



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ
(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN) (11) 
CỤC SỞ HỮU TRÍ TUỆ
1-0052095
(51)^{2020.01} H04N 19/70; H04N 19/172; H04N 19/30; H04N 19/105; H04N 19/187 (13) B

(21) 1-2022-02492 (22) 18/09/2020
(86) PCT/US2020/051605 18/09/2020 (87) WO 2021/061530 01/04/2021
(30) 62/905,128 24/09/2019 US
(45) 25/09/2025 450 (43) 25/07/2022 412A
(73) HUAWEI TECHNOLOGIES CO., LTD. (CN)
Huawei Administration Building Bantian, Longgang District Shenzhen, Guangdong
518129, China
(72) WANG, Ye-Kui (US).
(74) Công ty Luật TNHH Phạm và Liên danh (PHAM & ASSOCIATES)

(54) PHƯƠNG PHÁP TẠO MÃ, THIẾT BỊ TẠO MÃ VIDEO, BỘ MÃ HÓA, BỘ GIẢI
MÃ VÀ VẬT GHI MÁY TÍNH ĐƯỢC ĐƯỢC BẤT BIẾN

(21) 1-2022-02492

(57) Sáng chế đề cập đến cơ chế tạo mã video. Cơ chế này bao gồm mã hóa dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã. Tập hợp tham số video (video parameter set, VPS) cũng được mã hóa vào dòng bit. VPS bao gồm mã nhận diện chế độ tập hợp lớp đầu ra (output layer set, OLS) (ols_mode_idc) xác định rằng tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS. Dòng bit được lưu trữ để truyền thông về phía bộ giải mã.

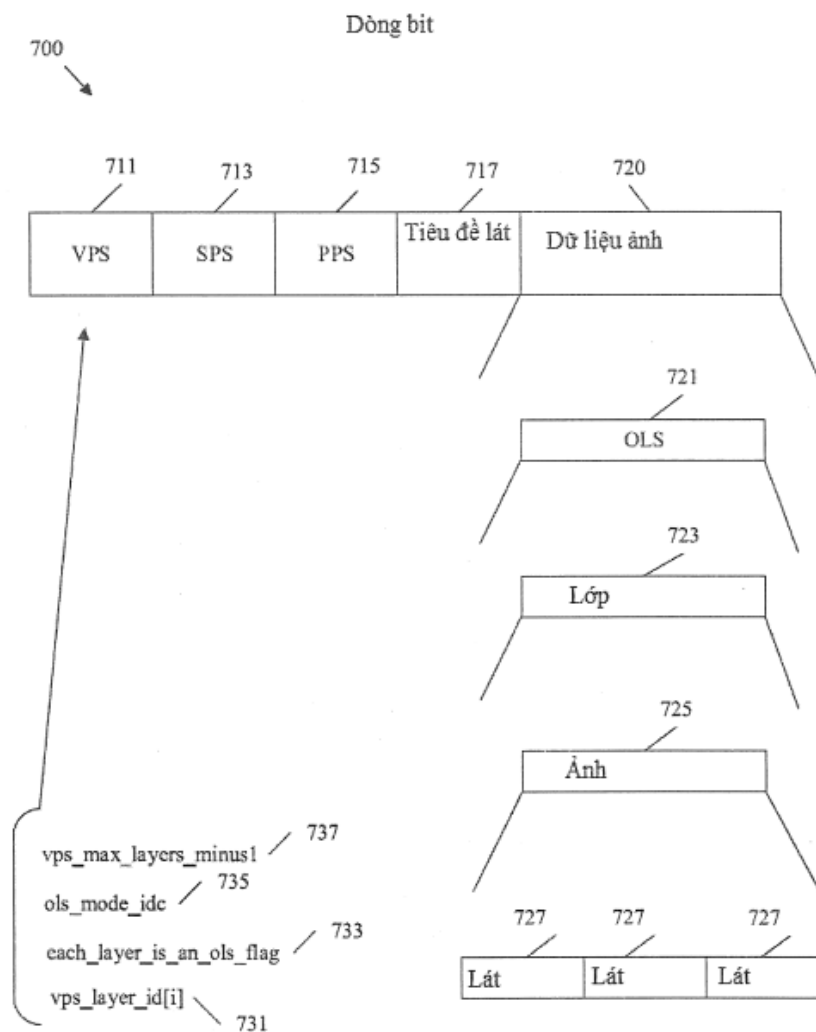


Fig. 7

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến kỹ thuật tạo mã video, và đề cập cụ thể đến việc tạo cấu hình các tập hợp lớp đầu ra (output layer set, OLS) trong các dòng bit nhiều lớp để hỗ trợ tính khởi mở không gian và hệ số tín trên tạp (signal to noise, SNR).

Tình trạng kỹ thuật của sáng chế

Lượng dữ liệu video cần mô tả đoạn video tương đối ngắn có thể là rất lớn, có thể gặp khó khăn khi dữ liệu cần được phát trực tuyến hoặc ngược lại được truyền thông trên mạng truyền thông có dung lượng băng thông hữu hạn. Do vậy, dữ liệu video thường được nén trước khi được truyền thông giữa các mạng viễn thông hiện đại. Kích thước của video cũng có thể là vấn đề khi video được lưu trữ trên thiết bị lưu trữ do các tài nguyên bộ nhớ có thể bị giới hạn. Các thiết bị nén video thường sử dụng phần mềm và/hoặc phần cứng ở phía nguồn để tạo mã dữ liệu video trước khi truyền hoặc lưu trữ, nhờ đó giảm lượng dữ liệu cần để biểu diễn các ảnh video số. Sau đó, dữ liệu được nén được nhận ở phía đích bởi thiết bị giải nén video giải mã dữ liệu video. Với các tài nguyên mạng hữu hạn và các nhu cầu ngày càng tăng có chất lượng video cao hơn, các kỹ thuật nén và giải nén được cải thiện mà cải thiện tỷ lệ nén mà không ảnh hưởng đến chất lượng ảnh được mong muốn.

Bản chất kỹ thuật của sáng chế

Theo phương án thực hiện, sáng chế đề cập đến phương pháp được thực hiện bằng bộ giải mã, phương pháp bao gồm các bước: nhận, bằng bộ nhận của bộ giải mã, dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã và tập hợp tham số video (video parameter set, VPS), trong đó VPS bao gồm mã nhận diện chế độ OLS (ols_mode_idc) xác định rằng tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS; xác định, bằng bộ xử lý của bộ

giải mã, lớp đầu ra dựa trên `ols_mode_idc` trong VPS; và giải mã, bằng bộ xử lý của bộ giải mã, ảnh được tạo mã từ lớp đầu ra để tạo ảnh được giải mã.

Các lớp ảnh có thể được sử dụng để hỗ trợ tính khả mở. Chẳng hạn, video có thể được tạo mã thành nhiều lớp. Lớp có thể được tạo mã mà không dựa vào các lớp khác. Lớp này được gọi là lớp đồng truyền phát. Do đó, lớp đồng truyền phát có thể được giải mã mà không dựa vào các lớp khác. Lấy ví dụ khác, lớp có thể được tạo mã nhờ sử dụng dự báo ngoài lớp. Điều này cho phép lớp hiện tại được tạo mã bằng cách bao gồm chỉ các khác biệt giữa lớp hiện tại và lớp tham chiếu. Chẳng hạn, lớp hiện tại và lớp tham chiếu có thể chứa chuỗi video tương tự được tạo mã bằng cách thay đổi đặc tính, chẳng hạn SNR, kích thước ảnh, tốc độ khung, v.v.. Một số hệ thống tạo mã video được tạo cấu hình để chỉ giải mã và xuất ra lớp được mã hóa cao nhất như được ký hiệu là định danh (identifier, ID) lớp cùng với một hoặc nhiều lớp thấp hơn được chỉ báo. Điều này có thể đặt ra vấn đề về tính khả mở, do bộ giải mã có thể không muốn giải mã lớp cao nhất. Điều này gây ra lỗi khi cố thử tính khả mở của video trong các hệ thống này. Đây có thể là vấn đề nghiêm trọng do yêu cầu rằng các bộ giải mã luôn hỗ trợ lớp cao nhất dẫn đến hệ thống mà không thể mở rộng sang các lớp trung gian dựa trên các yêu cầu mạng và phần cứng khác nhau.

Ví dụ bao gồm cơ chế sử dụng các OLS để hỗ trợ tính khả mở. Điều này bao gồm cả tính khả mở không gian và tính khả mở SNR. Ví dụ sử dụng phần tử cú pháp `ols_mode_idc` để sử dụng cùng với các OLS. Phần tử cú pháp `ols_mode_idc` có thể được chứa trong VPS và có thể được gán bằng 0 để chỉ báo rằng tổng số OLS trong chuỗi video bằng tổng số lớp được xác định trong VPS, rằng OLS thứ i bao gồm các lớp từ 0 đến i , bao gồm hai giá trị đầu mút, và đối với mỗi OLS, chỉ lớp cao nhất được xuất ra. Điều này hỗ trợ tính khả mở do bộ giải mã có thể nhanh chóng xác định rằng, chẳng hạn, OLS thứ ba chứa các lớp từ 0 đến 3 và lớp thứ ba cần được giải mã dựa trên các lớp từ 0 đến 2 và được xuất ra. Do đó, bộ giải mã có thể chỉ nhận các lớp cần để giải mã lớp cao nhất được nhận và lớp cao nhất được nhận có thể được giải mã và được hiển thị. Theo cách này, tổng số lớp được mã hóa có thể không ảnh hưởng đến quá trình giải mã và sai số có thể được tránh. Thực tế, các cơ chế được bộc lộ làm tăng chức năng của bộ mã

hóa và/hoặc bộ giải mã. Ngoài ra, các cơ chế được bộc lộ có thể giảm kích thước dòng bit, và do vậy giảm bộ xử lý, bộ nhớ, và/hoặc tận dụng tài nguyên mạng ở cả bộ mã hóa và bộ giải mã. Theo phương án thực hiện cụ thể, `ols_mode_idc` giúp tiết kiệm bit trong các dòng bit được mã hóa mà chứa nhiều OLS, trong số đó nhiều dữ liệu được chia sẻ, do vậy tiết kiệm trong các máy chủ streaming (phát trực tuyến) và tiết kiệm băng thông để truyền các dòng bit này.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó `ols_mode_idc` xác định rằng OLS thứ i bao gồm các lớp có các chỉ số lớp từ 0 đến i , bao gồm hai giá trị đầu mút.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó `ols_mode_idc` xác định rằng đối với mỗi OLS chỉ lớp cao nhất trong mỗi OLS là lớp đầu ra.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó `ols_mode_idc` bằng 0.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó VPS bao gồm các lớp lớn nhất của VPS trừ 1 (`vps_max_layers_minus1`) xác định số lượng lớp được xác định bởi VPS, vốn là số lượng lớp được phép lớn nhất trong mỗi chuỗi video được tạo mã (coded video sequence, CVS) viện dẫn đến VPS.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó tổng số OLSs (`TotalNumOls`) bằng `vps_max_layers_minus1` cộng 1 khi `ols_mode_idc` bằng 0 hoặc khi `ols_mode_idc` bằng 1.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó số lượng lớp trong OLS thứ i (`NumLayersInOls[i]`) và ID lớp trong OLS (`LayerIdInOls[i][j]`) xác định giá trị ID lớp tiêu đề đơn vị lớp trừu tượng mạng (network abstraction layer, NAL) (`nuh_layer_id`) của lớp thứ j trong OLS thứ i được dẫn xuất như sau:

`NumLayersInOls[0] = 1`

`LayerIdInOls[0][0] = vps_layer_id[0]`

```

for(i = 1, i < TotalNumOlss; i++) {
    if(each_layer_is_an_ols_flag) {
        NumLayersInOls[i] = 1
        LayerIdInOls[i][0] = vps_layer_id[i]
    } else if(ols_mode_idc == 0 || ols_mode_idc == 1) {
        NumLayersInOls[i] = i + 1
        for(j = 0; j < NumLayersInOls[i]; j++)
            LayerIdInOls[i][j] = vps_layer_id[j]
    }
}

```

trong đó `vps_layer_id[i]` là ID lớp VPS thứ *i*, `TotalNumOlss` là tổng số OLS được xác định bởi VPS, và `each_layer_is_an_ols_flag` là cờ mỗi lớp là OLS xác định liệu ít nhất một OLS chứa nhiều hơn một lớp.

Theo phương án thực hiện, sáng chế bao gồm phương pháp được thực hiện bằng bộ mã hóa, phương pháp bao gồm: mã hóa, bằng bộ xử lý của bộ mã hóa, dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã; mã hóa vào dòng bit, bằng bộ xử lý, VPS, trong đó VPS bao gồm `ols_mode_idc` xác định rằng tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS; và lưu trữ, bằng bộ nhớ được ghép nối với bộ xử lý, dòng bit để truyền thông về phía bộ giải mã.

Các lớp ảnh có thể được sử dụng để hỗ trợ tính khả mở. Chẳng hạn, video có thể được tạo mã thành nhiều lớp. Lớp có thể được tạo mã mà không tạo tham chiếu các lớp khác. Lớp này được gọi là lớp đồng truyền phát. Do đó, lớp đồng truyền phát có thể được giải mã mà không dựa vào các lớp khác. Lấy ví dụ khác, lớp có thể được tạo mã nhờ sử dụng dự báo ngoài lớp. Điều này cho phép lớp hiện tại được tạo mã bằng cách bao gồm chỉ các khác biệt giữa lớp hiện tại và lớp tham chiếu. Chẳng hạn, lớp hiện tại và lớp tham chiếu có thể chứa chuỗi video tương tự được tạo mã bằng cách thay đổi đặc tính, chẳng hạn SNR, kích thước ảnh, tốc độ khung, v.v.. Một số hệ thống tạo mã video được tạo cấu hình để chỉ giải mã và xuất ra lớp được mã hóa cao nhất như được ký hiệu là ID lớp cùng với một hoặc nhiều lớp thấp hơn được chỉ báo. Điều này có thể đặt ra vấn đề về tính khả mở, do bộ giải mã có thể không muốn giải mã lớp cao nhất. Điều này gây ra lỗi khi cố thử tính khả mở của video trong các hệ thống này. Đây có

thể là vấn đề nghiêm trọng do yêu cầu rằng các bộ giải mã luôn hỗ trợ lớp cao nhất dẫn đến hệ thống mà không thể mở rộng sang các lớp trung gian dựa trên các yêu cầu mạng và phần cứng khác nhau.

