hoặc nhiều trường, và xem hình ảnh được giải mã có nên được lặp lại không. Bản sao của Bảng D.2, “Interpretation of pic_struct,” từ HEVC được cung cấp để dễ tham chiếu trong Phụ lục.

Quan trọng cần lưu ý là, theo đánh giá của các tác giả sáng chế, phần tử cú pháp **pic_struct** hiện có có thể chỉ hỗ trợ tập hợp con cụ thể của tốc độ khung nội dung khi sử dụng

vùng chứa mã hóa tốc độ khung cố định. Ví dụ, khi sử dụng vùng chứa tốc độ khung cố định 60 fps, cú pháp `pic_struct` hiện có, khi `fixed_pic_rate_within_cvs_flag` bằng 1, có thể hỗ trợ 30 fps bằng cách sử dụng kỹ thuật gấp đôi khung, và 24 fps bằng cách sử dụng kỹ thuật gấp đôi khung và gấp ba khung trong sự kết hợp đan xen trên mỗi khung khác. Tuy nhiên, khi sử dụng vùng chứa tốc độ khung cố định 120 fps, cú pháp `pic_struct` hiện thời có thể không hỗ trợ tốc độ khung 24 fps hoặc 30 fps. Để giảm bớt vấn đề này, hai phương pháp mới được đề xuất: một là mở rộng phiên bản HEVC, và phương pháp còn lại là không mở rộng.

Phương pháp 1: `pic_struct` mà không có khả năng tương thích ngược

VVC vẫn đang phát triển, do đó người có hiểu biết trung bình trong lĩnh vực có thể thiết kế cú pháp với sự tự do tối đa. Theo một phương án, trong `pic_struct`, đề xuất loại bỏ các tùy chọn gấp đôi khung và gấp ba khung, sử dụng giá trị cụ thể của `pic_struct` để chỉ báo sự lặp lại khung tùy ý, và bổ sung phần tử cú pháp mới, **`num_frame_repetition_minus2`**, chỉ ra số lượng khung cần lặp lại. Ví dụ về cú pháp đề xuất được mô tả trong các bảng sau đây, trong đó bảng 10 biểu thị các thay đổi so với bảng D.2.3 trong HEVC và Bảng 11 biểu thị các thay đổi của Bảng D.2 được thể hiện trong Phụ lục.

Bảng 10: Ví dụ về cú pháp bản tin SEI định thời hình ảnh, phương pháp 1

<code>pic_timing(payloadSize) {</code>	Mô tả
<code>if(frame_field_info_present_flag) {</code>	
<code>pic_struct</code>	<code>u(4)</code>
<code>if(pic_struct == 7)</code>	<code>u(4)</code>
<code>num_frame_repetition_minus2</code>	<code>u(4)</code>
<code>source_scan_type</code>	<code>u(2)</code>
<code>duplicate_flag</code>	<code>u(1)</code>
<code>}</code>	
.... (Như bản gốc)	

`num_frame_repetition_minus2` cộng 2 chỉ báo rằng khi `fixed_pic_rate_within_cvs_flag` bằng 1, khung nên được hiển thị **`num_frame_repetition_minus2`** cộng 2 lần một cách liên tiếp trên các màn hình với khoảng thời gian làm mới khung bằng `DpbOutputElementalInterval[ n ]` như được cho bởi Phương trình E-73.

Bảng 11: Ví dụ về pic_struct sửa đổi theo phương pháp 1

Giá trị	Màn hiển thị hình ảnh được chỉ báo	Các giới hạn
0.	Khung (tăng dần)	field_seq_flag sẽ bằng 0
1.	Trường trên cùng	field_seq_flag sẽ bằng 1
2.	Trường dưới cùng	field_seq_flag sẽ bằng 1
3.	Trường trên cùng, trường dưới cùng, theo thứ tự đó	field_seq_flag sẽ bằng 0
4.	Trường dưới cùng, trường trên cùng, theo thứ tự đó	field_seq_flag sẽ bằng 0
5.	Trường trên cùng, trường dưới cùng, trường trên cùng được lặp lại, theo thứ tự đó	field_seq_flag sẽ bằng 0
6.	Trường dưới cùng, trường trên cùng, trường dưới cùng được lặp lại, theo thứ tự đó	field_seq_flag sẽ bằng 0
7.	Sự lặp lại khung	field_seq_flag sẽ bằng 0 fixed_pic_rate_within_cvs_flag sẽ bằng 1
8.	Trường trên cùng ghép cặp với trường dưới cùng trước đó theo thứ tự đầu ra	field_seq_flag sẽ bằng 1
9.	Trường dưới cùng ghép cặp với trường trên cùng trước đó theo thứ tự đầu ra	field_seq_flag sẽ bằng 1
10.	Trường trên cùng ghép cặp với trường dưới cùng tiếp theo theo thứ tự đầu ra	field_seq_flag sẽ bằng 1
11.	Trường dưới cùng ghép cặp với trường trên cùng tiếp theo theo thứ tự đầu ra	field_seq_flag sẽ bằng 1

Phương pháp 2: Phiên bản mở rộng của phiên bản HEVC của pic_struct

Các bộ giải mã AVC và HEVC đã được triển khai, do đó có thể mong muốn mở rộng đơn giản cú pháp pic_struct hiện có mà không loại bỏ các tùy chọn cũ. Theo một phương án, pic_struct mới = 13, giá trị “mở rộng lặp lại khung”, và phân tử cú pháp mới, num_frame_repetition_minus4, được bổ sung. Ví dụ về cú pháp đề xuất được mô tả trong

các Bảng 12 và 13. Đối với các giá trị `pic_struct` 0 đến 12, cú pháp đề xuất là giống với cú pháp trong Bảng D.2 (như được thể hiện trong Phụ lục), do đó các giá trị này được lược bỏ để làm đơn giản.

Bảng 12: Ví dụ về cú pháp bản tin SEI định thời hình ảnh, phương pháp 2

<code>pic_timing(payloadSize) {</code>	Mô tả
<code> if(frame_field_info_present_flag) {</code>	
<code> pic_struct</code>	u(4)
<code> if(pic_struct == 13)</code>	u(4)
<code> num_frame_repetition_minus4</code>	u(4)
<code> source_scan_type</code>	u(2)
<code> duplicate_flag</code>	u(1)
<code> }</code>	
... (như bản gốc)	

`num_frame_repetition_minus4` cộng 4 chỉ báo rằng khi `fixed_pic_rate_within_cvs_flag` bằng 1, khung nên được hiển thị `num_frame_repetition_minus4` cộng 4 lần một cách liên tiếp trên các màn hình với khoảng thời gian làm mới khung bằng `DpbOutputElementalInterval[ n ]` như được cho bởi phương trình E-73.

Bảng 13: Ví dụ về `pic_struct` sửa đổi, phương pháp 2

Giá trị	Màn hiển thị hình ảnh được chỉ báo	Các giới hạn
0 ± 12	Như trong Bảng D.2	Như trong Bảng D.2
13.	Mở rộng sự lặp lại khung	<code>field_seq_flag</code> sẽ bằng 0 <code>fixed_pic_rate_within_cvs_flag</code> sẽ bằng 1

Trong HEVC, tham số `frame_field_info_present_flag` có mặt trong thông tin khả năng sử dụng video (VUI), nhưng các phần tử cú pháp `pic_struct`, `source_scan_type`, và `duplicate_flag` nằm trong bản tin SEI `pic_timing()`. Theo một phương án, đề xuất chuyển tất

cả các phần tử cú pháp liên quan đến VUI, cùng với `frame_field_info_present_flag`. Ví dụ về cú pháp đề xuất được thể hiện trong Bảng 14.

Bảng 14: Ví dụ về cú pháp tham số VUI với sự hỗ trợ cho phần tử cú pháp `pic_struct` sửa đổi

<code>vui_parameters() {</code>	Mô tả
<code>...</code>	<code>u(1)</code>
<code>field_seq_flag</code>	<code>u(1)</code>
<code>frame_field_info_present_flag</code>	<code>u(1)</code>
<code>if(frame_field_info_present_flag) {</code>	
<code>pic_struct</code>	<code>u(4)</code>
<code>source_scan_type</code>	<code>u(2)</code>
<code>duplicate_flag</code>	<code>u(1)</code>
<code>}</code>	
<code>...</code>	
<code>}</code>	

Báo hiệu luân phiên thông tin góc màn trập

Khi xử lý tốc độ khung biến đổi, mong muốn nhận dạng cả tốc độ khung mong muốn và góc màn trập mong muốn. Các chuẩn mã hóa video đã biết, “thông tin khả năng sử dụng video” (VUI) cung cấp thông tin thiết yếu cho màn hình nội dung video thích hợp, như tỷ lệ co, các màu gốc, việc lấy mẫu phụ độ màu, v.v. VUI có thể còn cung cấp thông tin tốc độ khung nếu tốc độ hình cố định được thiết lập bằng 1; tuy nhiên, không có sự hỗ trợ đối với thông tin góc màn trập. Các phương án cho phép sử dụng các góc màn trập khác nhau cho các lớp thời gian khác nhau, và bộ giải mã có thể sử dụng thông tin góc màn trập để cải thiện sự nhìn cuối cùng trên màn hình.

Ví dụ, HEVC hỗ trợ các lớp con thời gian mà sử dụng cơ bản các kỹ thuật lược bỏ khung để đi từ tốc độ khung cao hơn xuống tốc độ khung thấp hơn. Vấn đề chủ yếu với điều này là ở chỗ góc màn trập hiệu quả bị làm giảm với mỗi lần bỏ khung. Ví dụ, 60 fps có thể được suy ra từ video 120 fps bằng cách bỏ mỗi khung khác; 30 fps có thể được suy ra bằng cách bỏ 3 trong số 4 khung; và 24 fps có thể được suy ra bằng cách bỏ 4 trong số 5 khung.

Giả sử màn trập 360 độ đầy đủ cho 120Hz, với kỹ thuật bỏ khung đơn giản, góc màn trập cho 60 fps, 30 fps, và 24 fps lần lượt là 180, 90, và 72 độ [3]. Kinh nghiệm đã chứng minh rằng các góc màn trập dưới 180 độ thường không phù hợp, nhất là với các tốc độ khung dưới 50 Hz. Bằng cách cung cấp thông tin góc màn trập, ví dụ, nếu mong muốn là màn hình tạo ra hiệu ứng điện ảnh từ video 120 Hz với góc màn trập được giảm xuống cho mỗi lớp thời gian, các kỹ thuật thông minh có thể được áp dụng để cải thiện sự nhìn cuối cùng.