Ví dụ bao gồm cơ chế sử dụng các OLS để hỗ trợ tính khả mở. Điều này bao gồm cả tính khả mở không gian và tính khả mở SNR. Ví dụ sử dụng `ols_mode_idc` phần tử cú pháp để sử dụng cùng với các OLS. Phần tử cú pháp `ols_mode_idc` có thể được chứa trong VPS và có thể được gán bằng 0 để chỉ báo rằng tổng số OLS trong chuỗi video bằng tổng số lớp được xác định trong VPS, rằng OLS thứ i bao gồm các lớp từ 0 đến i , bao gồm hai giá trị đầu mút, và đối với mỗi OLS chỉ lớp cao nhất được xuất ra. Điều này hỗ trợ tính khả mở do bộ giải mã có thể nhanh chóng xác định rằng, chẳng hạn, OLS thứ ba chứa các lớp từ 0 đến 3 và lớp thứ ba cần được giải mã dựa trên các lớp từ 0 đến 2 và được xuất ra. Do đó, bộ giải mã có thể chỉ nhận các lớp cần để giải mã lớp cao nhất được nhận và lớp cao nhất được nhận có thể được giải mã và được hiển thị. Theo cách này, tổng số lớp được mã hóa có thể không ảnh hưởng đến quá trình giải mã và sai số có thể được tránh. Thực tế, các cơ chế được bộc lộ làm tăng chức năng của bộ mã hóa và/hoặc bộ giải mã. Ngoài ra, các cơ chế được bộc lộ có thể giảm kích thước dòng bit, và do vậy giảm bộ xử lý, bộ nhớ, và/hoặc tận dụng tài nguyên mạng ở cả bộ mã hóa và bộ giải mã. Theo phương án thực hiện cụ thể, `ols_mode_idc` giúp tiết kiệm bit trong các dòng bit được mã hóa mà chứa nhiều OLS, trong số đó nhiều dữ liệu được chia sẻ, do vậy tiết kiệm trong các máy chủ streaming và tiết kiệm băng thông để truyền các dòng bit này.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó `ols_mode_idc` xác định rằng an OLS thứ i bao gồm các lớp có các chỉ số lớp từ 0 đến i , bao gồm hai giá trị đầu mút.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó `ols_mode_idc` xác định rằng đối với mỗi OLS chỉ lớp cao nhất trong mỗi OLS là lớp đầu ra.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề xuất, trong đó `ols_mode_idc` bằng 0.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó VPS bao gồm `vps_max_layers_minus1` xác định số lượng lớp được xác định bởi VPS, vốn là số lượng lớp được phép lớn nhất trong mỗi CVS viện dẫn đến VPS.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó `TotalNumOls` bằng `vps_max_layers_minus1` cộng 1 khi `ols_mode_idc` bằng 0 hoặc khi `ols_mode_idc` bằng 1.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó `NumLayersInOls[i]` và `LayerIdInOls[i][j]` xác định giá trị `nuh_layer_id` của lớp thứ `j` trong OLS thứ `i` được dẫn xuất như sau:

```
NumLayersInOls[0] = 1
LayerIdInOls[0][0] = vps_layer_id[0]
for(i = 1, i < TotalNumOls; i++) {
    if(each_layer_is_an_ols_flag) {
        NumLayersInOls[i] = 1
        LayerIdInOls[i][0] = vps_layer_id[i]
    } else if(ols_mode_idc == 0 || ols_mode_idc == 1) {
        NumLayersInOls[i] = i + 1
        for(j = 0; j < NumLayersInOls[i]; j++)
            LayerIdInOls[i][j] = vps_layer_id[j]
```

trong đó `vps_layer_id[i]` là ID lớp VPS thứ `i`, `TotalNumOls` là tổng số OLS được xác định bởi VPS, và `each_layer_is_an_ols_flag` là cờ mỗi lớp là OLS xác định liệu ít nhất một OLS chứa nhiều hơn một lớp.

Theo phương án thực hiện, sáng chế bao gồm thiết bị tạo mã video bao gồm: bộ xử lý, bộ nhận được ghép nối với bộ xử lý, bộ nhớ được ghép nối với bộ xử lý, và bộ truyền được ghép nối với bộ xử lý, trong đó bộ xử lý, bộ nhận, bộ nhớ, và bộ truyền được tạo cấu hình thực hiện phương pháp theo khía cạnh bất kỳ trong các khía cạnh trước đó.

Theo phương án thực hiện, sáng chế bao gồm vật ghi máy tính đọc được bất biến bao gồm sản phẩm chương trình máy tính để sử dụng bởi thiết bị tạo mã video, sản phẩm chương trình máy tính bao gồm các lệnh máy tính thực thi được được lưu trữ trên vật ghi máy tính đọc được bất biến sao cho khi được thực thi bằng bộ xử lý khiến thiết bị tạo mã video thực hiện phương pháp theo khía cạnh bất kỳ trong các khía cạnh trước đó.

Theo phương án thực hiện, sáng chế bao gồm bộ giải mã bao gồm: phương tiện nhận để nhận dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã và VPS, trong đó VPS bao gồm `ols_mode_idc` xác định rằng tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS; phương tiện xác định để xác định lớp đầu ra dựa trên `ols_mode_idc` trong VPS; phương tiện giải mã để giải mã ảnh được tạo mã từ lớp đầu ra để tạo ảnh được giải mã; và phương tiện chuyển tiếp để chuyển tiếp ảnh được giải mã để hiển thị dưới dạng một phần của chuỗi video được giải mã.

Các lớp ảnh có thể được sử dụng để hỗ trợ tính khả mở. Chẳng hạn, video có thể được tạo mã thành nhiều lớp. Lớp có thể được tạo mã mà không tạo tham chiếu các lớp khác. Lớp này được gọi là lớp đồng truyền phát. Do đó, lớp đồng truyền phát có thể được giải mã mà không dựa vào các lớp khác. Lấy ví dụ khác, lớp có thể được tạo mã nhờ sử dụng dự báo ngoài lớp. Điều này cho phép lớp hiện tại được tạo mã bằng cách bao gồm chỉ các khác biệt giữa lớp hiện tại và lớp tham chiếu. Chẳng hạn, lớp hiện tại và lớp tham chiếu có thể chứa chuỗi video tương tự được tạo mã bằng cách thay đổi đặc tính, chẳng hạn SNR, kích thước ảnh, tốc độ khung, v.v.. Một số hệ thống tạo mã video được tạo cấu hình để chỉ giải mã và xuất ra lớp được mã hóa cao nhất như được ký hiệu là ID lớp cùng với một hoặc nhiều lớp thấp hơn được chỉ báo. Điều này có thể đặt ra vấn đề về tính khả mở, do bộ giải mã có thể không muốn giải mã lớp cao nhất. Điều này gây ra lỗi khi cố thử tính khả mở của video trong các hệ thống này. Đây có thể là vấn đề nghiêm trọng do yêu cầu rằng các bộ giải mã luôn hỗ trợ lớp cao nhất dẫn đến hệ thống mà không thể mở rộng sang các lớp trung gian dựa trên các yêu cầu mạng và phần cứng khác nhau.

Ví dụ bao gồm cơ chế sử dụng các OLS để hỗ trợ tính khả mở. Điều này bao gồm cả tính khả mở không gian và tính khả mở SNR. Ví dụ sử dụng phân tử cú pháp `ols_mode_idc` để sử dụng cùng với các OLS. Phân tử cú pháp `ols_mode_idc` có thể được chứa trong VPS và có thể được gán bằng 0 để chỉ báo rằng tổng số OLS trong chuỗi video bằng tổng số lớp được xác định trong VPS, rằng OLS thứ i bao gồm các lớp từ 0 đến i , bao gồm hai giá trị đầu mút, và đối với mỗi OLS chỉ lớp cao nhất được xuất ra. Điều này hỗ trợ tính khả mở do bộ giải mã có thể nhanh chóng xác định rằng, chẳng hạn, OLS thứ ba chứa các lớp từ 0 đến 3 và lớp thứ ba cần được giải mã dựa trên các lớp từ 0 đến 2 và được xuất ra. Do đó, bộ giải mã có thể chỉ nhận các lớp cần để giải mã lớp cao nhất được nhận và lớp cao nhất được nhận có thể được giải mã và được hiển thị. Theo cách này, tổng số lớp được mã hóa có thể không ảnh hưởng đến quá trình giải mã và sai số có thể được tránh. Thực tế, các cơ chế được bộc lộ làm tăng chức năng của bộ mã hóa và/hoặc bộ giải mã. Ngoài ra, các cơ chế được bộc lộ có thể giảm kích thước dòng bit, và do vậy giảm bộ xử lý, bộ nhớ, và/hoặc tận dụng tài nguyên mạng ở cả bộ mã hóa và bộ giải mã. Theo phương án thực hiện cụ thể, `ols_mode_idc` giúp tiết kiệm bit trong các dòng bit được mã hóa mà chứa nhiều OLS, trong số đó nhiều dữ liệu được chia sẻ, do vậy tiết kiệm trong các máy chủ streaming và tiết kiệm băng thông để truyền các dòng bit này.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó bộ giải mã còn được tạo cấu hình để thực hiện phương pháp theo khía cạnh bất kỳ trong các khía cạnh trước đó.

Theo phương án thực hiện, sáng chế bao gồm bộ mã hóa bao gồm: phương tiện mã hóa để: mã hóa dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã; và mã hóa into dòng bit VPS, trong đó VPS bao gồm `ols_mode_idc` xác định rằng tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS; và phương tiện lưu trữ để lưu trữ dòng bit để truyền thông về phía bộ giải mã.

Các lớp ảnh có thể được sử dụng để hỗ trợ tính khả mở. Chẳng hạn, video có thể được tạo mã thành nhiều lớp. Lớp có thể được tạo mã mà không tạo tham

chiếu các lớp khác. Lớp này được gọi là lớp đồng truyền phát. Do đó, lớp đồng truyền phát có thể được giải mã mà không dựa vào các lớp khác. Lấy ví dụ khác, lớp có thể được tạo mã nhờ sử dụng dự báo ngoài lớp. Điều này cho phép lớp hiện tại được tạo mã bằng cách bao gồm chỉ các khác biệt giữa lớp hiện tại và lớp tham chiếu. Chẳng hạn, lớp hiện tại và lớp tham chiếu có thể chứa chuỗi video tương tự được tạo mã bằng cách thay đổi đặc tính, chẳng hạn SNR, kích thước ảnh, tốc độ khung, v.v.. Một số hệ thống tạo mã video được tạo cấu hình để chỉ giải mã và xuất ra lớp được mã hóa cao nhất như được ký hiệu là ID lớp cùng với một hoặc nhiều lớp thấp hơn được chỉ báo. Điều này có thể đặt ra vấn đề về tính khả mở, do bộ giải mã có thể không muốn giải mã lớp cao nhất. Điều này gây ra lỗi khi cố thử tính khả mở của video trong các hệ thống này. Đây có thể là vấn đề nghiêm trọng do yêu cầu rằng các bộ giải mã luôn hỗ trợ lớp cao nhất dẫn đến hệ thống mà không thể mở rộng sang các lớp trung gian dựa trên các yêu cầu mạng và phần cứng khác nhau.

Ví dụ bao gồm cơ chế sử dụng các OLS để hỗ trợ tính khả mở. Điều này bao gồm cả tính khả mở không gian và tính khả mở SNR. Ví dụ sử dụng phân tử cú pháp `ols_mode_idc` để sử dụng cùng với các OLS. Phân tử cú pháp `ols_mode_idc` có thể được chứa trong VPS và có thể được gán bằng 0 để chỉ báo rằng tổng số OLS trong chuỗi video bằng tổng số lớp được xác định trong VPS, rằng OLS thứ i bao gồm các lớp từ 0 đến i , bao gồm hai giá trị đầu mút, và thay đổi với mỗi OLS chỉ lớp cao nhất được xuất ra. Điều này hỗ trợ tính khả mở do bộ giải mã có thể nhanh chóng xác định rằng, chẳng hạn, OLS thứ ba chứa các lớp từ 0 đến 3 và lớp thứ ba cần được giải mã dựa trên các lớp từ 0 đến 2 và được xuất ra. Do đó, bộ giải mã có thể chỉ nhận các lớp cần để giải mã lớp cao nhất được nhận và lớp cao nhất được nhận có thể được giải mã và được hiển thị. Theo cách này, tổng số lớp được mã hóa có thể không ảnh hưởng đến quá trình giải mã và sai số có thể được tránh. Thực tế, các cơ chế được bộc lộ làm tăng chức năng của bộ mã hóa và/hoặc bộ giải mã. Ngoài ra, các cơ chế được bộc lộ có thể giảm kích thước dòng bit, và do vậy giảm bộ xử lý, bộ nhớ, và/hoặc tận dụng tài nguyên mạng ở cả bộ mã hóa và bộ giải mã. Theo phương án thực hiện cụ thể, `ols_mode_idc` giúp tiết kiệm bit trong các dòng bit được mã hóa mà chứa nhiều

OLS, trong số đó nhiều dữ liệu được chia sẻ, do vậy tiết kiệm trong các máy chủ streaming và tiết kiệm băng thông để truyền các dòng bit này.

Một cách tùy chọn, theo khía cạnh bất kỳ trong các khía cạnh trước đó, cách thực hiện khác của khía cạnh đề cập đến, trong đó bộ mã hóa còn được tạo cấu hình để thực hiện phương pháp theo khía cạnh bất kỳ trong các khía cạnh trước đó.