Trong một ví dụ khác, có thể muốn hỗ trợ lớp thời gian khác (tức là, dòng bit con 60 fps bên trong dòng bit 120 fps) với cùng góc màn trập. Sau đó, vấn đề chủ yếu là ở chỗ khi video 120 fps được hiển thị tại 120Hz, các khung chẵn/lẻ có góc màn trập hiệu quả khác nhau. Nếu màn hình có thông tin liên quan, thì các kỹ thuật thông minh có thể được áp dụng để cải thiện sự nhìn cuối cùng. Ví dụ về cú pháp đề xuất được thể hiện trong Bảng 15, trong đó bảng cú pháp tham số VUI E.2.1 trong HEVC (Ref. [1]) được sửa đổi để hỗ trợ thông tin góc màn trập như được lưu ý. Lưu ý rằng theo một phương án khác, thay vì biểu diễn cú pháp shutter_angle bằng các độ tuyệt đối, nó có thể được biểu diễn khác dưới dạng tỷ lệ của tốc độ khung với tốc độ màn trập (xem phương trình (1)).

Bảng 15: Ví dụ về cú pháp tham số VUI với sự hỗ trợ góc màn trập

vui_parameters() {	Mô tả
...	
vui_timing_info_present_flag	u(1)
if(vui_timing_info_present_flag) {	
vui_num_units_in_tick	u(32)
vui_time_scale	u(32)
vui_poc_proportional_to_timing_flag	u(1)
if(vui_poc_proportional_to_timing_flag)	
vui_num_ticks_poc_diff_one_minus1	ue(v)
vui_hrd_parameters_present_flag	u(1)
if(vui_hrd_parameters_present_flag)	
hrd_parameters(1, sps_max_sub_layers_minus1)	
}	
vui_shutter_angle_info_present_flag	u(1)
if(vui_shutter_angles_info_present_flag) {	
fixed_shutter_angle_within_cvs_flag	u(1)
if(fixed_shutter_angle_within_cvs_flag)	
fixed_shutter_angle	u(9)
else {	
for(i = 0; i <= sps_max_sub_layers_minus1; i++) {	
sub_layer_shutter_angle[i]	u(9)
}	
}	
...	
}	

vui_shutter_angle_info_present_flag bằng 1 chỉ ra rằng thông tin góc màn trập có mặt trong cấu trúc cú pháp vui_parameters(). **vui_shutter_angle_info_present_flag** bằng 0 chỉ ra rằng thông tin góc màn trập không có mặt trong cấu trúc cú pháp vui_parameters().

fixed_shutter_angle_within_cvs_flag bằng 1 chỉ ra rằng thông tin góc màn trập là giống nhau với tất cả các lớp con thời gian trong CVS. **fixed_shutter_angle_within_cvs_flag** bằng 0 chỉ ra rằng thông tin góc màn trập có thể không giống nhau với tất cả các lớp con thời gian trong CVS.

fixed_shutter_angle chỉ ra góc màn trập bằng độ trong CVS. Giá trị của **fixed_shutter_angle** sẽ nằm trong khoảng từ 0 đến 360.

sub_layer_shutter_angle[i] chỉ ra góc màn trập bằng độ khi HighestTid bằng i. Các giá trị của **sub_layer_shutter_angle[i]** sẽ nằm trong khoảng từ 0 đến 360.

Cập nhật tốc độ khung dần dần trong chuỗi video mã hóa (CVS)

Các thí nghiệm đã chứng minh rằng đối với nội dung HDR hiển thị trên màn hình HDR, để nhận thức sự rung do chuyển động giống như phát lại dải động chuẩn (standard dynamic range - SDR) trong màn hình 100 nit, tốc độ khung cần được tăng lên dựa vào độ sáng của nội dung. Trong hầu hết các chuẩn (AVC, HEVC, VVC, v.v.), tốc độ khung video có thể được chỉ báo trong VUI (được chứa trong SPS) sử dụng các phân tử cú pháp

vui_time_scale, **vui_num_units_in_tick** và

elemental_duration_in_tc_minus1[temporal_id_max], ví dụ, như được thể hiện trong Bảng 16 dưới đây (xem phần E.2.1 trong Ref.[1]).

Bảng 16: Các phân tử cú pháp VUI để chỉ báo tốc độ khung trong HEVC

vui_parameters() {	Mô tả
...	
vui_timing_info_present_flag	u(1)
if(vui_timing_info_present_flag) {	
vui_num_units_in_tick	u(32)
vui_time_scale	u(32)
vui_poc_proportional_to_timing_flag	u(1)
if(vui_poc_proportional_to_timing_flag)	

vui_num_ticks_poc_diff_one_minus1	ue(v)
vui_hrd_parameters_present_flag	u(1)
if(vui_hrd_parameters_present_flag)	
hrd_parameters(1, sps_max_sub_layers_minus1)	
}	
....	

Như được mô tả trong Ref. [1],

Biến ClockTick được suy ra như sau và được gọi là nhịp đồng hồ:

$$\text{ClockTick} = \text{vui_num_units_in_tick} \div \text{vui_time_scale}$$

$$\text{picture_duration} = \text{ClockTick} * (\text{elemental_duration_in_tc_minus1}[i] + 1)$$

$$\text{frame_rate} = 1/\text{pic_duration}.$$

Tuy nhiên, tốc độ khung có thể chỉ được thay đổi tại các thời điểm cụ thể, ví dụ, trong HEVC, duy nhất tại các khung điểm truy cập ngẫu nhiên nội ảnh (intra random access point - IRAP) hoặc tại lúc bắt đầu của CVS mới. Với phát lại HDR, khi có trường hợp làm rõ hoặc làm mờ, do độ sáng của hình đang thay đổi từng khung, có thể cần phải thay đổi tốc độ khung hoặc thời khoảng hình ảnh cho mỗi hình ảnh. Để cho phép làm mới tốc độ khung hoặc thời khoảng hình ảnh tại thời điểm bất kỳ (thậm chí trên cơ sở từng khung), theo một phương án, bản tin SEI mới cho “tốc độ làm mới dần dần” được đề xuất, như được thể hiện trong Bảng 17.

Bảng 17: Cú pháp làm ví dụ để hỗ trợ tốc độ khung làm mới dần dần trong việc bản tin SEI

gradual_refresh_rate(payloadSize) {	Mô tả
num_units_in_tick	u(32)
time_scale	u(32)
}	

Định nghĩa về cú pháp mới num_units_in_tick là giống như vui_num_units_in_tick, và định nghĩa về time_scale là giống như định nghĩa của vui_time_scale.

num_units_in_tick là số lượng đơn vị thời gian của đồng hồ hoạt động ở tần số **time_scale** Hz mà tương ứng với một lượng gia (được gọi là nhịp đồng hồ) của bộ đếm nhịp đồng hồ. **num_units_in_tick** sẽ lớn hơn 0. Nhịp đồng hồ, tính theo đơn vị giây, bằng thương số của **num_units_in_tick** chia cho **time_scale**. Ví dụ, khi tốc độ hình ảnh của tín hiệu video là 25 Hz, **time_scale** có thể bằng 27 000 000 và **num_units_in_tick** có thể bằng 1 080 000 và do vậy nhịp đồng hồ có thể bằng 0,04 giây.

time_scale là số lượng đơn vị thời gian trôi qua trong một giây. Ví dụ, hệ tọa độ thời gian mà đo thời gian sử dụng đồng hồ 27MHz có **time_scale** bằng 27 000 000. Giá trị của **time_scale** sẽ lớn hơn 0.

Thời gian hiển thị hình ảnh cho hình ảnh mà sử dụng bản tin SEI **gradual_refresh_rate** được xác định là:

$$\text{picture_duration} = \text{num_units_in_tick} \div \text{time_scale}.$$

Báo hiệu thông tin góc màn trập thông qua bản tin SEI

Như được thảo luận trên đây, bảng 15 cung cấp ví dụ về cú pháp tham số VUI với sự hỗ trợ góc màn trập. Ví dụ, và không giới hạn ở ví dụ, bảng 18 liệt kê các phần tử cú pháp giống nhau, nhưng bây giờ là một phần của bản tin SEI cho thông tin góc màn trập. Lưu ý rằng bản tin SEI chỉ được sử dụng làm ví dụ và bản tin tương tự có thể được tái tạo tại các lớp khác của cú pháp mức cao, như tập hợp tham số chuỗi (SPS), tập hợp tham số hình ảnh (PPS), phần đầu nhóm lát hoặc ô, và tương tự.

Bảng 18: Cú pháp bản tin SEI làm ví dụ cho thông tin góc màn trập

shutter_angle_information (payloadSize) {	Mô tả
fixed_shutter_angle_within_cvs_flag	u(1)
if (fixed_shutter_angle_within_cvs_flag)	
fixed_shutter_angle	u(9)
else {	
for(i = 0; i <= sps_max_sub_layers_minus1; i++)	
sub_layer_shutter_angle[i]	u(9)
}	
}	

Góc màn trập thường được biểu diễn bằng độ từ 0 đến 360 độ. Ví dụ, góc màn trập 180 độ chỉ báo thời khoảng phơi sáng là $\frac{1}{2}$ thời khoảng khung. Góc màn trập có thể được biểu diễn là: $\text{shutter_angle} = \text{frame_rate} * 360 * \text{shutter_speed}$, trong đó shutter_speed là thời khoảng phơi sáng và frame_rate là tỷ lệ nghịch của thời khoảng khung. Frame_rate cho lớp con thời gian cho trước Tid có thể được chỉ báo bởi **num_units_in_tick**, **time_scale**, **elemental_duration_in_tc_minus1**[Tid]. Ví dụ, khi $\text{fixed_pic_rate_within_cvs_flag}[\text{Tid}]$ bằng 1:

$$\text{frame_rate} = \text{time_scale} / (\text{num_units_in_tick} * (\text{elemental_duration_in_tc_minus1}[\text{Tid}] + 1))$$

Theo một số phương án, giá trị của góc màn trập (ví dụ, $\text{fixed_shutter_angle}$) có thể không là số nguyên, ví dụ, nó có thể là 135,75 độ. Để chính xác hơn, trong Bảng 21, có thể thay thế $u(9)$ (9 bit không dấu) với $u(16)$ hoặc một độ sâu bit phù hợp khác nào đó (ví dụ, 12 bit, 14 bit, hoặc nhiều hơn 16 bit).

Theo một số phương án, có thể có lợi nếu biểu diễn thông tin góc màn trập bằng “các nhịp đồng hồ.” Trong VVC, biến ClockTick được suy ra như sau:

$$\text{ClockTick} = \text{num_units_in_tick} \div \text{time_scale} \quad (8)$$

Sau đó, có thể biểu diễn cả thời khoảng khung và thời khoảng phơi sáng dưới dạng bội số hoặc phân số của các nhịp đồng hồ:

$$\text{exposure_duration} = \text{fN} * \text{ClockTick} \quad (9)$$

$$\text{frame_duration} = \text{fM} * \text{ClockTick} \quad (10)$$

trong đó fN và fM là các giá trị dấu phẩy động và $\text{fN} \leq \text{fM}$.