Để cho rõ ràng, phương án bất kỳ trong các phương án thực hiện nêu trên có thể được kết hợp với một hoặc nhiều phương án trong các phương án thực hiện nêu trên khác để tạo phương án thực hiện mới trong phạm vi của sáng chế.

Các dấu hiệu này và các dấu hiệu khác sẽ được hiểu rõ hơn từ phần mô tả chi tiết dưới đây dựa vào các hình vẽ đi kèm và các điểm yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

Để hiểu rõ hơn sáng chế, tham khảo mô tả vắn tắt dưới đây, dựa vào các hình vẽ đi kèm và phần mô tả chi tiết, trong đó các số chỉ dẫn giống nhau biểu diễn các bộ phận giống nhau.

Fig.1 là lưu đồ của phương pháp tạo mã ví dụ tín hiệu video;

Fig.2 là sơ đồ của hệ thống mã hóa và giải mã (codec) lấy làm ví dụ để tạo mã video;

Fig.3 là sơ đồ minh họa bộ mã hóa video ví dụ;

Fig.4 là sơ đồ minh họa bộ giải mã video ví dụ;

Fig.5 là sơ đồ minh họa chuỗi video nhiều lớp lấy làm ví dụ được tạo cấu hình cho dự báo ngoài lớp;

Fig.6 là sơ đồ minh họa chuỗi video lấy làm ví dụ có các OLS được tạo cấu hình cho tính khả mở không gian và/hoặc SNR;

Fig.7 là sơ đồ minh họa dòng bit lấy làm ví dụ bao gồm các OLS được tạo cấu hình cho tính khả mở;

Fig.8 là sơ đồ của thiết bị tạo mã video lấy làm ví dụ;

Fig.9 là lưu đồ của phương pháp lấy làm ví dụ mã hóa chuỗi video có các OLS được tạo cấu hình cho tính khả mở;

Fig.10 là lưu đồ của phương pháp lấy làm ví dụ giải mã chuỗi video bao gồm OLS được tạo cấu hình cho tính khả mở; và

Fig.11 là sơ đồ của hệ thống tạo mã chuỗi video lấy làm ví dụ với các OLS được tạo cấu hình cho tính khả mở.

Mô tả chi tiết sáng chế

Ngay từ đầu cần hiểu rằng mặc dù cách triển khai minh họa của một hoặc nhiều phương án thực hiện được nêu dưới đây, các hệ thống và/hoặc các phương pháp được bộc lộ có thể được thực hiện nhờ sử dụng các kỹ thuật, đang được biết đến hoặc hiện có. Sáng chế không bị giới hạn ở các cách thực hiện minh họa, các hình vẽ, và các kỹ thuật được minh họa dưới đây, bao gồm các thiết kế và các cách thực hiện lấy làm ví dụ được minh họa và mô tả ở đây, nhưng có thể được chỉnh sửa trong phạm vi của các điểm yêu cầu bảo hộ đi kèm cùng với toàn bộ phạm vi của các điểm tương đương.

Các cụm từ sau được định nghĩa như sau trừ khi được sử dụng trong ngữ cảnh ngược lại ở đây. Cụ thể là, các định nghĩa sau nhằm làm sáng tỏ hơn sáng chế. Tuy nhiên, các cụm từ có thể được mô tả khác nhau trong các ngữ cảnh khác nhau. Do đó, các định nghĩa sau nên được hiểu như là bổ sung và không được xem là giới hạn các định nghĩa khác bất kỳ của phần mô tả được nêu cho các cụm từ này ở đây.

Dòng bit là chuỗi bit bao gồm dữ liệu video được nén để truyền giữa bộ mã hóa và bộ giải mã. Bộ mã hóa là thiết bị được tạo cấu hình để sử dụng các quá trình mã hóa để nén dữ liệu video vào dòng bit. Bộ giải mã là thiết bị được tạo cấu hình để sử dụng các quá trình giải mã để tái tạo dữ liệu video từ dòng bit để hiển thị. Ảnh là mảng của các mẫu độ sáng và/hoặc mảng của các mẫu sắc độ tạo khung hoặc trường của khung. Ảnh được mã hóa hoặc giải mã có thể được gọi là ảnh hiện tại để mô tả rõ ràng.

Đơn vị lớp trừu tượng mạng (network abstraction layer, NAL) là cấu trúc cú pháp chứa dữ liệu ở dạng tải tin chuỗi byte thô (Raw Byte Sequence Payload, RBSP), chỉ báo loại dữ liệu, và được đặt rải rác như mong muốn với các byte ngăn ngừa mô phỏng. Đơn vị NAL lớp tạo mã video (video coding layer, VCL)

là đơn vị NAL được tạo mã để chứa dữ liệu video, chẳng hạn lát ảnh được tạo mã. Đơn vị NAL phi VCL là đơn vị NAL chứa dữ liệu không phải video chẳng hạn cú pháp và/hoặc các tham số hỗ trợ giải mã dữ liệu video, hiệu năng kiểm tra tính tương hợp, hoặc các hoạt động khác. Lớp là tập hợp các đơn vị NAL VCL chia sẻ đặc tính cụ thể (chẳng hạn, độ phân giải chung, tốc độ khung, kích thước ảnh, v.v.) và các đơn vị NAL phi VCL liên kết. Các đơn vị NAL VCL của lớp có thể chia sẻ giá trị cụ thể của định danh lớp tiêu đề đơn vị NAL (`nuh_layer_id`). Ảnh được tạo mã là biểu diễn được tạo mã của ảnh bao gồm các đơn vị NAL VCL có giá trị cụ thể của ID lớp tiêu đề đơn vị NAL (`nuh_layer_id`) trong đơn vị truy nhập (access unit, AU) và chứa tất cả các đơn vị cây mã (coding tree unit, CTU) của ảnh. Ảnh được giải mã là ảnh được tạo bằng cách áp dụng quá trình giải mã cho ảnh được tạo mã. Chuỗi video được tạo mã (coded video sequence, CVS) là chuỗi AU bao gồm, theo thứ tự giải mã, một hoặc nhiều AU bắt đầu chuỗi video được tạo mã (coded video sequence start, CVSS) và một cách tùy chọn một hoặc nhiều AU không phải là các AU CVSS. AU CVSS là AU bao gồm đơn vị dự báo (prediction unit, PU) đối với mỗi lớp được xác định bởi tập hợp tham số video (video parameter set, VPS), trong đó ảnh được tạo mã trong mỗi PU là ảnh bắt đầu đối với CVS/chuỗi video lớp được tạo mã (coded layer video sequence, CLVS).

OLS là tập hợp lớp mà một hoặc nhiều lớp được xác định dưới dạng (các) lớp đầu ra. Lớp đầu ra là lớp được chỉ định để xuất (chẳng hạn, ra bộ phận hiển thị). Lớp cao nhất là lớp trong OLS có ID lớp cao nhất của tất cả các lớp trong OLS. Trong một số chế độ OLS lấy làm ví dụ, lớp cao nhất có thể luôn là lớp đầu ra. VPS là đơn vị dữ liệu chứa các tham số liên quan đến toàn bộ video. Dự báo ngoài lớp là cơ chế tạo mã ảnh hiện tại trong lớp hiện tại bằng tham chiếu đến ảnh tham chiếu trong lớp tham chiếu, trong đó ảnh hiện tại và ảnh tham chiếu được bao gồm trong cùng AU và lớp tham chiếu bao gồm `nuh_layer_id` thấp hơn lớp hiện tại.

Mã nhận diện chế độ OLS (`ols_mode_idc`) là phân tử cú pháp chỉ báo thông tin liên quan đến số lượng OLS, các lớp của các OLS, và các lớp đầu ra trong các OLS. Các lớp lớn nhất của VPS trừ 1 (`vps_max_layers_minus1`) là phân tử

cú pháp báo hiệu số lượng lớp được xác định bởi VPS, và do vậy số lượng lớp lớn nhất được phép trong CVS tương ứng. Cờ mỗi lớp là OLS (`each_layer_is_an_ols_flag`) là phần tử cú pháp báo hiệu liệu mỗi OLS trong dòng bit chứa một lớp. Tổng số OLS (`TotalNumOLss`) là biến xác định tổng số OLSs được xác định bởi VPS. Số lượng lớp trong OLS thứ i (`NumLayersInOLS[i]`) là biến xác định số lượng lớp trong OLS cụ thể được ký hiệu bằng giá trị chỉ số OLS của i . ID lớp trong OLS (`LayerIdInOLS[i][j]`) là biến xác định giá trị `nuh_layer_id` của lớp thứ j trong OLS thứ i được ký hiệu bằng chỉ số lớp j và chỉ số OLS i . `vps_layer_id[i]` là phần tử cú pháp chỉ báo ID lớp của lớp thứ i .

Các từ viết tắt sau được sử dụng ở đây, khối cây mã (Coding Tree Block, CTB), đơn vị cây mã (Coding Tree Unit, CTU), đơn vị mã (Coding Unit, CU), chuỗi video được tạo mã (Coded Video Sequence, CVS), nhóm chuyên gia video chung (Joint Video Experts Team, JVET), tập hợp ô bị giới hạn chuyển động (Motion Constrained Tile Set, MCTS), đơn vị truyền lớn nhất (Maximum Transfer Unit, MTU), lớp trừu tượng mạng (Network Abstraction Layer, NAL), tập hợp lớp đầu ra (Output Layer Set, OLS), số đếm thứ tự ảnh (Picture Order Count, POC), tải tin chuỗi byte thô (Raw Byte Sequence Payload, RBSP), tập hợp tham số chuỗi (Sequence Parameter Set, SPS), tập hợp tham số video (Video Parameter Set, VPS), và tạo mã video đa dạng (Versatile Video Coding, VVC).

Nhiều kỹ thuật nén video có thể được sử dụng để giảm kích thước của các tệp tin video có tổn hao dữ liệu nhỏ nhất. Chẳng hạn, các kỹ thuật nén video có thể bao gồm bước thực hiện dự báo không gian (chẳng hạn, trong ảnh) và/hoặc dự báo thời gian (chẳng hạn, ngoài ảnh) để giảm hoặc loại bỏ dư thừa dữ liệu trong các chuỗi video. Để tạo mã video dựa trên khối, lát video (chẳng hạn, ảnh video hoặc một phần ảnh video) có thể được phân vùng thành các khối video, có thể cũng được gọi là các khối cây, các CTB, các CTU, các CU, và/hoặc các nút mã. Các khối video trong lát được tạo mã trong (I) của ảnh được tạo mã nhờ sử dụng dự báo không gian với các mẫu tham chiếu trong các khối lân cận trong cùng ảnh. Các khối video trong lát dự báo đơn hướng được tạo mã ngoài (P) hoặc dự báo hai hướng (B) của ảnh có thể được tạo mã bằng cách sử dụng dự báo

không gian với các mẫu tham chiếu trong các khối lân cận trong cùng ảnh hoặc dự báo thời gian với các mẫu tham chiếu trong các ảnh tham chiếu khác. Các ảnh có thể được gọi là các khung và/hoặc các ảnh, và các ảnh tham chiếu có thể được gọi là các khung tham chiếu và/hoặc các ảnh tham chiếu. Dự báo không gian hoặc thời gian tạo thành khối dự báo biểu diễn khối ảnh. Dữ liệu dư là các hiệu số pixel giữa khối ảnh gốc và khối dự báo. Do đó, khối được mã hóa ngoài được mã hóa theo vectơ chuyển động trở đến khối các mẫu tham chiếu tạo khối dự báo và dữ liệu dư chỉ báo hiệu số giữa khối được tạo mã và khối dự báo. Khối được mã hóa trong được mã hóa theo chế độ tạo mã trong và dữ liệu dư. Để nén tiếp, dữ liệu dư có thể được biến đổi từ miền pixel đến miền biến đổi. Các điều này tạo thành các hệ số biến đổi dư, có thể được lượng tử. Các hệ số biến đổi được lượng tử ban đầu có thể được bố trí trong mảng hai chiều. Các hệ số biến đổi được lượng tử có thể được quét theo thứ tự để tạo vectơ một chiều của các hệ số biến đổi. Kỹ thuật tạo mã entropy có thể được áp dụng để nén nhiều hơn. Các kỹ thuật nén video này được nêu chi tiết hơn dưới đây.

Để đảm bảo video được mã hóa có thể được giải mã chính xác, video được mã hóa và giải mã theo các chuẩn tạo mã video tương ứng. Các chuẩn tạo mã video bao gồm chuẩn H.261 của ủy ban chuẩn hóa liên đoàn viễn thông quốc tế ((International Telecommunication Union, ITU) Standardization Sector, ITU-T) H.261, tổ chức quốc tế về chuẩn hóa/ủy ban cơ điện quốc tế (International Organization for Standardization/International Electrotechnical Commission, ISO/IEC) nhóm chuyên gia ảnh động (Motion Picture Experts Group, MPEG)-1 Phần 2, ITU-T H.262 hoặc ISO/IEC MPEG-2 Phần 2, ITU-T H.263, ISO/IEC MPEG-4 Phần 2, tạo mã video cao cấp (Advanced Video Coding, AVC), cũng được biết đến là ITU-T H.264 hoặc ISO/IEC MPEG-4 Phần 10, và tạo mã video hiệu suất cao (High Efficiency Video Coding, HEVC), cũng được biết đến là ITU-T H.265 hoặc MPEG-H Part 2. AVC bao gồm các phần mở rộng, chẳng hạn, tạo mã vi đeo khả mở (Scalable Video Coding, SVC), tạo mã video đa khung nhìn (Multiview Video Coding, MVC) và tạo mã video đa khung nhìn có chiều sâu (Multiview Video Coding plus Depth, MVC+D), và AVC ba chiều (3D dimensional, 3D) AVC (3D-AVC). HEVC bao gồm các phần mở rộng, chẳng

hạn, HEVC khả mở (Scalable HEVC, SHVC), HEVC đa khung nhìn (Multiview HEVC, MV-HEVC), và 3D HEVC (3D-HEVC). Nhóm chuyên gia video chung (joint video experts team, JVET) của ITU-T và ISO/IEC đã bắt đầu phát triển chuẩn tạo mã video được gọi là tạo mã video đa dụng (Versatile Video Coding, VVC). VVC được bao gồm trong WD, bao gồm JVET-O2001-v14.