Sau đó

$$\begin{aligned} \text{shutter_angle} &= \text{frame_rate} * 360 * \text{shutter_speed} = \\ &= (1/\text{frame_duration}) * 360 * \text{exposure_duration} = \\ (11) \quad &= (\text{exposure_duration} * 360) / \text{frame_duration} = \\ &= (\text{fN} * \text{ClockTick} * 360) / (\text{fM} * \text{ClockTick}) = \\ &= (\text{fN} / \text{fM}) * 360 = (\text{Numerator} / \text{Denominator}) * 360, \end{aligned}$$

trong đó tử số và mẫu số là các số nguyên xấp xỉ tỷ lệ fN / fM .

Bảng 19 thể hiện ví dụ về bản tin SEI chỉ báo bởi phương trình (11). Theo ví dụ này, góc màn trập phải lớn hơn 0 đối với máy quay thể giới thực.

Bảng 19: Bản tin SEI làm ví dụ cho thông tin góc màn trập dựa vào các nhịp đồng hồ

shutter_angle_information (payloadSize) {	Mô tả
fixed_shutter_angle_within_cvs_flag	u(1)
if (fixed_shutter_angle_within_cvs_flag) {	
fixed_shutter_angle_numer_minus1	u(16)
fixed_shutter_angle_denom_minus1	u(16)
}	
else {	
for(i = 0; i <= sps_max_sub_layers_minus1; i++) {	
sub_layer_shutter_angle_numer_minus1[i]	u(16)
sub_layer_shutter_angle_denom_minus1[i]	u(16)
}	
}	

Như được thảo luận trên đây, việc sử dụng u(16) (16 bit không dấu) để đạt độ chính xác góc màn trập được thể hiện làm ví dụ và tương ứng với độ chính xác: $360/2^{16} = 0,0055$. Độ chính xác có thể được điều chỉnh dựa vào các ứng dụng thực tế. Ví dụ, sử dụng u(8), độ chính xác là $360/2^8 = 1,4063$.

Lưu ý – Góc màn trập được biểu diễn bằng độ lớn hơn 0 nhưng nhỏ hơn hoặc bằng 360 độ. Ví dụ, góc màn trập 180 độ chỉ báo thời khoảng phơi sáng là $\frac{1}{2}$ thời khoảng khung.

fixed_shutter_angle_within_cvs_flag bằng 1 chỉ ra giá trị góc màn trập là giống nhau với tất cả các lớp con thời gian trong CVS. **fixed_shutter_angle_within_cvs_flag** bằng 0 chỉ ra giá trị góc màn trập có thể không giống nhau với tất cả các lớp con thời gian trong CVS.

fixed_shutter_angle_numer_minus1 cộng 1 chỉ ra tử số được sử dụng để suy ra giá trị góc màn trập. Giá trị của **fixed_shutter_angle_numer_minus1** sẽ nằm trong khoảng từ 0 đến 65535, bao gồm giá trị đầu mút.

fixed_shutter_angle_demom_minus1 cộng 1 chỉ ra mẫu số được sử dụng để suy ra giá trị góc màn trập. Giá trị của **fixed_shutter_angle_demom_minus1** sẽ nằm trong khoảng từ 0 đến 65535, bao gồm giá trị đầu mút.

Giá trị của **fixed_shutter_angle_numer_minus1** sẽ nhỏ hơn hoặc bằng giá trị của **fixed_shutter_angle_demom_minus1**.

Biến **shutterAngle** theo độ được suy ra như sau:

$$\text{shutterAngle} = 360 * (\text{fixed_shutter_angle_numer_minus1} + 1) \div (\text{fixed_shutter_angle_demom_minus1} + 1)$$

sub_layer_shutter_angle_numer_minus1[i] cộng 1 chỉ ra tử số được sử dụng để suy ra giá trị góc màn trập khi **HighestTid** bằng **i**. Giá trị của **sub_layer_shutter_angle_numer_minus1[i]** sẽ nằm trong khoảng từ 0 đến 65535, bao gồm giá trị đầu mút.

sub_layer_shutter_angle_demom_minus1[i] cộng 1 chỉ ra mẫu số được sử dụng để suy ra giá trị góc màn trập khi **HighestTid** bằng **i**. Giá trị của **sub_layer_shutter_angle_demom_minus1[i]** sẽ nằm trong khoảng từ 0 đến 65535, bao gồm giá trị đầu mút.

Giá trị của **sub_layer_shutter_angle_numer_minus1[i]** sẽ nhỏ hơn hoặc bằng giá trị của **sub_layer_shutter_angle_denom_minus1[i]**.

Biến **subLayerShutterAngle[i]** theo độ được suy ra như sau:

$$\text{subLayerShutterAngle}[i] = 360 * (\text{sub_layer_shutter_angle_numer_minus1}[i] + 1) \div (\text{sub_layer_shutter_angle_demom_minus1}[i] + 1)$$

Theo một phương án khác, thời khoảng khung (ví dụ, **frame_duration**) có thể được chỉ ra bởi một số phương tiện khác. Ví dụ, trong DVB/ATSC, khi **fixed_pic_rate_within_cvs_flag[Tid]** bằng 1:

$$\begin{aligned} \text{frame_rate} &= \text{time_scale} / (\\ &\text{num_units_in_tick} * (\text{elemental_duration_in_tc_minus1}[\text{Tid}] + 1)), \\ \text{frame_duration} &= 1 / \text{frame_rate}. \end{aligned}$$

Cú pháp trong Bảng 19 và trong một số các bảng tiếp theo giả sử rằng góc màn trập sẽ luôn lớn hơn không; tuy nhiên, góc màn trập = 0 có thể được sử dụng để báo hiệu ý

tương sáng tạo trong đó nội dung sẽ được hiển thị mà không có bất kỳ độ nhòe chuyển động nào. Đây có thể là trường hợp đối với đồ họa chuyển động, hoạt hình, kết cấu CGI và màn chiếu, v.v. Do đó, ví dụ, việc báo hiệu góc màn trập = 0 có thể hữu ích để quyết định chế độ trong bộ chuyển mã (ví dụ, để lựa chọn các chế độ chuyển mã bảo toàn các cạnh) cũng như trong màn hình thu siêu dữ liệu góc màn trập qua giao diện CTA hoặc giao diện 3GPP. Ví dụ, góc màn trập = 0 có thể được sử dụng để chỉ báo cho màn hình rằng không nên thực hiện việc xử lý chuyển động bất kỳ như khử nhiễu, nội suy khung, và tương tự. Theo phương án này, các phần tử cú pháp **fixed_shutter_angle_numer_minus1** và **sub_layer_shutter_angle_numer_minus1[i]** có thể được thay thế bởi các phần tử cú pháp **fixed_shutter_angle_numer** và **sub_layer_shutter_angle_numer[i]**, trong đó **fixed_shutter_angle_numer** chỉ ra từ số được sử dụng để suy ra giá trị góc màn trập. Giá trị của **fixed_shutter_angle_numer** sẽ nằm trong khoảng từ 0 đến 65535, bao gồm giá trị đầu mút.

sub_layer_shutter_angle_numer[i] chỉ ra từ số được sử dụng để suy ra giá trị góc màn trập khi HighestTid bằng i. Giá trị của **sub_layer_shutter_angle_numer[i]** sẽ nằm trong khoảng từ 0 đến 65535, bao gồm giá trị đầu mút.

Theo một phương án khác, **fixed_shutter_angle_denom_minus1** và **sub_layer_shutter_angle_denom_minus1[i]** có thể còn được thay thế bởi các phần tử cú pháp **fixed_shutter_angle_denom** và **sub_layer_shutter_angle_denom[i]** như vậy.

Theo một phương án, như thể hiện trong Bảng 20, có thể dùng lại cú pháp **num_units_in_tick** và **time_scale** xác định trong SPS bằng cách đặt **general_hrd_parameters_present_flag** bằng 1 trong VVC. Theo kịch bản này, bản tin SEI có thể được đổi tên là bản tin SEI về thời khoảng phơi sáng.

Bảng 20: Bản tin SEI làm ví dụ để báo hiệu thời khoảng phơi sáng

exposure_duration_information (payloadSize) {	Mô tả
fixed_exposure_duration_within_cvs_flag	u(1)
if (fixed_shutter_angle_within_cvs_flag) {	
fixed_exposure_duration_numer_minus1	u(16)
fixed_exposure_duration_denom_minus1	u(16)
}	
else {	
for(i = 0; i <= sps_max_sub_layers_minus1; i++) {	
sub_layer_exposure_duration_numer_minus1[i]	u(16)
sub_layer_exposure_duration_denom_minus1[i]	u(16)
}	
}	

fixed_exposure_duration_within_cvs_flag bằng 1 chỉ ra giá trị thời khoảng phơi sáng hiệu quả là giống nhau với tất cả các lớp con thời gian trong CVS.

fixed_exposure_duration_within_cvs_flag bằng 0 chỉ ra giá trị thời khoảng phơi sáng hiệu quả có thể không giống nhau với tất cả các lớp con thời gian trong CVS.

fixed_exposure_duration_numer_minus1 cộng 1 chỉ ra tử số được sử dụng để suy ra giá trị thời khoảng phơi sáng. Giá trị của **fixed_exposure_duration_numer_minus1** sẽ nằm trong khoảng từ 0 đến 65535, bao gồm giá trị đầu mút.

fixed_exposure_duration_denom_minus1 cộng 1 chỉ ra mẫu số được sử dụng để suy ra giá trị thời khoảng phơi sáng. Giá trị của **fixed_exposure_duration_denom_minus1** sẽ nằm trong khoảng từ 0 đến 65535, bao gồm giá trị đầu mút.

Giá trị của **fixed_exposure_during_numer_minus1** sẽ nhỏ hơn hoặc bằng giá trị của **fixed_exposure_duration_denom_minus1**.

Biến **fixedExposureDuration** được suy ra như sau:

$$\text{fixedExposureDuration} = (\text{fixed_exposure_duration_numer_minus1} + 1) \div (\text{fixed_exposure_duration_denom_minus1} + 1) * \text{ClockTicks}$$

sub_layer_exposure_duration_numer_minus1[i] cộng 1 chỉ ra tử số được sử dụng để suy ra giá trị thời khoảng phơi sáng khi HighestTid bằng i. Giá trị của **sub_layer_exposure_duration_numer_minus1[i]** sẽ nằm trong khoảng từ 0 đến 65535, bao gồm giá trị đầu mút.

sub_layer_exposure_duration_demom_minus1[i] cộng 1 chỉ ra mẫu số được sử dụng để suy ra giá trị thời khoảng phơi sáng khi HighestTid bằng i. Giá trị của **sub_layer_exposure_duration_demom_minus1[i]** sẽ nằm trong khoảng từ 0 đến 65535, bao gồm giá trị đầu mút.

Giá trị của **sub_layer_exposure_duration_numer_minus1[i]** sẽ nhỏ hơn hoặc bằng giá trị của **sub_layer_exposure_duration_demom_minus1[i]**.

Biến **subLayerExposureDuration[i]** cho HighestTid bằng i được suy ra như sau:

$$\text{subLayerExposureDuration}[i] = (\text{sub_layer_exposure_duration_numer_minus1}[i] + 1) \div (\text{sub_layer_exposure_duration_demom_minus1}[i] + 1) * \text{ClockTicks}.$$

Theo một phương án khác, như được thể hiện trong Bảng 21, có thể xác định rõ ràng clockTick bởi các phân tử cú pháp **expo_num_units_in_tick** và **expo_time_scale**. Ưu điểm ở đây là ở chỗ nó không dựa vào việc tập hợp **general_hrd_parameters_present_flag** có được đặt bằng 1 trong VVC như phương án trước đó hay không, khi đó

$$\text{clockTick} = \text{expo_num_units_in_tick} \div \text{expo_time_scale} \quad (12)$$

Bảng 21: Bản tin SEI làm ví dụ để báo hiệu thời gian phơi sáng

exposure_duration_information (payloadSize) {	Mô tả
expo_num_units_in_tick	u(32)
expo_time_scale	u(32)
fixed_exposure_duration_within_cvs_flag	u(1)
if (!fixed_exposure_duration_within_cvs_flag)	
for(i = 0; i <= sps_max_sub_layers_minus1; i++) {	
sub_layer_exposure_duration_numer_minus1[i]	u(16)
sub_layer_exposure_duration_denom_minus1[i]	u(16)
}	
}	

expo_num_units_in_tick là số lượng đơn vị thời gian của đồng hồ hoạt động ở tần số **time_scale** Hz mà tương ứng với một lượng gia (được gọi là nhịp đồng hồ) của bộ đếm nhịp đồng hồ. **expo_num_units_in_tick** sẽ lớn hơn 0. Nhịp đồng hồ, xác định bởi biến **clockTick**, tính theo đơn vị giây, bằng thương số của **expo_num_units_in_tick** chia cho **expo_time_scale**.

expo_time_scale là số lượng đơn vị thời gian trôi qua trong một giây.

$\text{clockTick} = \text{expo_num_units_in_tick} \div \text{expo_time_scale}$.

LƯU Ý: Hai phần tử cú pháp: **expo_num_units_in_tick** và **expo_time_scale** được xác định để đo thời khoảng phơi sáng.

Yêu cầu về sự phù hợp dòng bit là **clockTick** sẽ nhỏ hơn hoặc bằng **ClockTick** khi **num_units_in_tick** và **time_scale** có mặt.

fixed_exposure_duration_within_cvs_flag bằng 1 chỉ ra giá trị thời khoảng phơi sáng hiệu quả là giống nhau với tất cả các lớp con thời gian trong CVS.

fixed_exposure_duration_within_cvs_flag bằng 0 chỉ ra giá trị thời khoảng phơi sáng hiệu quả có thể không giống nhau với tất cả các lớp con thời gian trong CVS. Khi

fixed_exposure_duration_within_cvs_flag bằng 1, biến **fixedExposureDuration** được đặt bằng **clockTick**.

$\text{sub_layer_exposure_duration_numer_minus1}[i]$ cộng 1 chỉ ra tử số được sử dụng để suy ra giá trị thời khoảng phơi sáng khi HighestTid bằng i . Giá trị của $\text{sub_layer_exposure_duration_numer_minus1}[i]$ sẽ nằm trong khoảng từ 0 đến 65535, bao gồm giá trị đầu mút.

$\text{sub_layer_exposure_duration_denom_minus1}[i]$ cộng 1 chỉ ra mẫu số được sử dụng để suy ra giá trị thời khoảng phơi sáng khi HighestTid bằng i . Giá trị của $\text{sub_layer_exposure_duration_denom_minus1}[i]$ sẽ nằm trong khoảng từ 0 đến 65535, bao gồm giá trị đầu mút.

Giá trị của $\text{sub_layer_exposure_duration_numer_minus1}[i]$ sẽ nhỏ hơn hoặc bằng giá trị của $\text{sub_layer_exposure_duration_denom_minus1}[i]$.

Biến $\text{subLayerExposureDuration}[i]$ cho HighestTid bằng i được suy ra như sau:

$$\text{subLayerExposureDuration}[i] = (\text{sub_layer_exposure_duration_numer_minus1}[i] + 1) \div (\text{sub_layer_exposure_duration_denom_minus1}[i] + 1) * \text{clockTick}.$$

Như được thảo luận trên đây, các tham số cú pháp

$\text{sub_layer_exposure_duration_numer_minus1}[i]$ và

$\text{sub_layer_exposure_duration_denom_minus1}[i]$ có thể còn được thay thế bởi

$\text{sub_layer_exposure_duration_numer}[i]$ và $\text{sub_layer_exposure_duration_denom}[i]$.

Theo một phương án khác, như được thể hiện trong Bảng 22, có thể xác định tham số ShutterInterval (tức là, thời khoảng phơi sáng) bằng các phần tử cú pháp

$\text{sii_num_units_in_shutter_interval}$ và sii_time_scale , trong đó

$$\text{ShutterInterval} = \text{sii_num_units_in_shutter_interval} \div \text{sii_time_scale}.$$

(13)

Bảng 22: Bản tin SEI làm ví dụ cho việc báo hiệu thời khoảng phơi sáng (thông tin khoảng thời gian màn trập)

shutter_interval_information (payloadSize) {	Mô tả
sii_num_units_in_shutter_interval	u(32)
sii_time_scale	u(32)
fixed_shutter_interval_within_cvs_flag	u(1)
if (!fixed_shutter_interval_within_cvs_flag)	
for(i = 0; i <= sps_max_sub_layers_minus1; i++) {	
sub_layer_shutter_interval_numer[i]	u(16)
sub_layer_shutter_interval_denom[i]	u(16)
}	

Các ngữ nghĩa của bản tin SEI thông tin khoảng thời gian màn trập

Bản tin SEI thông tin khoảng thời gian màn trập chỉ báo khoảng thời gian màn trập cho nội dung video liên quan trước khi mã hóa và hiển thị – ví dụ, cho nội dung được ghi lại bằng máy quay, lượng thời gian mà bộ cảm biến ảnh được phơi sáng để tạo ra hình ảnh.

sii_num_units_in_shutter_interval chỉ ra số lượng đơn vị thời gian của đồng hồ hoạt động ở tần số **sii_time_scale** Hz mà tương ứng với một lượng gia của bộ đếm nhịp đồng hồ màn trập. Khoảng thời gian màn trập, được xác định bởi biến ShutterInterval, tính theo đơn vị giây, bằng thương số của **sii_num_units_in_shutter_interval** chia cho **sii_time_scale**. Ví dụ, khi ShutterInterval bằng 0,04 giây, **sii_time_scale** có thể bằng 27 000 000 và **sii_num_units_in_shutter_interval** có thể bằng 1 080 000.

sii_time_scale chỉ ra số lượng đơn vị thời gian trôi qua trong một giây. Ví dụ, hệ tọa độ thời gian mà đo thời gian sử dụng đồng hồ 27MHz có **sii_time_scale** bằng 27 000 000.

Khi giá trị của **sii_time_scale** lớn hơn 0, giá trị của ShutterInterval được chỉ ra bởi:

$$\text{ShutterInterval} = \text{sii_num_units_in_shutter_interval} \div \text{sii_time_scale}$$

Nếu không (giá trị của **sii_time_scale** bằng 0), thì ShutterInterval nên được hiểu là ẩn số hoặc không được chỉ ra.

LƯU Ý 1 – Giá trị của ShutterInterval bằng 0 có thể chỉ báo nội dung video liên quan chứa nội dung ghi bằng màn hình, nội dung được tạo ra bằng máy tính, hoặc nội dung ghi không phải bằng máy quay khác.

LƯU Ý 2 – Giá trị của ShutterInterval lớn hơn giá trị của nghịch đảo của tốc độ hình ảnh được mã hóa, khoảng thời gian hình ảnh được mã hóa, có thể chỉ báo tốc độ hình ảnh được mã hóa lớn hơn tốc độ hình ảnh mà nội dung video liên quan được tạo ra – ví dụ, khi tốc độ hình ảnh được mã hóa là 120 Hz và tốc độ hình ảnh của nội dung video liên quan trước khi mã hóa và hiển thị là 60 Hz. Khoảng thời gian mã hóa cho lớp con thời gian cho trước Tid có thể được chỉ báo bởi ClockTick và $\text{elemental_duration_in_tc_minus1}[Tid]$. Ví dụ, khi

$\text{fixed_pic_rate_within_cvs_flag}[Tid]$ bằng 1, khoảng thời gian hình ảnh cho lớp con thời gian cho trước Tid, được xác định bởi biến $\text{PictureInterval}[Tid]$, có thể được chỉ ra bởi:

$$\text{PictureInterval}[Tid] = \text{ClockTick} * (\text{elemental_duration_in_tc_minus1}[Tid] + 1).$$

$\text{fixed_shutter_interval_within_cvs_flag}$ bằng 1 chỉ ra giá trị của ShutterInterval là giống nhau với tất cả các lớp con thời gian trong CVS. $\text{fixed_shutter_interval_within_cvs_flag}$ bằng 0 chỉ ra giá trị của ShutterInterval có thể không giống nhau với tất cả các lớp con thời gian trong CVS.

$\text{sub_layer_shutter_interval_numer}[i]$ chỉ ra tử số được sử dụng để suy ra khoảng thời gian màn trập lớp con, được xác định bởi biến $\text{subLayerShutterInterval}[i]$, tính theo đơn vị giây, khi HighestTid bằng i.

$\text{sub_layer_shutter_interval_denom}[i]$ chỉ ra mẫu số được sử dụng để suy ra khoảng thời gian màn trập lớp con, được xác định bởi biến $\text{subLayerShutterInterval}[i]$, tính theo đơn vị giây, khi HighestTid bằng i.

Giá trị của $\text{subLayerShutterInterval}[i]$ cho HighestTid bằng i được suy ra như sau. Khi giá trị của $\text{fixed_shutter_interval_within_cvs_flag}$ bằng 0 và giá trị của $\text{sub_layer_shutter_interval_denom}[i]$ lớn hơn 0:

$$\text{subLayerShutterInterval}[i] = \text{ShutterInterval} * \text{sub_layer_shutter_interval_numer}[i] \div \text{sub_layer_shutter_interval_denom}[i]$$

Nếu không (giá trị của `sub_layer_shutter_interval_denom[ i ]` bằng 0), thì `subLayerShutterInterval[ i ]` nên được hiểu là ẩn số hoặc không được chỉ ra. Khi giá trị của `fixed_shutter_interval_within_cvs_flag` không bằng 0,

`subLayerShutterInterval[ i ] = ShutterInterval`.