Các lớp ảnh có thể được sử dụng để hỗ trợ tính khả mở. Chẳng hạn, video có thể được tạo mã thành nhiều lớp. Lớp có thể được tạo mã mà không tạo tham chiếu các lớp khác. Lớp này được gọi là lớp đồng truyền phát. Do đó, lớp đồng truyền phát có thể được giải mã mà không dựa vào các lớp khác. Lấy ví dụ khác, lớp có thể được tạo mã nhờ sử dụng dự báo ngoài lớp. Điều này cho phép lớp hiện tại được tạo mã bằng cách bao gồm chỉ các khác biệt giữa lớp hiện tại và lớp tham chiếu. Chẳng hạn, lớp hiện tại và lớp tham chiếu có thể chứa chuỗi video tương tự được tạo mã bằng cách thay đổi đặc tính, chẳng hạn SNR, kích thước ảnh, tốc độ khung, v.v..

Một số hệ thống tạo mã video được tạo cấu hình để chỉ giải mã và xuất ra lớp được mã hóa cao nhất như được ký hiệu là ID lớp cùng với một hoặc nhiều lớp thấp hơn được chỉ báo. Điều này có thể đặt ra vấn đề về tính khả mở, do bộ giải mã có thể không muốn giải mã lớp cao nhất. Cụ thể là, bộ giải mã nói chung yêu cầu lớp cao nhất mà bộ giải mã có thể hỗ trợ, nhưng bộ giải mã nói chung không thể giải mã lớp cao hơn lớp được yêu cầu. Lấy ví dụ cụ thể, bộ giải mã có thể muốn nhận và giải mã lớp thứ ba trong số 15 lớp được mã hóa. Lớp thứ ba có thể được truyền đến bộ giải mã mà không có các lớp từ 4 đến 15 do các lớp này không cần để giải mã lớp thứ ba. Nhưng bộ giải mã không thể giải mã và hiển thị đúng lớp thứ ba do lớp cao nhất (lớp 15) không xuất hiện và hệ thống video dành để luôn giải mã và hiển thị lớp cao nhất. Điều này gây ra lỗi khi cố thử tính khả mở của video trong các hệ thống này. Đây có thể là vấn đề nghiêm trọng do yêu cầu rằng các bộ giải mã luôn hỗ trợ lớp cao nhất dẫn đến hệ thống mà không thể mở rộng sang các lớp trung gian dựa trên các yêu cầu mạng và phần cứng khác nhau.

Hệ thống được bộc lộ ở đây sử dụng các OLS để hỗ trợ tính khả mở. Hệ thống bao gồm cả tính khả mở không gian và tính khả mở SNR. Tính khả mở không

gian cho phép chuỗi video được tạo mã thành các lớp sao cho các lớp được đặt vào các OLS sao cho mỗi OLS chứa đủ dữ liệu để giải mã chuỗi video thành kích thước màn hình đầu ra tương ứng. Do vậy, tính khả mở không gian có thể bao gồm tập hợp (các) lớp để giải mã video cho màn hình điện thoại thông minh, tập hợp các lớp để giải mã video cho màn hình TV lớn, và các tập hợp của các lớp cho các kích thước màn hình trung bình. Tính khả mở SNR cho phép chuỗi video được tạo mã thành các lớp sao cho các lớp được đặt vào các OLS sao cho mỗi OLS chứa đủ dữ liệu để giải mã chuỗi video ở SNR khác nhau. Do vậy, tính khả mở SNR có thể bao gồm tập hợp lớp có thể được giải mã cho video chất lượng thấp, video chất lượng cao, và các chất lượng video trung bình khác nhau dựa trên các điều kiện mạng. Sáng chế sử dụng phần tử cú pháp mã nhận diện chế độ OLS (`ols_mode_idc`) để sử dụng cùng với các OLS. Phần tử cú pháp `ols_mode_idc` có thể được chứa trong VPS và có thể được gán bằng 0 để chỉ báo rằng tổng số OLS trong chuỗi video bằng tổng số lớp được xác định trong VPS, rằng OLS thứ i bao gồm các lớp từ 0 đến i , bao gồm hai giá trị đầu mút, và đối với mỗi OLS chỉ lớp cao nhất được xuất ra. Điều này hỗ trợ tính khả mở do bộ giải mã có thể nhanh chóng xác định rằng, chẳng hạn, OLS thứ ba chứa các lớp từ 0 đến 3 và lớp thứ ba cần được giải mã dựa trên các lớp từ 0 đến 2 và được xuất ra. Do đó, bộ giải mã có thể chỉ nhận các lớp cần để giải mã lớp cao nhất được nhận và lớp cao nhất được nhận có thể được giải mã và được hiển thị. Theo cách này, tổng số lớp được mã hóa có thể không ảnh hưởng đến quá trình giải mã và sai số có thể được tránh. Thực tế, các cơ chế được bộc lộ làm tăng chức năng của bộ mã hóa và/hoặc bộ giải mã. Ngoài ra, các cơ chế được bộc lộ có thể giảm kích thước dòng bit, và do vậy giảm bộ xử lý, bộ nhớ, và/hoặc tận dụng tài nguyên mạng ở cả bộ mã hóa và bộ giải mã.

Fig.1 là lưu đồ của phương pháp hoạt động lấy làm ví dụ 100 tạo mã tín hiệu video. Cụ thể là, tín hiệu video được mã hóa ở bộ mã hóa. Quá trình mã hóa nén tín hiệu video bằng cách sử dụng các cơ chế khác nhau để giảm kích thước tệp tin video. Kích thước tệp tin nhỏ hơn cho phép tệp tin video được nén được truyền về phía người dùng, trong khi giảm phụ tải băng thông liên kết. Sau đó, bộ giải mã giải mã tệp tin video được nén để tái tạo tín hiệu video ban đầu để

hiển thị đến người dùng cuối. Quá trình giải mã thường phản xạ quá trình mã hóa để cho phép bộ giải mã tái tạo liên tục tín hiệu video.

Ở bước 101, tín hiệu video được đưa vào bộ mã hóa. Chẳng hạn, tín hiệu video có thể là tệp tin video không được nén được lưu trữ trong bộ nhớ. Lấy ví dụ khác, tệp tin video có thể được ghi nhận bởi thiết bị ghi nhận video, chẳng hạn video camera, và được mã hóa để hỗ trợ live streaming (phát trực tiếp trên mạng) video. Tệp tin video có thể bao gồm cả thành phần audio và thành phần video. Thành phần video chứa chuỗi khung ảnh mà, khi được xem lần lượt, tạo ấn tượng chuyển động hình ảnh. Các khung chứa các pixel được biểu diễn liên quan đến ánh sáng, ở đây được gọi là các thành phần độ sáng (hoặc các mẫu độ sáng), và màu sắc, được gọi là các thành phần sắc độ (hoặc các mẫu màu). Trong một số ví dụ, các khung có thể cũng chứa các giá trị chiều sâu để hỗ trợ xem ba chiều.

Ở bước 103, video được phân vùng thành các khối. Bước phân vùng bao gồm bước phân chia phụ các pixel trong mỗi khung thành các khối vuông và/hoặc hình chữ nhật để nén. Chẳng hạn, trong kỹ thuật tạo mã video chất lượng cao (High Efficiency Video Coding, HEVC) (cũng được biết là H.265 và MPEG-H Part 2) khung có thể được phân chia trước thành các CTU, vốn là các khối có kích thước định trước (chẳng hạn, 64 pixel x 64 pixel). Các CTU chứa cả các mẫu độ sáng lẫn sắc độ. Các cây mã có thể được sử dụng để phân chia các CTU thành các khối và sau đó phân chia phụ đệ quy các khối cho đến khi các cấu hình đạt được mà hỗ trợ mã hóa tiếp. Chẳng hạn, các thành phần độ sáng của khung có thể được phân chia phụ cho đến khi các khối riêng rẽ chứa các giá trị chiếu sáng tương đối đồng nhất. Ngoài ra, các thành phần sắc độ của khung có thể được phân chia phụ cho đến khi các khối riêng rẽ chứa các giá trị màu tương đối đồng nhất. Do đó, các cơ chế phân vùng thay đổi phụ thuộc vào nội dung của các khung video.

Ở bước 105, các cơ chế nén khác nhau được sử dụng để nén các khối ảnh được phân vùng ở bước 103. Chẳng hạn, dự báo ngoài và/hoặc dự báo trong có thể được sử dụng. Dự báo ngoài được thiết kế để tận dụng thực tế rằng các đối tượng trong khung cảnh thông thường có xu hướng xuất hiện trong các khung

liên tiếp. Do đó, khối mô tả đối tượng trong khung tham chiếu không cần được mô tả lặp lại trong các khung liên kế. Cụ thể là, đối tượng, chẳng hạn bảng, có thể vẫn ở nguyên một vị trí trên nhiều khung. Do vậy bảng được mô tả một lần và các khung liên kế có thể quay lại khung tham chiếu. Các cơ chế so khớp mẫu hình có thể được sử dụng để so khớp với các đối tượng trên nhiều khung. Ngoài ra, các đối tượng di chuyển có thể được biểu diễn trên nhiều khung, chẳng hạn do đối tượng di chuyển hoặc camera di chuyển. Lấy ví dụ cụ thể, video có thể thể hiện ô tô di chuyển ngang màn hình trên nhiều khung. Các vector chuyển động có thể được sử dụng để mô tả chuyển động này. Vector chuyển động là vector hai chiều tạo độ lệch từ các tọa độ của đối tượng trong khung đến các tọa độ của đối tượng trong khung tham chiếu. Thực tế, dự báo ngoài có thể mã hóa khối ảnh trong khung hiện tại dưới dạng tập hợp vector chuyển động chỉ báo độ lệch từ khối tương ứng trong khung tham chiếu.

Dự báo trong mã hóa các khối trong khung chung. Dự báo trong tận dụng thực tế rằng các thành phần độ sáng và sắc độ có xu hướng tạo cụm trong khung. Chẳng hạn, mảng lục trong phần cây sẽ được định vị gần với các mảng lục tương tự. Dự báo trong sử dụng các chế độ dự báo đa hướng (chẳng hạn, 33 trong HEVC), chế độ phẳng, và chế độ dòng trực tiếp (direct current, DC). Các chế độ định hướng chỉ báo rằng khối hiện tại giống như các mẫu của khối lân cận ở hướng tương ứng. Chế độ phẳng chỉ báo rằng chuỗi khối dọc theo hàng/cột (chẳng hạn, mặc phẳng) có thể được nội suy dựa trên các khối lân cận ở các mép hàng. Thực tế, chế độ phẳng chỉ báo dịch chuyển trơn tru độ sáng/màu sắc dọc theo hàng/cột nhờ sử dụng đường dốc gần như không đổi khi thay đổi các giá trị. Chế độ DC được sử dụng để làm mượt đường biên và chỉ báo rằng khối giống như giá trị trung bình được liên kết với các mẫu của tất cả các khối lân cận được liên kết với các hướng góc của các chế độ dự báo có hướng. Do đó, các khối dự báo trong có thể biểu diễn các khối ảnh dưới dạng các giá trị chế độ dự báo quan hệ khác nhau thay vì các giá trị thực. Ngoài ra, các khối dự báo ngoài có thể biểu diễn các khối ảnh dưới dạng các giá trị vector chuyển động thay vì các giá trị thực. Trong trường hợp bất kỳ, các khối dự báo có thể không biểu diễn chính xác các khối ảnh trong một số trường hợp. Các độ lệch bất kỳ được lưu trữ trong các

khối dư. Các phép biến đổi có thể được áp dụng cho các khối dư để nén tiếp tệp tin.

Ở bước 107, các kỹ thuật lọc khác nhau có thể được áp dụng. Trong HEVC, các bộ lọc được áp dụng theo phương tiện lọc trong vòng. Dự báo dựa trên khối nêu trên có thể dẫn đến việc tạo các ảnh có khối ở bộ giải mã. Ngoài ra, phương tiện dự báo dựa trên khối có thể mã hóa khối và sau đó tái tạo khối được mã hóa để sử dụng sau này làm khối dự báo. Phương tiện lọc trong vòng áp dụng lặp lại các bộ lọc chặn tạp âm, các bộ lọc tách khối, ALF, và các bộ lọc độ lệch mẫu thích ứng (sample adaptive offset, SAO) cho các khối/các khung. Các bộ lọc này di trú các giả tượng chặn này sao cho tệp tin được mã hóa có thể được tái tạo chính xác. Ngoài ra, các bộ lọc này di trú các giả tượng trong các khối tham chiếu được tái tạo sao cho các giả tượng ít có khả năng hơn trong việc tạo các giả tượng bổ sung trong các khối tiếp theo được mã hóa dựa trên các khối tham chiếu được tái tạo.

Khi tín hiệu video đã được phân vùng, được nén, và được lọc, dữ liệu thu được sẽ được mã hóa trong dòng bit ở bước 109. Dòng bit bao gồm dữ liệu nêu trên cũng như bất kỳ dữ liệu báo hiệu nào cần để hỗ trợ tái tạo tín hiệu video thích hợp ở bộ giải mã. Chẳng hạn, dữ liệu này có thể bao gồm dữ liệu phân vùng, dữ liệu dự báo, các khối dư, và các cờ khác nhau cung cấp các lệnh tạo mã cho bộ giải mã. Dòng bit có thể được lưu trữ trong bộ nhớ để truyền về phía bộ giải mã khi yêu cầu. Dòng bit có thể cũng được phát quảng bá và/hoặc được phát đa hướng về phía các bộ giải mã. Việc tạo dòng bit là quá trình lặp lại. Do đó, các bước 101, 103, 105, 107, và 109 có thể xuất hiện liên tục và/hoặc đồng thời trên nhiều khung và khối. Thứ tự được thể hiện trên Fig.1 được trình bày cho rõ ràng và dễ hiểu, và không nhằm giới hạn quá trình tạo mã video theo thứ tự cụ thể.

Bộ giải mã nhận dòng bit và bắt đầu quá trình giải mã ở bước 111. Cụ thể là, bộ giải mã sử dụng phương tiện giải mã entropy để biến đổi dòng bit thành cú pháp tương ứng và dữ liệu video. Bộ giải mã sử dụng dữ liệu cú pháp từ dòng bit để xác định các phân vùng cho các khung ở bước 111. Việc phân vùng cần so khớp các kết quả phân vùng khối ở bước 103. Mã hóa/giải mã entropy như

được sử dụng ở bước 111 đang được mô tả. Bộ mã hóa tạo nhiều lựa chọn trong quá trình nén, chẳng hạn lựa chọn các phương tiện phân vùng khối từ vài lựa chọn khả thi dựa trên định vị không gian các giá trị trong (các) ảnh đầu vào. Bảo hiệu các lựa chọn chính xác có thể sử dụng số lượng bin lớn. Như được sử dụng ở đây, bin là giá trị nhị phân được xử lý như là biến (chẳng hạn, giá trị bit có thể thay đổi tùy thuộc ngữ cảnh). Kỹ thuật tạo mã entropy cho phép bộ mã hóa để loại bỏ bất kỳ tùy chọn mà rõ ràng không khả thi cho trường hợp cụ thể, để lại tập hợp tùy chọn được phép. Sau đó, mỗi tùy chọn được phép được gán từ mã. Chiều dài của các từ mã dựa trên số lượng tùy chọn được phép (chẳng hạn, một bin cho hai tùy chọn, hai bin cho ba đến bốn tùy chọn, v.v.) Sau đó, bộ mã hóa sẽ mã hóa từ mã cho tùy chọn được chọn. Phương tiện này giảm kích thước của các từ mã do các từ mã lớn như mong muốn để chỉ báo duy nhất lựa chọn từ tập con nhỏ của các tùy chọn được phép ngược với việc chỉ báo duy nhất lựa chọn từ tập hợp có thể lớn của tất cả các tùy chọn khả thi. Sau đó, bộ giải mã giải mã lựa chọn bằng cách xác định tập hợp các tùy chọn được phép theo cách tương tự với bộ mã hóa. Bằng cách xác định tập hợp các tùy chọn được phép, bộ giải mã có thể đọc từ mã và xác định lựa chọn được thực hiện bởi bộ mã hóa.

Ở bước 113, bộ giải mã thực hiện giải mã khối. Cụ thể là, bộ giải mã sử dụng các phép biến đổi ngược để tạo các khối dư. Sau đó, bộ giải mã sử dụng các khối dư và các khối dự báo tương ứng để tái tạo các khối ảnh theo phân vùng. Các khối dự báo có thể bao gồm cả các khối dự báo trong và các khối dự báo ngoài như được tạo ở bộ mã hóa ở bước 105. Các khối ảnh được tái tạo are then được định vị thành các khung của tín hiệu video được tái tạo theo dữ liệu phân vùng được xác định ở bước 111. Cú pháp cho bước 113 có thể cũng được báo hiệu trong dòng bit qua tạo mã entropy như nêu trên.

Ở bước 115, thực hiện lọc trên các khung của tín hiệu video được tái tạo theo cách tương tự với bước 107 ở bộ mã hóa. Chẳng hạn, các bộ lọc chặn tạp âm, các bộ lọc tách khối, ALF, và các bộ lọc SAO có thể được áp dụng cho các khung để loại bỏ các giả tượng chặn khối. Khi các khung được lọc, tín hiệu video có thể được xuất ra bộ phận hiển thị ở bước 117 để nhìn bởi người dùng cuối.

Fig.2 là sơ đồ của hệ thống mã hóa và giải mã (codec) lấy làm ví dụ 200 để tạo mã video. Cụ thể là, hệ thống codec 200 cung cấp chức năng để hỗ trợ triển khai phương pháp hoạt động 100. Hệ thống codec 200 được tổng hợp để mô tả các thành phần được sử dụng trong cả bộ mã hóa và bộ giải mã. Hệ thống codec 200 nhận và phân vùng tín hiệu video như được nêu dựa vào các bước 101 và 103 theo phương pháp hoạt động 100, dẫn đến tín hiệu video được phân vùng 201. Sau đó, hệ thống codec 200 nén tín hiệu video được phân vùng 201 thành dòng bit được tạo mã khi hoạt động như là bộ mã hóa như được nêu dựa vào các bước 105, 107, và 109 trong phương pháp 100. Khi hoạt động như là bộ giải mã, hệ thống codec 200 tạo tín hiệu video đầu ra từ dòng bit như được nêu dựa vào các bước 111, 113, 115, và 117 theo phương pháp hoạt động 100. Hệ thống codec 200 bao gồm thành phần điều khiển bộ tạo mã tổng quát 211, thành phần lượng tử và chia tỷ lệ biến đổi 213, thành phần ước tính trong ảnh 215, thành phần dự báo trong ảnh 217, thành phần bù chuyển động 219, thành phần ước tính chuyển động 221, thành phần chia tỷ lệ và biến đổi ngược 229, thành phần phân tích điều khiển lọc 227, thành phần bộ lọc trong vòng 225, thành phần bộ đệm ảnh được giải mã 223, và thành phần định dạng tiêu đề và tạo mã số học nhị phân ngữ cảnh thích ứng (context adaptive binary arithmetic coding, CABAC) 231. Các thành phần này được ghép nối như được thể hiện trên hình vẽ. Trên Fig.2, các đường màu đen chỉ báo dữ liệu di chuyển cần được mã hóa/giải mã trong khi các đường nét đứt chỉ báo dữ liệu điều khiển di chuyển mà điều khiển hoạt động của các thành phần khác. Các thành phần của hệ thống codec 200 có thể đều xuất hiện trong bộ mã hóa. Bộ giải mã có thể bao gồm tập con của các thành phần của hệ thống codec 200. Chẳng hạn, bộ giải mã có thể bao gồm thành phần dự báo trong ảnh 217, thành phần bù chuyển động 219, thành phần chia tỷ lệ và biến đổi ngược 229, thành phần bộ lọc trong vòng 225, và thành phần bộ đệm ảnh được giải mã 223. Các thành phần này đang được mô tả.

Tín hiệu video được phân vùng 201 là chuỗi video được ghi nhận đã được phân vùng thành các khối của các pixel by cây mã. Cây mã sử dụng các chế độ tách khác nhau để phân chia phụ khối của các pixel thành các khối nhỏ hơn của các pixel. Sau đó, các khối này có thể còn được phân chia phụ thành các khối

nhỏ hơn. Các khối có thể được gọi là các nút trên cây mã. Các nút cha mẹ lớn hơn được tách thành các nút con nhỏ hơn. Số lần nút được phân chia phụ được gọi là chiều sâu của nút/cây mã. Các khối được phân chia có thể được bao gồm trong các CU trong một số trường hợp. Chẳng hạn, CU có thể là phần phụ của CTU mà chứa khối độ sáng, (các) khối sắc độ lệch đỏ (Cr), và (các) khối sắc độ lệch lam (Cb) cùng với các lệnh cú pháp tương ứng cho CU. Các chế độ tách có thể bao gồm cây nhị phân (binary tree, BT), cây tam phân (triple tree, TT), và cây tứ phân (quad tree, QT) lần lượt được sử dụng để phân vùng nút thành hai, ba, hoặc bốn nút con, có các hình dạng khác nhau phụ thuộc vào các chế độ tách được sử dụng. Tín hiệu video được phân vùng 201 được chuyển tiếp về thành phần điều khiển bộ tạo mã tổng quát 211, thành phần lượng tử và chia tỷ lệ biến đổi 213, thành phần ước tính trong ảnh 215, thành phần phân tích điều khiển lọc 227, và thành phần ước tính chuyển động 221 để nén.

Thành phần điều khiển bộ tạo mã tổng quát 211 được tạo cấu hình để ra quyết định liên quan đến việc tạo mã các ảnh của chuỗi video thành dòng bit theo các ràng buộc ứng dụng. Chẳng hạn, thành phần điều khiển bộ tạo mã tổng quát 211 quản lý tối ưu tốc độ bit/kích thước dòng bit với chất lượng tái tạo. Các quyết định này có thể được thực hiện dựa trên không gian lưu trữ/tính khả dụng băng thông và các yêu cầu độ phân giải ảnh. Thành phần điều khiển bộ tạo mã tổng quát 211 cũng quản lý tận dụng bộ đệm theo tốc độ truyền để giảm bớt vấn đề chạy quá mức và chạy dưới mức của bộ đệm. Để quản lý các vấn đề này, thành phần điều khiển bộ tạo mã tổng quát 211 quản lý phân vùng, dự báo, và lọc bởi các thành phần khác. Chẳng hạn, thành phần điều khiển bộ tạo mã tổng quát 211 có thể tăng động độ phức tạp nén để tăng độ phân giải và tăng sử dụng băng thông hoặc giảm độ phức tạp nén để giảm độ phân giải và sử dụng băng thông. Do vậy, thành phần điều khiển bộ tạo mã tổng quát 211 điều khiển các thành phần khác của hệ thống codec 200 để cân bằng chất lượng tái tạo tín hiệu video với các quan tâm tốc độ bit. Thành phần điều khiển bộ tạo mã tổng quát 211 tạo dữ liệu điều khiển, điều khiển hoạt động của các thành phần khác. Dữ liệu điều khiển cũng được chuyển tiếp đến thành phần định dạng tiêu đề và CABAC 231 cần được mã hóa trong dòng bit để báo hiệu các tham số để giải mã ở bộ giải mã.

Tín hiệu video được phân vùng 201 cũng được gửi đến thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219 để dự báo ngoài. Khung hoặc lát của tín hiệu video được phân vùng 201 có thể được phân chia thành nhiều khối video. Thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219 thực hiện tạo mã dự báo ngoài khối video được nhận liên quan đến một hoặc nhiều khối trong một hoặc nhiều khung tham chiếu để dự báo thời gian. Hệ thống codec 200 có thể thực hiện nhiều bước tạo mã, chẳng hạn, để chọn chế độ mã thích hợp đối với mỗi khối dữ liệu video.

Thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219 có thể được tích hợp chặt chẽ, nhưng được minh họa riêng rẽ cho các khái niệm. Ước tính chuyển động, được thực hiện bởi thành phần ước tính chuyển động 221, là quá trình tạo các vector chuyển động, ước tính chuyển động cho các khối video. Vector chuyển động, chẳng hạn, có thể chỉ báo đối tượng được tạo mã xê dịch so với khối dự báo. Khối dự báo là khối được tìm thấy so khớp nhất với khối cần được tạo mã, theo hiệu số pixel. Khối dự báo có thể cũng được gọi là khối tham chiếu. Hiệu số pixel này có thể được xác định bằng tổng hiệu số tuyệt đối (sum of absolute difference, SAD), tổng hiệu số bình phương (sum of square difference, SSD), hoặc các hệ đo độ lệch khác. HEVC sử dụng vài đối tượng được tạo mã bao gồm CTU, CTB, và CU. Chẳng hạn, CTU có thể được phân chia thành các CTB, mà sau đó có thể được phân chia thành các CB để bao gồm trong các CU. CU có thể được mã hóa dưới dạng đơn vị dự báo (PU) chứa dữ liệu dự báo và/hoặc đơn vị biến đổi (transform unit, TU) chứa dữ liệu dư được biến đổi cho CU. Thành phần ước tính chuyển động 221 tạo các vector chuyển động, các PU, và các TU nhờ sử dụng phân tích méo dạng tốc độ như là một phần của quá trình tối ưu hóa méo dạng tốc độ. Chẳng hạn, thành phần ước tính chuyển động 221 có thể xác định nhiều khối tham chiếu, nhiều vector chuyển động, v.v. cho khối/khung hiện tại, và có thể chọn các khối tham chiếu, các vector chuyển động, v.v. có các đặc tính méo dạng tốc độ tốt nhất. Các đặc tính méo dạng tốc độ tốt nhất cân bằng cả hai chất lượng tái tạo video (chẳng hạn, lượng tổn hao dữ liệu bằng cách nén) với hiệu suất tạo mã (chẳng hạn, kích thước mã hóa cuối cùng).

Trong một số ví dụ, hệ thống codec 200 có thể tính toán các giá trị cho các vị trí pixel con số nguyên của các ảnh tham chiếu được lưu trữ trong thành phần DPB 223. Chẳng hạn, hệ thống codec video 200 có thể nội suy các giá trị của các vị trí pixel 1/4, các vị trí pixel 1/8, hoặc các vị trí pixel phân số khác của ảnh tham chiếu. Do vậy, thành phần ước tính chuyển động 221 có thể thực hiện tìm kiếm chuyển động tương đối với các vị trí pixel nguyên và các vị trí pixel phân số và xuất ra vector chuyển động có độ chính xác pixel phân số. Thành phần ước tính chuyển động 221 tính toán vector chuyển động đối với PU của khối video trong lát được tạo mã ngoài bằng cách so sánh vị trí của PU với vị trí của khối ảnh tham chiếu dự báo. Thành phần ước tính chuyển động 221 xuất ra vector chuyển động được tính toán dưới dạng dữ liệu chuyển động ra thành phần định dạng tiêu đề và CABAC 231 để mã hóa và chuyển động đến thành phần bù chuyển động 219.

Bù chuyển động, được thực hiện bởi thành phần bù chuyển động 219, có thể bao gồm nạp hoặc tạo khối dự báo dựa trên vector chuyển động được xác định bằng thành phần ước tính chuyển động 221. Thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219 lại có thể lại được tích hợp về mặt chức năng, trong một số ví dụ. Khi nhận vector chuyển động đối với PU của khối video hiện tại, thành phần bù chuyển động 219 có thể định vị khối dự báo mà vector chuyển động trỏ đến. Khối video dư sau đó được tạo bằng cách lấy các giá trị pixel của khối video hiện tại đang được tạo mã trừ đi các giá trị pixel của khối dự báo, tạo hiệu số các giá trị pixel. Nói chung, thành phần ước tính chuyển động 221 thực hiện ước tính chuyển động so với các thành phần độ sáng, và thành phần bù chuyển động 219 sử dụng các vector chuyển động được tính toán dựa trên các thành phần độ sáng cho cả các thành phần sắc độ và các thành phần độ sáng. Khối dự báo và khối dư được chuyển tiếp đến thành phần lượng tử và chia tỷ lệ biến đổi 213.