Theo phương án khác, thay vì sử dụng tử số và mẫu số để báo hiệu khoảng thời gian màn trập lớp con, người có hiểu biết trung bình trong lĩnh vực sử dụng một giá trị. Ví dụ về cú pháp như vậy được thể hiện trong Bảng 23.

Bảng 23: Bản tin SEI làm ví dụ để báo hiệu khoảng thời gian màn trập

<code>shutter_interval_information (payloadSize) {</code>	Mô tả
<code> sii_num_units_in_shutter_interval</code>	u(32)
<code> sii_time_scale</code>	u(32)
<code> fixed_shutter_interval_within_cvs_flag</code>	u(1)
<code> if (!fixed_shutter_interval_within_cvs_flag)</code>	
<code> for(i = 0; i <= sps_max_sub_layers_minus1; i++) {</code>	
<code> sub_layer_num_units_in_shutter_interval[i]</code>	u(32)
<code> }</code>	
<code>}</code>	

Các ngữ nghĩa của bản tin SEI thông tin khoảng thời gian màn trập

Bản tin SEI thông tin khoảng thời gian màn trập chỉ báo khoảng thời gian màn trập cho nội dung video liên quan trước khi mã hóa và hiển thị – ví dụ, cho nội dung được ghi lại bằng máy quay, lượng thời gian mà bộ cảm biến ảnh được phơi sáng để tạo ra hình ảnh.

`sii_num_units_in_shutter` chỉ ra số lượng đơn vị thời gian của đồng hồ hoạt động ở tần số

`sii_time_scale` Hz mà tương ứng với một lượng gia của bộ đếm nhịp đồng hồ màn trập.

Khoảng thời gian màn trập, được xác định bởi biến `ShutterInterval`, tính theo đơn vị giây, bằng thương số của `sii_num_units_in_shutter_interval` chia cho `sii_time_scale`. Ví dụ, khi `ShutterInterval` bằng 0,04 giây, `sii_time_scale` có thể bằng 27 000 000 và `sii_num_units_in_shutter_interval` có thể bằng 1 080 000.

`sii_time_scale` chỉ ra số lượng đơn vị thời gian trôi qua trong một giây. Ví dụ, hệ tọa độ thời gian mà đo thời gian sử dụng đồng hồ 27MHz có `sii_time_scale` bằng 27 000 000.

Khi giá trị của `sii_time_scale` lớn hơn 0, giá trị của `ShutterInterval` được chỉ ra bởi:

$$\text{ShutterInterval} = \text{sii_num_units_in_shutter_interval} \div \text{sii_time_scale}$$

Nếu không (giá trị của `sii_time_scale` bằng 0), thì `ShutterInterval` nên được hiểu là âm số hoặc không được chỉ ra.

LƯU Ý 1 – Giá trị của `ShutterInterval` bằng 0 có thể chỉ báo nội dung video liên quan chứa nội dung ghi bằng màn hình, nội dung được tạo ra bằng máy tính, hoặc nội dung ghi không phải bằng máy quay khác.

LƯU Ý 2 – Giá trị của `ShutterInterval` lớn hơn giá trị của nghịch đảo của tốc độ hình ảnh được mã hóa, khoảng thời gian hình ảnh được mã hóa, có thể chỉ báo tốc độ hình ảnh được mã hóa lớn hơn tốc độ hình ảnh mà nội dung video liên quan được tạo ra – ví dụ, khi tốc độ hình ảnh được mã hóa là 120 Hz và tốc độ hình ảnh của nội dung video liên quan trước khi mã hóa và hiển thị là 60 Hz. Khoảng thời gian hình ảnh được mã hóa cho `Tid` lớp con thời gian cho trước có thể được chỉ báo bởi `ClockTick` và `elemental_duration_in_tc_minus1[ Tid ]`. Ví dụ, khi

`fixed_pic_rate_within_cvs_flag[ Tid ]` bằng 1, khoảng thời gian hình ảnh cho lớp con thời gian cho trước `Tid`, được xác định bởi biến `PictureInterval[ Tid ]`, có thể được chỉ ra bởi:

$$\text{PictureInterval}[\text{Tid}] = \text{ClockTick} * (\text{elemental_duration_in_tc_minus1}[\text{Tid}] + 1) .$$

fixed_shutter_interval_within_cvs_flag bằng 1 chỉ ra giá trị của `ShutterInterval` là giống nhau với tất cả các lớp con thời gian trong CVS. **fixed_shutter_interval_within_cvs_flag** bằng 0 chỉ ra giá trị của `ShutterInterval` có thể không giống nhau với tất cả các lớp con thời gian trong CVS.

sub_layer_num_units_in_shutter_interval[i] chỉ ra số lượng đơn vị thời gian của đồng hồ hoạt động ở tần số `sii_time_scale` Hz mà tương ứng với một lượng gia của bộ đếm nhịp đồng hồ màn trập. Khoảng thời gian màn trập lớp con, được xác định bởi biến `subLayerShutterInterval[ i ]`, tính theo đơn vị giây, khi `HighestTid` bằng `i`, bằng thương số của `sub_layer_num_units_in_shutter_interval[ i ]` chia cho `sii_time_scale`.

Khi giá trị của **fixed_shutter_interval_within_cvs_flag** bằng 0 và giá trị của `sii_time_scale` lớn hơn 0, giá trị của `subLayerShutterInterval[ i ]` được chỉ ra bởi:

$$\text{subLayerShutterInterval}[i] = \frac{\text{sub_layer_num_units_in_shutter_interval}[i]}{\text{sii_time_scale}}$$

Nếu không (giá trị của `sii_time_scale` bằng 0), thì `subLayerShutterInterval[ i ]` nên được hiểu là ẩn số hoặc không được chỉ ra. Khi giá trị của `fixed_shutter_interval_within_cvs_flag` không bằng 0,

`subLayerShutterInterval[ i ] = ShutterInterval.`

Bảng 24 cung cấp bản tóm tắt của sáu cách tiếp cận được thảo luận trong các Bảng 18 đến 23 để cung cấp bản tin SEI liên quan đến góc màn trập hoặc thời khoảng phơi sáng.

Bảng 24: Tóm tắt về các cách tiếp cận gửi bản tin SEI để báo hiệu thông tin góc màn trập của tín hiệu

Bảng số	Các phần tử báo hiệu chính và phần tử phụ thuộc
18.	Góc màn trập (0 đến 360) được báo hiệu rõ ràng
19.	Góc màn trập được biểu diễn dưới dạng tỷ lệ của giá trị từ số và mẫu số được định tỷ lệ bởi 360 (giá trị nhịp đồng hồ là ngầm định)
20.	Thời khoảng phơi sáng được báo hiệu dưới dạng tỷ lệ của giá trị từ số và mẫu số (giá trị nhịp đồng hồ là ngầm định)
21.	Thời khoảng phơi sáng được báo hiệu dưới dạng tỷ lệ của giá trị từ số và mẫu số; giá trị nhịp đồng hồ được báo hiệu rõ ràng dưới dạng tỷ số của hai giá trị
22.	Thông tin khoảng thời gian màn trập được báo hiệu dưới dạng tỷ lệ của hai giá trị: lượng đơn vị nhịp đồng hồ trong sự phơi sáng và tỷ lệ thời gian phơi sáng; các lần phơi sáng liên quan đến lớp con được báo hiệu dưới dạng tỷ lệ của hai giá trị
23.	Thông tin khoảng thời gian màn trập hoặc thời khoảng phơi sáng được báo hiệu dưới dạng tỷ lệ của hai giá trị: lượng đơn vị nhịp đồng hồ trong sự phơi sáng và tỷ lệ phơi sáng theo thời gian; các lần phơi sáng liên quan đến lớp con được báo hiệu bằng số lượng đơn vị thời gian nhịp đồng hồ trong sự phơi sáng trong mỗi lớp con

Báo hiệu tốc độ khung biến đổi

Như được mô tả trong đơn tạm thời Mỹ số 62/883,195 nộp ngày 16/08/2019, trong nhiều đơn, mong muốn bộ giải mã hỗ trợ phát lại ở tốc độ khung biến đổi. Sự thích ứng tốc

độ khung thường là một phần của các hoạt động trong bộ giải mã tham chiếu giả định (HRD), như được mô tả, ví dụ, trong Phụ lục C của Ref. [2]. Theo một phương án, đề xuất báo hiệu, thông qua bản tin SEI hoặc phương tiện khác, một phần tử cú pháp xác định thời gian trình diễn hình ảnh (picture presentation time - PPT) như là chức năng của đồng hồ 90 kHz. Đây là kiểu lặp của thời gian đầu ra của bộ đệm hình ảnh giải mã danh nghĩa (decoder picture buffer - DPB) như được chỉ ra trong HRD, nhưng hiện nay sử dụng độ chính xác ClockTick 90 kHz như được chỉ ra trong hệ thống MPEG 2. Lợi ích của bản tin SEI này là a) nếu HRD không được kích hoạt, thì người có hiểu biết trung bình có thể vẫn sử dụng bản tin SEI PPT để chỉ báo sự định thời cho mỗi khung; b) nó có thể hỗ trợ sự tịnh tiến của định thời dòng bit và định thời hệ thống.

Bảng 25 mô tả ví dụ về cú pháp của bản tin định thời PPT được đề xuất, khớp với cú pháp của biến dấu thời gian trình diễn (presentation time stamp - PTS) được sử dụng trong truyền tải MPEG-2 (H.222) (Ref.[4]).

Bảng 25: Ví dụ về cú pháp cho bản tin thời gian trình diễn hình ảnh

picture_presentation_time (payloadSize) {	Mô tả
PPT	u(33)
}	

PPT (thời gian trình diễn hình ảnh)

– Thời gian trình diễn sẽ liên quan đến thời gian giải mã như sau: PPT là số 33 bit được mã hóa trong ba trường riêng biệt. Nó chỉ báo thời gian trình diễn, $tp_n(k)$, trong bộ giải mã đích hệ thống của đơn vị trình diễn k của dòng cơ sở n . Giá trị của PPT được chỉ rõ theo các đơn vị của chu kỳ về tần số đồng hồ hệ thống chia cho 300 (tạo ra 90 kHz). Thời gian trình diễn hình ảnh được suy ra từ PPT theo phương trình dưới đây

$$PPT(k) = ((system_clock_frequency \times tp_n(k))/300) \% 2^{33}$$

trong đó $tp_n(k)$ là thời gian trình diễn của đơn vị trình diễn $P_n(k)$.