Tín hiệu video được phân vùng 201 cũng được gửi đến thành phần ước tính trong ảnh 215 và thành phần dự báo trong ảnh 217. Như với thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219, thành phần ước tính trong ảnh 215 và thành phần dự báo trong ảnh 217 có thể được tích hợp chặt chẽ, nhưng

được minh họa riêng rẽ cho các khái niệm. Thành phần ước tính trong ảnh 215 và thành phần dự báo trong ảnh 217 dự báo trong khối hiện tại so với các khối trong khung hiện tại, như là tùy chọn với dự báo ngoài được thực hiện bằng thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219 giữa các khung, như nêu trên. Cụ thể là, thành phần ước tính trong ảnh 215 xác định chế độ dự báo trong để sử dụng để mã hóa khối hiện tại. Trong một số ví dụ, thành phần ước tính trong ảnh 215 lựa chọn chế độ dự báo trong thích hợp để mã hóa khối hiện tại từ nhiều chế độ dự báo trong được kiểm tra. Sau đó, các chế độ dự báo trong được chọn được chuyển tiếp về thành phần định dạng tiêu đề và CABAC 231 để mã hóa.

Chẳng hạn, thành phần ước tính trong ảnh 215 tính toán các giá trị méo dạng tốc độ nhờ sử dụng phân tích méo dạng tốc độ cho các chế độ dự báo trong được kiểm tra khác nhau, và lựa chọn chế độ dự báo trong có các đặc tính méo dạng tốc độ tốt nhất trong số các chế độ được kiểm tra. Phân tích méo dạng tốc độ thường xác định lượng méo dạng (hoặc sai số) giữa khối được mã hóa và khối không được mã hóa ban đầu được mã hóa để tạo khối được mã hóa, cũng như tốc độ bit (chẳng hạn, số lượng bit) được sử dụng để tạo khối được mã hóa. Thành phần ước tính trong ảnh 215 tính toán các tỷ lệ từ các méo dạng và các tỷ lệ cho các khối được mã hóa khác nhau để xác định chế độ dự báo trong nào cho giá trị méo dạng tốc độ tốt nhất cho khối. Ngoài ra, thành phần ước tính trong ảnh 215 có thể được tạo cấu hình để tạo mã các khối chiều sâu của bản đồ chiều sâu nhờ sử dụng chế độ tạo mẫu hình chiều sâu (depth modeling mode, DMM) dựa trên tối ưu hóa méo dạng tốc độ (rate-distortion optimization, RDO).

Thành phần dự báo trong ảnh 217 có thể tạo khối dư từ khối dự báo dựa trên các chế độ dự báo trong được chọn được xác định bằng thành phần ước tính trong ảnh 215 khi được triển khai trên bộ mã hóa hoặc đọc khối dư từ dòng bit khi được triển khai trên bộ giải mã. Khối dư bao gồm hiệu số trong các giá trị giữa khối dự báo và khối ban đầu, được biểu diễn dưới dạng ma trận. Khối dư sau đó được chuyển tiếp đến thành phần lượng tử và chia tỷ lệ biến đổi 213. Thành phần ước tính trong ảnh 215 và thành phần dự báo trong ảnh 217 có thể hoạt động trên cả các thành phần độ sáng và sắc độ.

Thành phần lượng tử và chia tỷ lệ biến đổi 213 được tạo cấu hình để nén tiếp khối dư. Thành phần lượng tử và chia tỷ lệ biến đổi 213 áp dụng biến đổi, chẳng hạn, biến đổi cosin rời rạc (discrete cosine transform, DCT), biến đổi sin rời rạc (discrete sine transform, DST), hoặc biến đổi tương tự về mặt khái niệm, cho khối dư, tạo khối video bao gồm các giá trị hệ số biến đổi dư. Các biến đổi wavelet, biến đổi số nguyên, biến đổi băng thành phần hoặc các loại biến đổi khác cũng có thể được sử dụng. Phép biến đổi có thể biến đổi thông tin dư từ miền giá trị pixel đến miền biến đổi, chẳng hạn miền tần số. Thành phần lượng tử và chia tỷ lệ biến đổi 213 cũng được tạo cấu hình để chia tỷ lệ thông tin dư được biến đổi, chẳng hạn dựa trên tần số. Phép chia tỷ lệ này bao gồm việc áp dụng hệ số tỷ lệ cho thông tin dư sao cho thông tin tần số khác nhau được lượng tử ở các độ hạt khác nhau, có thể ảnh hưởng chất lượng hình ảnh của video được tái tạo. Thành phần lượng tử và chia tỷ lệ biến đổi 213 cũng được tạo cấu hình để lượng tử các hệ số biến đổi để giảm tiếp tốc độ bit. Quá trình lượng tử có thể giảm chiều sâu bit được liên kết với một số hoặc tất cả các hệ số. Mức độ lượng tử có thể được chỉnh sửa bằng cách điều chỉnh tham số lượng tử (quantization parameter, QP). Sau đó, trong một số ví dụ, thành phần lượng tử và chia tỷ lệ biến đổi 213 có thể thực hiện quét ma trận bao gồm các hệ số biến đổi được lượng tử. Các hệ số biến đổi được lượng tử được chuyển tiếp đến thành phần định dạng tiêu đề và CABAC 231 cần được mã hóa trong dòng bit.

Thành phần chia tỷ lệ và biến đổi ngược 229 thực hiện hoạt động ngược lại của thành phần lượng tử và chia tỷ lệ biến đổi 213 để hỗ trợ ước tính chuyển động. Thành phần chia tỷ lệ và biến đổi ngược 229 thực hiện chia tỷ lệ ngược, biến đổi, và/hoặc lượng tử để tái tạo khối dư trong miền pixel, chẳng hạn, để sử dụng sau này làm khối tham chiếu mà có thể trở thành khối dự báo cho khối hiện tại khác. Thành phần ước tính chuyển động 221 và/hoặc thành phần bù chuyển động 219 có thể tính toán khối tham chiếu bằng cách bổ sung khối dư lại vào khối dự báo tương ứng để sử dụng trong ước tính chuyển động của khối/khung sau này. Các bộ lọc được áp dụng cho các khối tham chiếu được tái tạo để di chuyển các giả tượng được tạo trong khi chia tỷ lệ, lượng tử, và biến đổi. Ngược

lại, các giả tượng này sẽ dự báo không chính xác (và tạo các giả tượng bổ sung) khi các khối tiếp theo được tạo.

Thành phần phân tích điều khiển lọc 227 và thành phần bộ lọc trong vòng 225 áp dụng các bộ lọc cho các khối dư và/hoặc cho các khối ảnh được tái tạo. Chẳng hạn, khối dư được biến đổi từ thành phần chia tỷ lệ và biến đổi ngược 229 có thể được kết hợp với khối dự báo tương ứng từ thành phần dự báo trong ảnh 217 và/hoặc thành phần bù chuyển động 219 để tái tạo khối ảnh gốc. Sau đó, các bộ lọc có thể được áp dụng cho khối ảnh được tái tạo. Thay vào đó, trong một số ví dụ, các bộ lọc có thể được áp dụng cho các khối dư. Như với các thành phần khác trên Fig.2, thành phần phân tích điều khiển lọc 227 và thành phần bộ lọc trong vòng 225 được tích hợp chặt chẽ và có thể được thực hiện cùng nhau, nhưng được mô tả riêng rẽ cho các khái niệm. Các bộ lọc được áp dụng cho các khối tham chiếu được tái tạo được áp dụng cho các vùng không gian cụ thể và bao gồm nhiều tham số để điều chỉnh cách thức các bộ lọc này được áp dụng. Thành phần phân tích điều khiển lọc 227 phân tích các khối tham chiếu được tái tạo để xác định trong đó các bộ lọc này cần được áp dụng và thiết lập các tham số tương ứng. Dữ liệu này được chuyển tiếp về thành phần định dạng tiêu đề và CABAC 231 làm dữ liệu điều khiển lọc để mã hóa. Thành phần bộ lọc trong vòng 225 áp dụng các bộ lọc này dựa trên dữ liệu điều khiển lọc. Các bộ lọc có thể bao gồm DBF, bộ lọc chặn nhiễu, bộ lọc SAO, và ALF. Các bộ lọc này có thể được áp dụng trong miền không gian/pixel (chẳng hạn, trên khối pixel được tái tạo) hoặc trong miền tần số, phụ thuộc vào ví dụ.

Khi hoạt động như là bộ mã hóa, khối ảnh được tái tạo được lọc, khối dư, và/hoặc khối dự báo được lưu trữ trong thành phần bộ đệm ảnh được giải mã 223 để sử dụng sau này khi ước tính chuyển động như nêu trên. Khi hoạt động như là bộ giải mã, thành phần bộ đệm ảnh được giải mã 223 lưu trữ và chuyển tiếp các khối được tái tạo và được lọc về phía bộ phận hiển thị như là một phần của tín hiệu video đầu ra. Thành phần bộ đệm ảnh được giải mã 223 có thể có thể là thiết bị nhớ bất kỳ có thể lưu trữ các khối dự báo, các khối dư, và/hoặc các khối ảnh được tái tạo.

Thành phần định dạng tiêu đề và CABAC 231 nhận dữ liệu từ các thành phần khác nhau của hệ thống codec 200 và mã hóa dữ liệu này thành dòng bit được tạo mã để truyền về phía bộ giải mã. Cụ thể là, thành phần định dạng tiêu đề và CABAC 231 tạo các tiêu đề khác nhau để mã hóa dữ liệu điều khiển, chẳng hạn dữ liệu điều khiển chung và dữ liệu điều khiển lọc. Ngoài ra, dữ liệu dự báo, bao gồm dữ liệu dự báo trong và chuyển động, cũng như dữ liệu dư ở dạng dữ liệu hệ số biến đổi được lượng tử đều được mã hóa trong dòng bit. Dòng bit cuối cùng bao gồm tất cả thông tin được mong muốn bởi bộ giải mã để tái tạo tín hiệu video được phân vùng ban đầu 201. Thông tin này có thể cũng bao gồm các bảng chỉ số chế độ dự báo trong (cũng được gọi là các bảng ánh xạ từ mã), các định nghĩa ngữ cảnh mã hóa cho các khối khác nhau, các chỉ báo của các chế độ dự báo trong có xác suất lớn nhất, chỉ báo thông tin phân vùng, v.v.. Dữ liệu này có thể được mã hóa nhờ sử dụng tạo mã entropy. Chẳng hạn, thông tin có thể được mã hóa nhờ sử dụng tạo mã chiều dài biến thiên ngữ cảnh thích ứng (context adaptive variable length coding, CAVLC), CABAC, CABAC dựa trên cú pháp (syntax-based context-adaptive binary arithmetic coding, SBAC), tạo mã entropy phân vùng khoảng xác suất (probability interval partitioning entropy, PIPE), hoặc kỹ thuật tạo mã entropy khác. Tiếp theo kỹ thuật tạo mã entropy, dòng bit được tạo mã có thể được truyền đến thiết bị khác (chẳng hạn, bộ giải mã video) hoặc được lưu trữ để truyền hoặc truy xuất sau này.

Fig.3 là sơ đồ khối minh họa bộ mã hóa video ví dụ 300. Bộ mã hóa video 300 có thể được sử dụng để thực hiện các chức năng mã hóa của hệ thống codec 200 và/hoặc thực hiện các bước 101, 103, 105, 107, và/hoặc 109 của phương pháp hoạt động 100. Bộ mã hóa 300 phân vùng tín hiệu video đầu vào, thu được tín hiệu video được phân vùng 301, gần như tương tự với tín hiệu video được phân vùng 201. Sau đó, tín hiệu video được phân vùng 301 được nén và được mã hóa thành dòng bit bởi các thành phần của bộ mã hóa 300.

Cụ thể là, tín hiệu video được phân vùng 301 được chuyển tiếp về thành phần dự báo trong ảnh 317 để dự báo trong. Thành phần dự báo trong ảnh 317 có thể gần như tương tự với thành phần ước tính trong ảnh 215 và thành phần dự báo trong ảnh 217. Tín hiệu video được phân vùng 301 cũng được chuyển tiếp đến

thành phần bù chuyển động 321 để dự báo ngoài dựa trên các khối tham chiếu trong thành phần bộ đệm ảnh được giải mã 323. Thành phần bù chuyển động 321 có thể gần như tương tự với thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219. Các khối dự báo và các khối dư từ thành phần dự báo trong ảnh 317 và thành phần bù chuyển động 321 được chuyển tiếp đến thành phần biến đổi và lượng tử 313 để biến đổi và lượng tử các khối dư. Thành phần biến đổi và lượng tử 313 có thể gần như tương tự với thành phần lượng tử và chia tỷ lệ biến đổi 213. Các khối dư được biến đổi và lượng tử và các khối dự báo tương ứng (cùng với dữ liệu điều khiển liên kết) được chuyển tiếp đến thành phần tạo mã entropy 331 để tạo mã vào dòng bit. Thành phần tạo mã entropy 331 có thể gần như tương tự với thành phần định dạng tiêu đề và CABAC 231.

Các khối dư được biến đổi và lượng tử và/hoặc các khối dự báo tương ứng cũng được chuyển tiếp từ thành phần biến đổi và lượng tử 313 sang thành phần biến đổi ngược và lượng tử 329 để tái tạo thành các khối tham chiếu để sử dụng bằng thành phần bù chuyển động 321. Thành phần biến đổi ngược và lượng tử 329 có thể gần như tương tự với thành phần chia tỷ lệ và biến đổi ngược 229. Các bộ lọc trong vòng trong thành phần bộ lọc trong vòng 325 cũng được áp dụng cho các khối dư và/hoặc các khối tham chiếu được tái tạo, phụ thuộc vào ví dụ. Thành phần bộ lọc trong vòng 325 có thể gần như tương tự với thành phần phân tích điều khiển lọc 227 và thành phần bộ lọc trong vòng 225. Thành phần bộ lọc trong vòng 325 có thể bao gồm nhiều bộ lọc như được nêu dựa vào thành phần lọc trong vòng 225. Sau đó, các khối được lọc được lưu trữ trong thành phần bộ đệm ảnh được giải mã 323 để sử dụng làm các khối tham chiếu bằng thành phần bù chuyển động 321. Thành phần bộ đệm ảnh được giải mã 323 có thể gần như tương tự với thành phần bộ đệm ảnh được giải mã 223.