Tài liệu tham khảo

Mỗi trong số các tài liệu tham khảo được liệt kê ở đây được đưa vào đây bằng cách viện dẫn toàn bộ.

- [1] *High efficiency video coding*, H.265, Series H, Coding of moving video, ITU, (02/2018).
- [2] B. Bross, J. Chen, and S. Liu, “*Versatile Video Coding (Draft 5)*,” JVET output document, JVET-N1001, v5, uploaded May 14, 2019.
- [3] C. Carbonara, J. DeFilippis, M. Korpi, “*High Frame Rate Capture and Production*,” SMPTE 2015 Annual Technical Conference and Exhibition, Oct 26-29, 2015.
- [4] *Infrastructure of audiovisual services – Transmission multiplexing and synchronization*, H.222.0, Series H, Generic coding of moving pictures and associated audio information: Systems, ITU, 08/2018.

VÍ DỤ VỀ PHƯƠNG ÁN THỰC HIỆN HỆ THỐNG MÁY TÍNH

Các phương án của sáng chế có thể được thực thi bằng hệ thống máy tính, các hệ thống được tạo cấu hình trong mạch và thành phần điện tử, thiết bị mạch tích hợp (integrated circuit - IC) chẳng hạn như bộ vi điều khiển, mảng công lập trình được theo trường (field programmable gate array - FPGA), hoặc thiết bị logic có thể tạo cấu hình hoặc có thể lập trình được (programmable logic device - PLD) khác, bộ xử lý thời gian rời rạc hoặc bộ xử lý tín hiệu số (digital signal processor - DSP), mạch tích hợp chuyên dụng (application specific integrated circuit - ASIC), và/hoặc thiết bị mà bao gồm một hoặc nhiều trong số các hệ thống, thiết bị hoặc thành phần như vậy. Máy tính và/hoặc IC có thể thực hiện, điều khiển, hoặc thực thi các lệnh liên quan đến khả năng định tỷ lệ tốc độ khung, như các lệnh được mô tả trong bản mô tả này. Máy tính và/hoặc IC có thể tính toán bất kỳ trong số nhiều tham số hoặc giá trị liên quan đến khả năng định tỷ lệ tốc độ khung như được mô tả trong sáng chế. Các phương án về ảnh hoặc video có thể được triển khai trong phần cứng, phần mềm, firmware hoặc các tổ hợp khác nhau của chúng.

Một số phương án thực hiện của sáng chế bao gồm các bộ xử lý máy tính thực thi các lệnh phần mềm khiến cho các bộ xử lý thực hiện phương pháp của sáng chế. Ví dụ, một hoặc nhiều bộ xử lý trong màn hình, bộ mã hóa, hộp giải mã tín hiệu, bộ chuyển mã hoặc trung tự có thể thực thi các phương pháp liên quan đến khả năng định tỷ lệ tốc độ khung như được mô tả trên đây bằng cách thực thi các lệnh phần mềm trong bộ nhớ chương trình truy cập được bởi các bộ xử lý. Các phương án của sáng chế cũng có thể được đề xuất dưới dạng sản phẩm chương trình. Sản phẩm chương trình có thể bao gồm phương tiện bất biến và hữu hình bất kỳ mang tập hợp tín hiệu đọc được bằng máy tính chứa các lệnh mà, khi được thực

thi bởi bộ xử lý dữ liệu, khiến cho bộ xử lý dữ liệu thực thi phương pháp theo sáng chế. Các sản phẩm chương trình theo sáng chế có thể có dạng bất kỳ trong số các dạng bất biến hoặc hữu hình. Sản phẩm chương trình có thể bao gồm, ví dụ, phương tiện vật lý chẳng hạn như phương tiện lưu trữ dữ liệu từ tính bao gồm đĩa mềm, ổ đĩa cứng, phương tiện lưu trữ dữ liệu quang bao gồm CD ROM, DVD, phương tiện lưu trữ dữ liệu điện tử bao gồm ROM, RAM nhanh (flash), hoặc tương tự. Tín hiệu đọc được bằng máy tính trên sản phẩm chương trình có thể được nén hoặc mã hóa một cách tùy ý.

Trường hợp bộ phận (ví dụ, modul phần mềm, bộ xử lý, bộ, thiết bị, mạch, v.v.) được đề cập ở trên, thì trừ khi có quy định khác, tham chiếu đến thành phần đó (bao gồm tham chiếu đến “phương tiện”) nên được hiểu là bao gồm dưới dạng các tương đương của thành phần đó thành phần bất kỳ mà thực hiện chức năng của thành phần được mô tả (ví dụ, thành phần tương đương về mặt chức năng), bao gồm cả các thành phần không tương đương về mặt cấu trúc với cấu trúc được bộc lộ mà thực hiện chức năng theo các phương án làm ví dụ được minh họa của sáng chế.

CÁC PHƯƠNG ÁN TƯƠNG ĐƯƠNG, MỞ RỘNG, THAY THẾ VÀ CÁC PHƯƠNG ÁN KHÁC

Các phương án làm ví dụ liên quan đến khả năng định tỷ lệ tốc độ khung do đó được mô tả. Trong phần mô tả ở trên, các phương án theo sáng chế đã được mô tả dựa vào nhiều chi tiết cụ thể mà có thể thay đổi từ phương án thực hiện này đến phương án thực hiện khác. Do đó, phần duy nhất và riêng thể hiện cái gì là sáng chế, và được người nộp đơn dự định là là sáng chế, là bộ yêu cầu bảo hộ nộp kèm theo đơn này, dưới hình thức cụ thể mà ở đó các điểm yêu cầu bảo hộ được thể hiện, bao gồm cả sự chỉnh sửa sau đó bất kỳ. Mọi định nghĩa được nêu rõ ràng ở đây về các thuật ngữ được chứa trong các điểm yêu cầu bảo hộ sẽ điều chỉnh nghĩa của các thuật ngữ này như được sử dụng trong các điểm yêu cầu bảo hộ. Do đó, trong mọi trường hợp, không sự giới hạn, phần tử, tính chất, dấu hiệu, ưu điểm hoặc thuộc tính nào mà không được thể hiện rõ ràng trong yêu cầu bảo hộ làm giới hạn phạm vi của điểm yêu cầu bảo hộ đó. Do đó, bản mô tả sáng chế và các hình vẽ phải được hiểu theo nghĩa có tính chất minh họa chứ không phải theo nghĩa có tính chất thu hẹp phạm vi bảo hộ sáng chế.

Phụ lục

Phụ lục này cung cấp bản sao của Bảng D.2 và thông tin về pic_struct liên quan từ bản đặc tả kỹ thuật H.265 (Ref. [1]).

Bảng D.2 – Diễn giải về pic_struct

Giá trị	Màn hiển thị hình ảnh được chỉ báo	Các giới hạn
0	Khung (tăng dần)	field_seq_flag sẽ bằng 0
1	Trường trên cùng	field_seq_flag sẽ bằng 1
2	Trường dưới cùng	field_seq_flag sẽ bằng 1
3	Trường trên cùng, trường dưới cùng, theo thứ tự đó	field_seq_flag sẽ bằng 0
4	Trường dưới cùng, trường trên cùng, theo thứ tự đó	field_seq_flag sẽ bằng 0
5	Trường trên cùng, trường dưới cùng, trường trên cùng được lặp lại, theo thứ tự đó	field_seq_flag sẽ bằng 0
6	Trường dưới cùng, trường trên cùng, trường dưới cùng được lặp lại, theo thứ tự đó	field_seq_flag sẽ bằng 0
7	Gấp đôi khung	field_seq_flag sẽ bằng 0 fixed_pic_rate_within_cvs_flag sẽ bằng 1
8	Gấp ba khung	field_seq_flag sẽ bằng 0 fixed_pic_rate_within_cvs_flag sẽ bằng 1
9	Trường trên cùng ghép cặp với trường dưới cùng trước đó theo thứ tự đầu ra	field_seq_flag sẽ bằng 1

10	Trường dưới cùng ghép cặp với trường trên cùng trước đó theo thứ tự đầu ra	field_seq_flag sẽ bằng 1
11	Trường trên cùng ghép cặp với trường dưới cùng tiếp theo theo thứ tự đầu ra	field_seq_flag sẽ bằng 1
12	Trường dưới cùng ghép cặp với trường trên cùng tiếp theo theo thứ tự đầu ra	field_seq_flag sẽ bằng 1

Ngữ nghĩa của phần tử cú pháp pic_struct

pic_struct chỉ báo liệu hình nên được hiển thị dưới dạng khung hay dưới dạng một hoặc nhiều trường và, đối với hiển thị các khung khi fixed_pic_rate_within_cvs_flag bằng 1, có thể chỉ báo chu kỳ lặp gấp đôi hoặc gấp ba khung cho các màn hình mà sử dụng khoảng thời gian làm mới khung cố định bằng DpbOutputElementalInterval[n] như được cho bởi Phương trình E-73. Diễn giải về pic_struct được chỉ ra trong Bảng D.2. Các giá trị của pic_struct mà không được liệt kê trong Bảng D.2 được dự trữ cho việc sử dụng trong tương lai bởi ITU-T | ISO/IEC và sẽ không có mặt trong dòng bit tuân theo phiên bản của bản đặc tả này. Các bộ giải mã sẽ bỏ qua các giá trị dự trữ của pic_struct.

Khi có mặt, yêu cầu về sự phù hợp dòng bit là giá trị của pic_struct sẽ bị ràng buộc sao cho chỉ một trong số các điều kiện sau đây là đúng:

- Giá trị của pic_struct bằng 0, 7 hoặc 8 với tất cả các hình ảnh trong CVS.
- Giá trị của pic_struct bằng 1, 2, 9, 10, 11 hoặc 12 với tất cả các hình ảnh trong CVS.
- Giá trị của pic_struct bằng 3, 4, 5 hoặc 6 với tất cả các hình ảnh trong CVS.

Khi fixed_pic_rate_within_cvs_flag bằng 1, kỹ thuật gấp đôi khung được chỉ báo bởi pic_struct bằng 7, chỉ báo rằng khung nên được hiển thị hai lần liên tiếp trên màn hình với khoảng thời gian làm mới khung bằng DpbOutputElementalInterval[n] như được cho bởi Phương trình E-73, và kỹ thuật gấp ba khung được chỉ báo bởi pic_struct bằng 3, chỉ báo rằng khung nên được hiển thị ba lần liên tiếp trên các màn hình với khoảng thời gian làm mới khung bằng DpbOutputElementalInterval[n] như được cho bởi Phương trình E-73.