Fig.4 là sơ đồ khối minh họa bộ giải mã video ví dụ 400. Bộ giải mã video 400 có thể được sử dụng để thực hiện các chức năng giải mã của hệ thống codec 200 và/hoặc thực hiện các bước 111, 113, 115, và/hoặc 117 của phương pháp hoạt động 100. Bộ giải mã 400 nhận dòng bit, chẳng hạn từ bộ mã hóa 300, và tạo tín hiệu video đầu ra được tái tạo dựa trên dòng bit để hiển thị đến người dùng cuối.

Dòng bit được nhận bởi thành phần giải mã entropy 433. Thành phần giải mã entropy 433 được tạo cấu hình để thực hiện phương tiện giải mã entropy, chẳng hạn CAVLC, CABAC, SBAC, tạo mã PIPE, hoặc các kỹ thuật tạo mã entropy khác. Chẳng hạn, thành phần giải mã entropy 433 có thể sử dụng thông tin tiêu đề để tạo ngữ cảnh để thông dịch dữ liệu bổ sung được mã hóa dưới dạng các từ mã trong dòng bit. Thông tin được giải mã bao gồm bất kỳ thông tin mong muốn nào để giải mã tín hiệu video, chẳng hạn dữ liệu điều khiển chung, dữ liệu điều khiển lọc, thông tin phân vùng, dữ liệu chuyển động, dữ liệu dự báo, và các hệ số biến đổi được lượng tử từ các khối dư. Các hệ số biến đổi được lượng tử được chuyển tiếp đến thành phần biến đổi ngược và lượng tử 429 để tái tạo thành các khối dư. Thành phần biến đổi ngược và lượng tử 429 có thể giống như thành phần biến đổi ngược và lượng tử 329.

Các khối dư được tái tạo và/hoặc các khối dự báo được chuyển tiếp đến thành phần dự báo trong ảnh 417 để tái tạo thành các khối ảnh dựa trên các hoạt động dự báo trong. Thành phần dự báo trong ảnh 417 có thể giống như thành phần ước tính trong ảnh 215 và thành phần dự báo trong ảnh 217. Cụ thể là, thành phần dự báo trong ảnh 417 sử dụng các chế độ dự báo để định vị khối tham chiếu trong khung và áp dụng khối dư cho kết quả để tái tạo các khối ảnh được dự báo trong. Các khối ảnh được dự báo trong được tái tạo và/hoặc các khối dư và dữ liệu dự báo ngoài tương ứng được chuyển tiếp đến thành phần bộ đệm ảnh được giải mã 423 qua thành phần bộ lọc trong vòng 425, có thể lần lượt gần như tương tự với thành phần DPB 223 và thành phần lọc trong vòng 225. Thành phần bộ lọc trong vòng 425 lọc các khối ảnh được tái tạo, các khối dư và/hoặc các khối dự báo, và thông tin này được lưu trữ trong thành phần bộ đệm ảnh được giải mã 423. Các khối ảnh được tái tạo từ thành phần DPB 423 được chuyển tiếp đến thành phần bù chuyển động 421 để dự báo ngoài. Thành phần bù chuyển động 421 có thể gần như tương tự với thành phần ước tính chuyển động 221 và/hoặc thành phần bù chuyển động 219. Cụ thể là, thành phần bù chuyển động 421 sử dụng các vector chuyển động từ khối tham chiếu để tạo khối dự báo và áp dụng khối dư cho kết quả để tái tạo khối ảnh. Các khối được tái tạo thu được có thể cũng được chuyển tiếp qua thành phần bộ lọc trong vòng 425 đến thành phần bộ đệm ảnh

được giải mã 423. Thành phần bộ đệm ảnh được giải mã 423 tiếp tục lưu trữ các khối ảnh được tái tạo bổ sung, mà có thể được tái tạo thành các khung qua thông tin phân vùng. Các khung này có thể cũng được đặt trong chuỗi. Chuỗi này được xuất ra về phía bộ phận hiển thị làm tín hiệu video đầu ra được tái tạo.

Fig.5 là sơ đồ minh họa chuỗi video nhiều lớp lấy làm ví dụ 500 được tạo cấu hình cho dự báo ngoài lớp 521. Chuỗi video nhiều lớp 500 có thể được mã hóa bởi bộ mã hóa, chẳng hạn hệ thống codec 200 và/hoặc bộ mã hóa 300 và được giải mã bởi bộ giải mã, chẳng hạn, hệ thống codec 200 và/hoặc bộ giải mã 400, chẳng hạn theo phương pháp 100. Chuỗi video nhiều lớp 500 được bao gồm để mô tả ứng dụng ví dụ cho các lớp trong CVS. Chuỗi video nhiều lớp 500 là chuỗi video bất kỳ sử dụng các lớp, chẳng hạn lớp N 531 và lớp N+1 532.

Trong ví dụ, chuỗi video nhiều lớp 500 có thể sử dụng dự báo ngoài lớp 521. Dự báo ngoài lớp 521 được áp dụng giữa các ảnh 511, 512, 513, và 514 và các ảnh 515, 516, 517, và 518 trong các lớp khác nhau. Trong ví dụ được thể hiện, các ảnh 511, 512, 513, và 514 là một phần của lớp N+1 532 và các ảnh 515, 516, 517, và 518 là một phần của lớp N 531. Lớp, chẳng hạn lớp N 531 và/hoặc lớp N+1 532, là nhóm ảnh đều được liên kết với giá trị đặc tính tương tự, chẳng hạn kích thước tương tự, chất lượng, độ phân giải, SNR, khả năng, v.v.. Lớp có thể được định nghĩa chính thức dưới dạng tập hợp của các đơn vị NAL VCL và các đơn vị NAL phi VCL liên kết chia sẻ cùng `nuh_layer_id`. Đơn vị NAL VCL là đơn vị NAL được tạo mã để chứa dữ liệu video, chẳng hạn lát ảnh được tạo mã. Đơn vị NAL phi VCL là đơn vị NAL chứa dữ liệu không phải video, chẳng hạn, cú pháp và/hoặc các tham số hỗ trợ giải mã dữ liệu video, hiệu năng kiểm tra tính tương hợp, hoặc các hoạt động khác.

Trong ví dụ được thể hiện, lớp N+1 532 được liên kết với kích thước ảnh lớn hơn lớp N 531. Do đó, các ảnh 511, 512, 513, và 514 trong lớp N+1 532 có kích thước ảnh lớn hơn (chẳng hạn, chiều cao và chiều rộng lớn hơn và do vậy nhiều mẫu hơn) các ảnh 515, 516, 517, và 518 trong lớp N 531 trong ví dụ này. Tuy nhiên, các ảnh này có thể được tách riêng giữa lớp N+1 532 và lớp N 531 bởi các đặc tính khác. Trong khi chỉ hai lớp, lớp N+1 532 và lớp N 531, được thể hiện, tập hợp ảnh có thể được tách riêng thành số lượng lớp bất kỳ dựa trên các

đặc tính liên kết. Lớp N+1 532 và lớp N 531 có thể cũng được ký hiệu bằng ID lớp. ID lớp là mục dữ liệu được liên kết với ảnh và ký hiệu ảnh là một phần của lớp được chỉ báo. Do đó, mỗi ảnh từ 511 đến 518 có thể được liên kết với ID lớp tương ứng để chỉ báo lớp nào trong lớp N+1 532 hoặc lớp N 531 bao gồm ảnh tương ứng. Chẳng hạn, ID lớp có thể bao gồm ID lớp tiêu đề đơn vị NAL (nuh_layer_id), vốn là phần tử cú pháp xác định ID lớp bao gồm đơn vị NAL (chẳng hạn, bao gồm các lát và/hoặc các tham số của các ảnh trong lớp). Lớp được liên kết với chất lượng/kích thước dòng bit thấp hơn, chẳng hạn lớp N 531, thường được gán cho ID lớp thấp hơn và được gọi là lớp thấp hơn. Ngoài ra, lớp được liên kết với chất lượng/kích thước dòng bit cao hơn, chẳng hạn lớp N+1 532, thường được gán ID lớp cao hơn và được gọi là lớp cao hơn.

Các ảnh từ 511 đến 518 trong các lớp khác nhau từ 531 đến 532 được tạo cấu hình để được hiển thị theo tùy chọn. Lấy ví dụ cụ thể, bộ giải mã có thể giải mã và hiển thị ảnh 515 ở thời gian hiển thị hiện tại nếu ảnh nhỏ hơn được mong muốn hoặc bộ giải mã có thể giải mã và hiển thị ảnh 511 ở thời gian hiển thị hiện tại nếu cần ảnh lớn hơn. Thực tế, các ảnh từ 511 đến 514 ở lớp N+1 532 cao hơn chứa dữ liệu ảnh gần như tương tự dưới dạng các ảnh từ 515 đến 518 tương ứng ở lớp N thấp hơn 531 (bất kể khác biệt trong kích thước ảnh). Cụ thể là, ảnh 511 chứa dữ liệu ảnh gần như tương tự với ảnh 515, ảnh 512 chứa dữ liệu ảnh gần như tương tự với ảnh 516, v.v..

Các ảnh từ 511 đến 518 có thể được tạo mã nhờ tham chiếu đến các ảnh khác từ 511 đến 518 trong cùng lớp N 531 hoặc N+1 532. Việc tạo mã ảnh dựa vào ảnh khác trong cùng lớp dẫn đến dự báo ngoài 523. Dự báo ngoài 523 được mô tả bằng các mũi tên nét liền. Chẳng hạn, ảnh 513 có thể được tạo mã nhờ sử dụng dự báo ngoài 523 bằng cách sử dụng một hoặc hai trong các ảnh 511, 512, và/hoặc 514 trong lớp N+1 532 làm tham chiếu, trong đó một ảnh được tạo tham chiếu cho dự báo ngoài đơn hướng và/hoặc hai ảnh được tạo tham chiếu cho dự báo ngoài hai hướng. Ngoài ra, ảnh 517 có thể được tạo mã bằng cách sử dụng dự báo ngoài 523 nhờ sử dụng một hoặc hai ảnh 515, 516, và/hoặc 518 trong lớp N 531 làm tham chiếu, trong đó một ảnh được tạo tham chiếu cho dự báo ngoài đơn hướng và/hoặc hai ảnh được tạo tham chiếu cho dự báo ngoài hai hướng. Khi

ảnh được sử dụng làm tham chiếu cho ảnh khác trong cùng lớp khi thực hiện dự báo ngoài 523, ảnh có thể được gọi là ảnh tham chiếu. Chẳng hạn, ảnh 512 có thể là ảnh tham chiếu được sử dụng để tạo mã ảnh 513 theo dự báo ngoài 523. Dự báo ngoài 523 có thể cũng được gọi là dự báo trong lớp trong ngữ cảnh nhiều lớp. Thực tế, dự báo ngoài 523 là cơ chế tạo mã các mẫu của ảnh hiện tại nhờ tham chiếu đến các mẫu được chỉ báo trong ảnh tham chiếu khác với ảnh hiện tại trong đó ảnh tham chiếu và ảnh hiện tại nằm trong cùng lớp.

Các ảnh từ 511 đến 518 cũng có thể được tạo mã nhờ tham chiếu đến các ảnh khác từ 511 đến 518 trong các lớp khác nhau. Quá trình này được biết đến như là dự báo ngoài lớp 521, và được mô tả bằng các mũi tên nét đứt. Dự báo ngoài lớp 521 là cơ chế tạo mã các mẫu của ảnh hiện tại nhờ tham chiếu đến các mẫu được chỉ báo trong ảnh tham chiếu trong đó ảnh hiện tại và ảnh tham chiếu nằm trong các lớp khác nhau và do vậy có các ID lớp khác nhau. Chẳng hạn, ảnh trong lớp thấp hơn N 531 có thể được sử dụng làm ảnh tham chiếu để tạo mã ảnh tương ứng ở lớp cao hơn N+1 532. Như là ví dụ cụ thể, ảnh 511 có thể được tạo mã nhờ tham chiếu đến ảnh 515 theo dự báo ngoài lớp 521. Trong trường hợp này, ảnh 515 được sử dụng làm ảnh tham chiếu lớp ngoài. Ảnh tham chiếu lớp ngoài là ảnh tham chiếu được sử dụng để dự báo ngoài lớp 521. Trong phần lớn trường hợp, dự báo ngoài lớp 521 bị giới hạn sao cho ảnh hiện tại, chẳng hạn ảnh 511, có thể chỉ sử dụng (các) ảnh tham chiếu lớp ngoài được bao gồm trong cùng AU và nằm ở lớp thấp hơn, chẳng hạn ảnh 515. AU là tập hợp của các ảnh được liên kết với thời gian xuất ra cụ thể trong chuỗi video, và do vậy AU có thể bao gồm nhiều nhất một ảnh trên lớp. Khi nhiều lớp (chẳng hạn, nhiều hơn 2) khả dụng, dự báo ngoài lớp 521 có thể mã hóa/giải mã ảnh hiện tại dựa trên (các) ảnh tham chiếu ngoài lớp ở các mức thấp hơn so với ảnh hiện tại.