LƯU Ý 3 – Kỹ thuật gấp đôi khung có thể được sử dụng để tạo thuận lợi cho việc hiển thị, ví dụ, video quét tăng dần 25 Hz trên màn hình quét tăng dần 50 Hz hoặc video quét tăng dần 30 Hz trên màn hình quét tăng dần 60 Hz. Sử dụng kỹ thuật gấp đôi khung và gấp ba khung trong sự kết hợp đan xen trên từng khung khác có thể được sử dụng để tạo thuận lợi cho việc hiển thị video quét tăng dần 24 Hz trên màn hình quét tăng dần 60 Hz.

Các vị trí lấy mẫu dọc và ngang danh nghĩa của các mẫu trong các trường trên cùng và dưới cùng cho các định dạng màu 4:2:0, 4:2:2 và 4:4:4 được thể hiện lần lượt trên Fig.D.1, Fig.D.2, và Fig.D.3.

Các bộ chỉ báo kết hợp cho các trường (pic_struct bằng 9 đến 12) cung cấp các gợi ý để kết hợp các trường có tính chẵn lẻ bổ sung cùng nhau làm các khung. Tính chẵn lẻ của trường có thể là ở trên cùng hoặc dưới cùng và tính chẵn lẻ của hai trường được coi là bổ sung khi tính chẵn lẻ của một trường là ở trên cùng và tính chẵn lẻ của trường khác là ở dưới cùng.

Khi frame_field_info_present_flag bằng 1, yêu cầu về sự phù hợp dòng bit là các ràng buộc được chỉ ra trong cột thứ ba của Bảng D.2 sẽ áp dụng.

LƯU Ý 4 – Khi frame_field_info_present_flag bằng 0, thì trong nhiều trường hợp các giá trị mặc định có thể được suy ra hoặc được chỉ báo bằng phương tiện khác. Khi không có các chỉ báo khác về loại hiển thị hình ảnh được dự định, bộ giải mã sẽ suy ra giá trị của pic_struct là bằng 0 khi frame_field_info_present_flag bằng 0.

YÊU CẦU BẢO HỘ

1. Phương tiện bất biến đọc được bằng bộ xử lý lưu trữ trên đó cấu trúc dòng video mã hóa, cấu trúc dòng video mã hóa bao gồm:

phần hình ảnh mã hóa bao gồm việc mã hóa chuỗi hình ảnh video; và

phần báo hiệu bao gồm việc mã hóa:

cờ góc màn trập thứ nhất chỉ báo liệu thông tin góc màn trập có cố định cho tất cả các lớp con thời gian trong phần hình ảnh mã hóa hay không; và

nếu cờ góc màn trập thứ nhất chỉ báo rằng thông tin góc màn trập là cố định, thì phần báo hiệu bao gồm giá trị góc màn trập cố định để hiển thị phiên bản giải mã của chuỗi hình ảnh video cho tất cả các lớp con thời gian trong phần hình ảnh được mã hóa bằng cách sử dụng giá trị góc màn trập cố định, nếu không thì

phần báo hiệu bao gồm mảng giá trị góc màn trập lớp con, trong đó đối với mỗi lớp con thời gian trong số các lớp con thời gian, giá trị trong mảng giá trị góc màn trập lớp con chỉ báo góc màn trập tương ứng để hiển thị phiên bản giải mã của lớp con thời gian của chuỗi hình ảnh video.

2. Phương tiện bất biến đọc được bằng bộ xử lý theo điểm 1, trong đó phần báo hiệu còn bao gồm mã hóa giá trị lặp khung chỉ báo số lần mà hình giải mã trong chuỗi hình ảnh video nên được hiển thị liên tiếp.

3. Phương tiện bất biến đọc được bằng bộ xử lý theo điểm 1, trong đó phần báo hiệu còn bao gồm mã hóa giá trị `num_units_in_tick` chỉ báo số lượng đơn vị thời gian của đồng hồ hoạt động ở tần số có giá trị `time_scale` chỉ báo số lượng đơn vị thời gian trôi qua trong một giây, và mã hóa giá trị `time_scale`, trong đó các giá trị `num_units_in_tick` và `time_scale` được sử dụng bởi bộ giải mã để điều chỉnh sự lặp khung và/hoặc thời khoảng trong khi phát lại trên cơ sở từng khung.

4. Phương tiện bất biến đọc được bằng bộ xử lý theo điểm 1, trong đó phần báo hiệu bao gồm phần bản tin thông tin nâng cao bổ sung (supplemental enhancement information - SEI) hoặc phần bản tin thông tin người dùng video (video user information - VUI).

5. Phương tiện bất biến đọc được bằng bộ xử lý lưu trữ trên đó cấu trúc dòng video mã hóa, cấu trúc dòng video mã hóa bao gồm:

phần hình ảnh mã hóa bao gồm việc mã hóa chuỗi hình ảnh video; và

phần báo hiệu bao gồm việc mã hóa:

cờ góc màn trập thứ nhất chỉ báo liệu thông tin góc màn trập có cố định cho tất cả các lớp con thời gian trong phần hình ảnh mã hóa hay không; và

nếu cờ góc màn trập thứ nhất chỉ báo rằng thông tin góc màn trập là cố định, thì phần báo hiệu bao gồm giá trị tử số của góc màn trập cố định và giá trị mẫu số của góc màn trập cố định để hiển thị phiên bản giải mã của chuỗi hình ảnh video cho tất cả các lớp con thời gian trong phần hình ảnh mã hóa bằng cách sử dụng góc màn trập cố định dựa vào giá trị tử số của góc màn trập cố định và giá trị mẫu số của góc màn trập cố định, nếu không thì

phần báo hiệu bao gồm mảng của các giá trị tử số của góc màn trập lớp con và mảng của các giá trị mẫu số của góc màn trập lớp con, trong đó mỗi trong số các giá trị tử số của góc màn trập lớp con và các giá trị mẫu số của góc màn trập lớp con được áp dụng để tính toán góc màn trập lớp con tương ứng để hiển thị phiên bản giải mã của lớp con thời gian của chuỗi hình ảnh video.

6. Phương tiện bất biến đọc được bằng bộ xử lý lưu trữ trên đó cấu trúc dòng video mã hóa, cấu trúc dòng video mã hóa bao gồm:

phần hình ảnh mã hóa bao gồm việc mã hóa chuỗi hình ảnh video; và

phần báo hiệu bao gồm việc mã hóa:

cờ thời khoảng phơi sáng thứ nhất chỉ báo liệu thông tin thời khoảng phơi sáng có cố định cho tất cả các lớp con thời gian trong phần hình ảnh mã hóa hay không; và

nếu cờ thời khoảng phơi sáng thứ nhất chỉ báo rằng thông tin thời khoảng phơi sáng là cố định, thì phần báo hiệu bao gồm giá trị tử số của thời khoảng phơi sáng cố định và giá trị mẫu số của thời khoảng phơi sáng cố định để hiển thị phiên bản giải mã của chuỗi hình ảnh video cho tất cả các lớp con thời gian trong phần hình ảnh mã hóa bằng cách sử dụng thời khoảng phơi sáng cố định dựa vào giá trị tử số của thời khoảng phơi sáng cố định và giá trị mẫu số của thời khoảng phơi sáng cố định, nếu không thì

phần báo hiệu bao gồm mảng các giá trị tử số của thời khoảng phơi sáng lớp con và mảng các giá trị mẫu số của thời khoảng phơi sáng lớp con, trong đó mỗi trong số các giá trị tử số của thời khoảng phơi sáng lớp con và các giá trị mẫu số của thời khoảng phơi sáng lớp con được áp dụng để tính toán thời khoảng phơi sáng lớp con tương ứng để hiển thị phiên bản giải mã của lớp con thời gian của chuỗi hình ảnh video.

7. Phương tiện bất biến đọc được bằng bộ xử lý theo điểm 6, trong đó phần báo hiệu còn bao gồm các tham số báo hiệu để xác định giá trị nhịp đồng hồ.

8. Phương tiện bất biến đọc được bằng bộ xử lý theo điểm 7, trong đó các tham số báo hiệu để xác định giá trị nhịp đồng hồ bao gồm tham số nhịp đồng hồ thứ nhất biểu thị thời gian phơi sáng bằng số lượng đơn vị thời gian của đồng hồ và tham số nhịp đồng hồ thứ hai biểu thị thang đo thời gian phơi sáng của số lượng đơn vị thời gian trong một giây, và giá trị nhịp đồng hồ được xác định là tỷ lệ của tham số nhịp đồng hồ thứ nhất và tham số nhịp đồng hồ thứ hai.

9. Phương tiện bất biến đọc được bằng bộ xử lý lưu trữ trên đó cấu trúc dòng video mã hóa, cấu trúc dòng video mã hóa bao gồm:

phần hình ảnh mã hóa bao gồm việc mã hóa chuỗi hình ảnh video; và

phần báo hiệu bao gồm việc mã hóa:

tham số thang đo thời gian trong khoảng thời gian màn trập chỉ báo số lượng đơn vị thời gian trôi qua trong một giây;

tham số nhịp đồng hồ trong khoảng thời gian màn trập chỉ báo số lượng đơn vị thời gian của đồng hồ hoạt động ở tần số của tham số thang đo thời gian trong khoảng thời gian màn trập, trong đó tham số nhịp đồng hồ trong khoảng thời gian màn trập chia cho tham số thang đo thời gian trong khoảng thời gian màn trập chỉ báo giá trị thời khoảng phơi sáng;

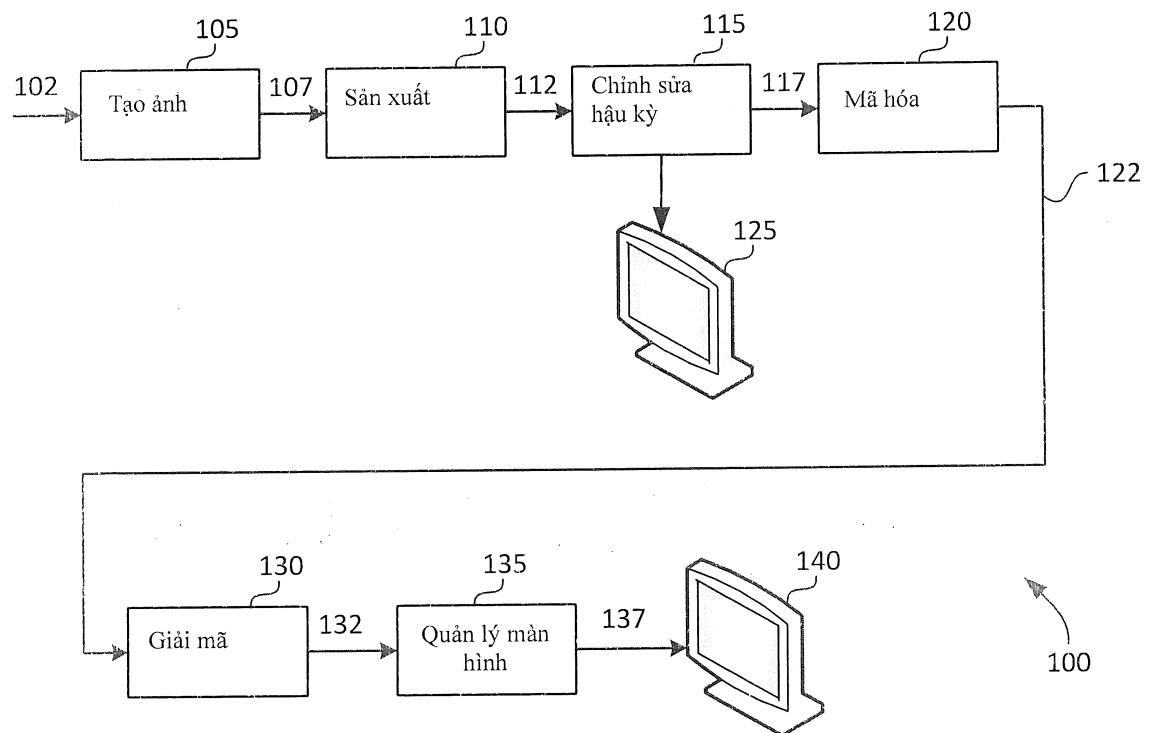
cờ thời khoảng trong khoảng thời gian màn trập chỉ báo liệu thông tin thời khoảng phơi sáng có cố định cho tất cả các lớp con thời gian trong phần hình ảnh mã hóa không; và

nếu cờ thời khoảng trong khoảng thời gian màn trập chỉ báo rằng thông tin thời khoảng phơi sáng là cố định, thì phiên bản giải mã của chuỗi hình ảnh video cho tất cả các lớp con thời gian trong phần hình ảnh mã hóa được giải mã bằng cách tính toán giá trị thời khoảng phơi sáng dựa vào tham số thang đo thời gian trong khoảng thời gian màn trập và tham số nhịp đồng hồ trong khoảng thời gian màn trập, nếu không thì

phần báo hiệu bao gồm một hoặc nhiều mảng tham số lớp con, trong đó các giá trị trong một hoặc nhiều mảng tham số lớp con kết hợp với tham số thang đo thời gian trong khoảng thời gian màn trập được sử dụng để tính toán cho mỗi lớp con một giá trị thời khoảng phơi sáng lớp con tương ứng để hiển thị phiên bản giải mã của lớp con thời gian của chuỗi hình ảnh video.

10. Phương tiện bất biến đọc được bằng bộ xử lý theo điểm 9, trong đó một hoặc nhiều mảng tham số lớp con bao gồm mảng các giá trị nhịp đồng hồ trong khoảng thời gian màn trập lớp con, mỗi giá trị chỉ báo số lượng đơn vị thời gian của đồng hồ hoạt động ở tần số của tham số thang đo thời gian trong khoảng thời gian màn trập.

1/3

**FIG. 1**

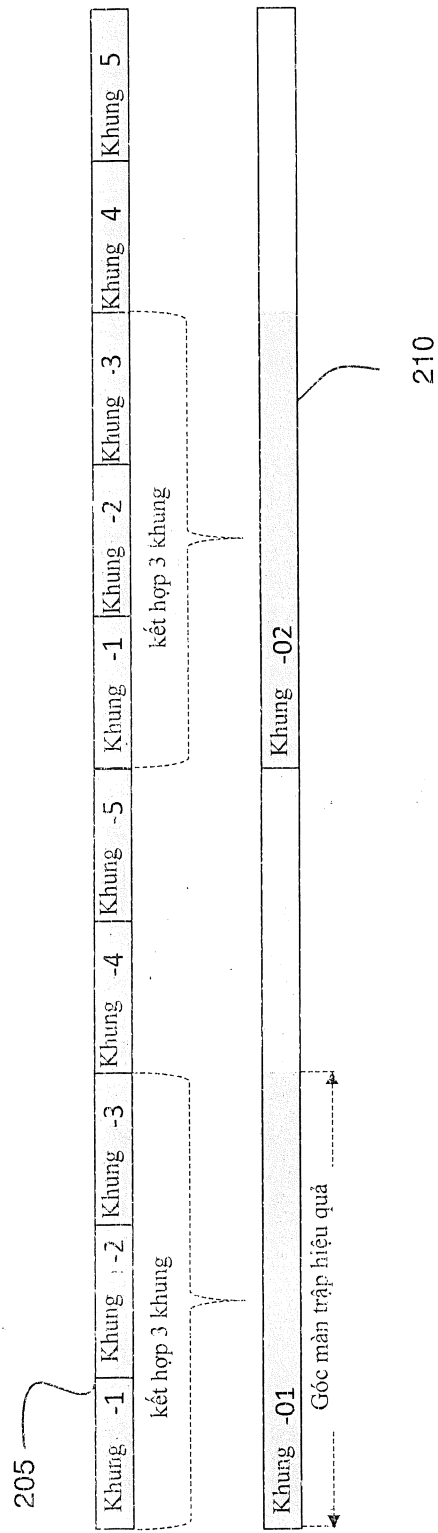
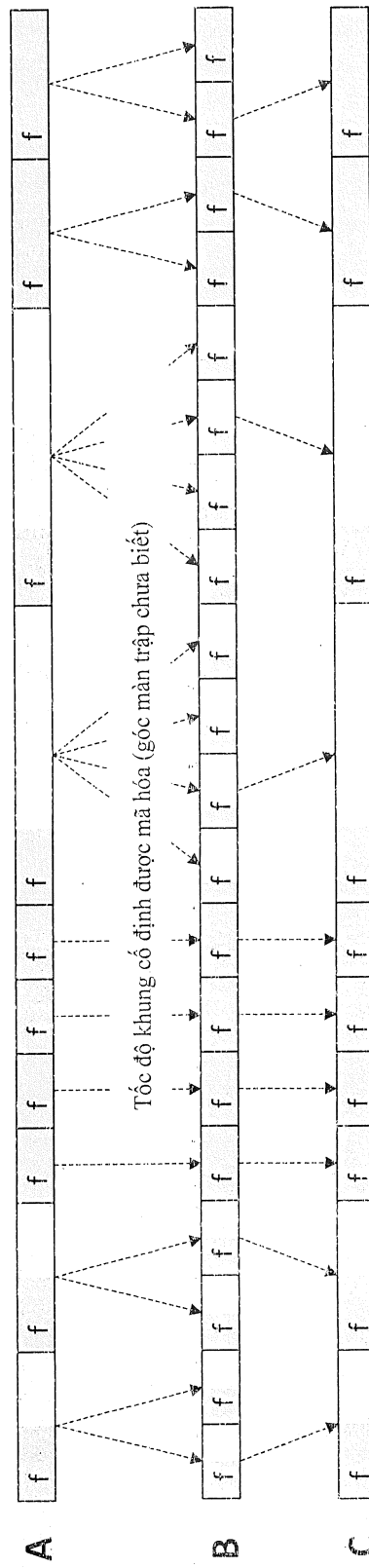


FIG. 2

Tốc độ khung thay đổi và góc màn trập thay đổi góc



Tốc độ khung biến đổi và góc màn trập biến đổi được kết xuất

FIG. 3

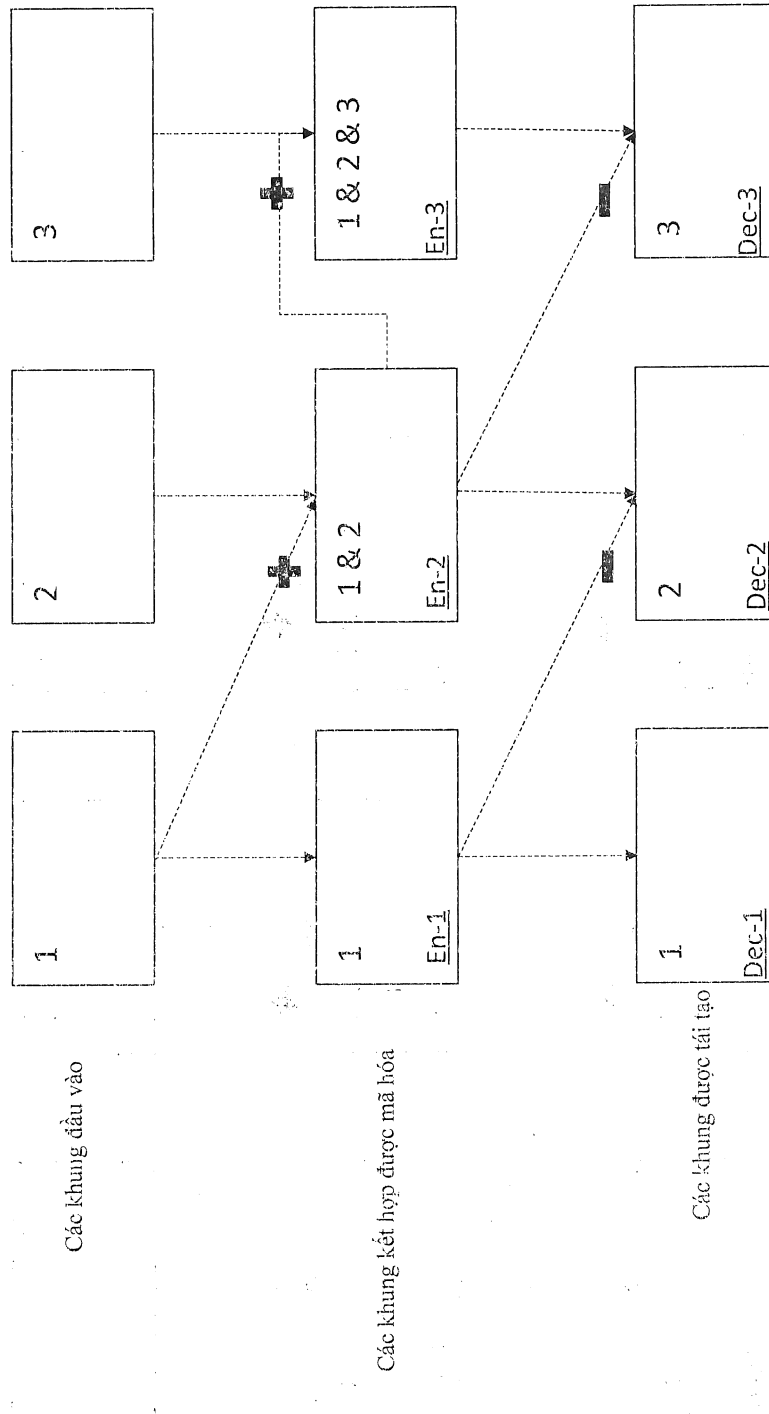


FIG. 4