Bộ mã hóa video có thể sử dụng chuỗi video nhiều lớp 500 để mã hóa các ảnh từ 511 đến 518 qua nhiều tổ hợp và/hoặc hoán vị khác nhau của dự báo ngoài 523 và dự báo ngoài lớp 521. Chẳng hạn, ảnh 515 có thể được tạo mã theo dự báo trong. Sau đó, các ảnh từ 516 đến 518 có thể được tạo mã theo dự báo ngoài 523 nhờ sử dụng ảnh 515 làm ảnh tham chiếu. Ngoài ra, ảnh 511 có thể được tạo mã theo dự báo ngoài lớp 521 nhờ sử dụng ảnh 515 làm ảnh tham chiếu lớp

ngoài. Sau đó, các ảnh từ 512 đến 514 có thể được tạo mã theo dự báo ngoài 523 nhờ sử dụng ảnh 511 làm ảnh tham chiếu. Thực tế, ảnh tham chiếu có thể dùng làm cả ảnh tham chiếu một lớp lẫn ảnh tham chiếu lớp ngoài cho các cơ chế tạo mã khác nhau. Bằng cách tạo mã các ảnh lớp cao hơn $N+1$ 532 dựa trên các ảnh lớp thấp hơn N 531, lớp cao hơn $N+1$ 532 có thể tránh sử dụng dự báo trong, có hiệu suất tạo mã thấp hơn nhiều so với dự báo ngoài 523 và dự báo ngoài lớp 521. Thực tế, hiệu suất tạo mã kém của dự báo trong có thể bị giới hạn ở các ảnh có chất lượng nhỏ nhất/thấp nhất, và do vậy bị giới hạn ở việc tạo mã lượng dữ liệu video nhỏ nhất. Các ảnh được sử dụng làm các ảnh tham chiếu và/hoặc các ảnh tham chiếu lớp ngoài có thể được chỉ báo trong các mục nhập của (các) danh sách ảnh tham chiếu được chứa trong cấu trúc danh sách ảnh tham chiếu.

Để thực hiện các hoạt động này, các lớp, chẳng hạn, lớp N 531 và lớp $N+1$ 532 có thể được bao gồm trong OLS 525. OLS 525 là tập hợp lớp mà một hoặc nhiều lớp cho nó được xác định làm lớp đầu ra. Lớp đầu ra là lớp được chỉ định cho đầu ra (chẳng hạn, đến bộ phận hiển thị). Chẳng hạn, lớp N 531 có thể được bao gồm chỉ để hỗ trợ dự báo ngoài lớp 521 và có thể không bao giờ được xuất ra. Trong trường hợp này, lớp $N+1$ 532 được giải mã dựa trên lớp N 531 và được xuất ra. Trong trường hợp này, OLS 525 bao gồm lớp $N+1$ 532 làm lớp đầu ra. OLS 525 có thể chứa nhiều lớp trong các tổ hợp khác nhau. Chẳng hạn, lớp đầu ra trong OLS 525 có thể được tạo mã theo dự báo ngoài lớp 521 dựa trên một, hai, hoặc nhiều lớp thấp hơn. Ngoài ra, OLS 525 có thể chứa nhiều hơn một lớp đầu ra. Do vậy, OLS 525 có thể chứa một hoặc nhiều lớp đầu ra và bất kỳ lớp hỗ trợ nào cần để tái tạo các lớp đầu ra. Chuỗi video nhiều lớp 500 có thể được tạo mã nhờ sử dụng nhiều OLS 525 khác nhau mà mỗi OLS sử dụng các tổ hợp khác nhau của các lớp.

Như là ví dụ cụ thể, dự báo ngoài lớp 521, có thể được sử dụng để hỗ trợ tính khả mở. Chẳng hạn, video có thể được tạo mã thành lớp cơ sở, chẳng hạn lớp N 531, và vài lớp tăng cường, chẳng hạn lớp $N+1$ 532, lớp $N+2$, lớp $N+3$, v.v., được tạo mã theo dự báo ngoài lớp 521. Chuỗi video có thể được tạo mã cho vài đặc tính khả mở, chẳng hạn SNR, tốc độ khung, kích thước ảnh, v.v.. Sau đó, OLS 525 có thể được tạo đối với mỗi đặc tính được chấp nhận. Chẳng hạn, OLS

525 đối với độ phân giải thứ nhất có thể bao gồm chỉ lớp N 531, OLS 525 đối với độ phân giải thứ hai có thể bao gồm lớp N 531 và lớp N+1 532, OLS đối với độ phân giải thứ ba có thể bao gồm lớp N 531, lớp N+1 532, lớp N+2, v.v.. Theo cách này, OLS 525 có thể được truyền để cho phép bộ giải mã để giải mã phiên bản nào của chuỗi video nhiều lớp 500 được mong muốn dựa trên các điều kiện mạng, các giới hạn phần cứng, v.v..

Fig.6 là sơ đồ minh họa chuỗi video lấy làm ví dụ 600 với các OLS được tạo cấu hình cho tính khả mở không gian và/hoặc SNR. Chuỗi video 600 là ví dụ cụ thể của chuỗi video nhiều lớp 500. Thực tế, chuỗi video 600 có thể được mã hóa bởi bộ mã hóa, chẳng hạn hệ thống codec 200 và/hoặc bộ mã hóa 300 và được giải mã bởi bộ giải mã, chẳng hạn hệ thống codec 200 và/hoặc bộ giải mã 400, chẳng hạn theo phương pháp 100. Chuỗi video 600 hữu ích cho tính khả mở.

Chuỗi video 600 ví dụ bao gồm OLS 620, OLS 621, và OLS 622, có thể gần như tương tự với OLS 525. Trong khi ba OLS được mô tả, thì số lượng OLS bất kỳ có thể được sử dụng. Mỗi OLS 620, 621, và 622 được tạo tham chiếu bởi chỉ số OLS và bao gồm một hoặc nhiều lớp. Cụ thể là, OLS 620, 621, và 622 lần lượt bao gồm lớp 630, các lớp 630 và 631, và các lớp 630, 631, và 632. Các lớp 630, 631, và 632 có thể gần như tương tự với lớp N 531 và lớp N+1 532. Các lớp 630, 631, và 632 được tạo tham chiếu bởi chỉ số lớp. Chuỗi video 600 bao gồm số lượng lớp tương tự như số lượng OLS. Cụ thể là, OLS 620 có chỉ số OLS thấp nhất chứa lớp 630 có chỉ số lớp thấp nhất. Mỗi OLS khác bao gồm tất cả các lớp OLS trước đó với chỉ số OLS thấp hơn cộng 1. Chẳng hạn, OLS 621 có chỉ số OLS cao hơn OLS 620 và chứa các lớp 630 và 631, vốn là tất cả các lớp OLS 620 cộng 1. Tương tự, OLS 622 có chỉ số OLS cao hơn OLS 621 và chứa các lớp 630, 631, và 632 vốn là tất cả các lớp OLS 621 cộng 1. Mẫu hình này có thể tiếp tục cho đến khi đạt đến lớp có chỉ số lớp cao nhất và OLS có chỉ số OLS cao nhất.

Ngoài ra, lớp 630 là lớp cơ sở. Tất cả các lớp khác 631 và 632 là các lớp tăng cường được tạo mã theo dự báo ngoài lớp dựa trên tất cả các lớp có các chỉ số lớp thấp hơn. Cụ thể là, lớp 630 là lớp cơ sở và không được tạo mã theo dự báo ngoài lớp. Lớp 631 là lớp tăng cường được tạo mã theo dự báo ngoài lớp dựa

trên lớp 630. Ngoài ra, lớp 632 là lớp tăng cường được tạo mã theo dự báo ngoài lớp dựa trên lớp 630 và 631. Kết quả này chính là OLS 620 chứa lớp 630 có SNR chất lượng thấp nhất và/hoặc kích thước ảnh nhỏ nhất. Do OLS 620 không sử dụng bất kỳ dự báo ngoài lớp, OLS 620 có thể được giải mã hoàn toàn không dựa vào bất kỳ lớp nào ngoài 630. OLS 621 chứa lớp 631 có SNR chất lượng và/hoặc kích thước ảnh cao hơn lớp 630, và lớp 631 có thể được giải mã hoàn toàn theo dự báo ngoài lớp do OLS 621 cũng chứa lớp 630. Tương tự, OLS 622 chứa lớp 632 có SNR chất lượng và/hoặc kích thước ảnh cao hơn các lớp 630 và 631, và lớp 632 có thể được giải mã hoàn toàn theo dự báo ngoài lớp do OLS 622 cũng chứa các lớp 630 và 631. Do đó, chuỗi video 600 được tạo mã để mở rộng bất kỳ SNR và/hoặc kích thước ảnh được chọn trước bằng cách gửi OLS 622, 621, hoặc 620 tương ứng đến bộ giải mã. Khi nhiều OLS 622, 621, và 620 hơn được sử dụng, chuỗi video 600 có thể mở rộng đến nhiều chất lượng ảnh SNR hơn và/hoặc kích thước ảnh hơn.

Do đó, chuỗi video 600 có thể hỗ trợ tính khả mở không gian. Tính khả mở không gian cho phép chuỗi video 600 được tạo mã thành các lớp 630, 631, và 632 sao cho các lớp 630, 631, và 632 được đặt vào các OLS 620, 621, và 622 sao cho mỗi OLS 620, 621, và 622 chứa đủ dữ liệu để giải mã chuỗi video 600 đến kích thước màn hình đầu ra tương ứng. Do vậy, tính khả mở không gian có thể bao gồm tập hợp (các) lớp (chẳng hạn, lớp 630) để giải mã video cho màn hình điện thoại thông minh, tập hợp lớp để giải mã video cho màn hình TV lớn (chẳng hạn, các lớp 630, 631, và 632), và các tập hợp lớp (chẳng hạn, các lớp 630 và 631) cho các kích thước màn hình trung bình. Tính khả mở SNR cho phép chuỗi video 600 được tạo mã thành các lớp 630, 631, và 632 sao cho các lớp 630, 631, và 632 được đặt vào các OLS 620, 621, và 622 sao cho mỗi OLS 620, 621, và 622 chứa đủ dữ liệu để giải mã chuỗi video 600 ở SNR khác nhau. Do vậy, tính khả mở SNR có thể bao gồm tập hợp (các) lớp (chẳng hạn, lớp 630) có thể được giải mã cho video chất lượng thấp, video chất lượng cao (chẳng hạn, các lớp 630, 631, và 632), và các chất lượng video trung gian khác nhau (chẳng hạn, các lớp 630 và 631) để hỗ trợ các điều kiện mạng khác nhau.

Sáng chế tạo báo hiệu hiệu quả để cho phép chuỗi video 600 được sử dụng đúng và hiệu quả. Chẳng hạn, chuỗi video 600 có thể được ký hiệu theo phần tử cú pháp `ols_mode_idc`. Chẳng hạn, phần tử cú pháp `ols_mode_idc` có thể nhận diện chuỗi video 600 làm chế độ OLS 0. Do vậy, phần tử cú pháp `ols_mode_idc` có thể được gán bằng 0 và được báo hiệu trong dòng bit để chỉ báo rằng chuỗi video 600 được sử dụng. Do đó, bộ giải mã có thể nhận bất kỳ OLS và có thể xác định, dựa trên `ols_mode_idc`, rằng lớp được nhận có chỉ số lớp cao nhất là lớp đầu ra cho OLS được nhận và lớp đầu ra có thể được giải mã qua dự báo ngoài lớp dựa trên các lớp thấp hơn khác trong OLS.

Fig.7 là sơ đồ minh họa dòng bit lấy làm ví dụ 700 bao gồm các OLS được tạo cấu hình cho tính khả mở. Chẳng hạn, dòng bit 700 có thể được tạo bằng hệ thống codec 200 và/hoặc bộ mã hóa 300 cho giải mã bằng hệ thống codec 200 và/hoặc bộ giải mã 400 theo phương pháp 100. Ngoài ra, dòng bit 700 có thể bao gồm chuỗi video nhiều lớp được tạo mã 500 và/hoặc chuỗi video 600.

Dòng bit 700 bao gồm VPS 711, một hoặc nhiều tập hợp tham số chuỗi (Sequence Parameter Set, SPS) 713, các tập hợp tham số ảnh (picture parameter set, PPS) 715, các tiêu đề lát 717, và dữ liệu ảnh 720. VPS 711 chứa dữ liệu liên quan đến toàn bộ dòng bit 700. Chẳng hạn, VPS 711 có thể chứa dữ liệu liên quan đến các OLS, các lớp, và/hoặc các lớp con được sử dụng trong dòng bit 700. SPS 713 chứa dữ liệu chuỗi chung cho tất cả các ảnh trong CVS được chứa trong dòng bit 700. Chẳng hạn, mỗi lớp có thể chứa một hoặc nhiều CVS, và mỗi CVS có thể tham chiếu SPS 713 đối với các tham số tương ứng. Các tham số trong SPS 713 có thể bao gồm kích thước ảnh, chiều sâu bit, các tham số công cụ mã, các giới hạn tốc độ bit, v.v.. Cần lưu ý rằng, trong khi mỗi chuỗi viện dẫn đến SPS 713, một SPS 713 có thể chứa dữ liệu cho nhiều chuỗi trong một số ví dụ. PPS 715 chứa các tham số áp dụng cho toàn bộ ảnh. Do vậy, mỗi ảnh trong chuỗi video có thể viện dẫn đến PPS 715. Cần lưu ý rằng, trong khi mỗi ảnh viện dẫn đến PPS 715, một PPS 715 có thể chứa dữ liệu cho nhiều ảnh trong một số ví dụ. Chẳng hạn, nhiều ảnh giống nhau có thể được tạo mã theo các tham số giống nhau. Trong trường hợp này, một PPS 715 có thể chứa dữ liệu cho các ảnh

giống nhau này. PPS 715 có thể chỉ báo các công cụ mã khả dụng cho các lát trong các ảnh tương ứng, các QP, các độ lệch, v.v..

Tiêu đề lát 717 chứa các tham số riêng cho mỗi lát 727 trong ảnh 725. Do vậy, có thể có một tiêu đề lát 717 trên lát 727 trong chuỗi video. Tiêu đề lát 717 có thể chứa thông tin loại lát, các POC, các danh sách ảnh tham chiếu, các trọng số dự báo, các mục nhập của ô, các tham số tách khối, v.v.. Cần lưu ý rằng trong một số ví dụ, dòng bit 700 có thể cũng bao gồm tiêu đề ảnh, vốn là cấu trúc cú pháp chứa các tham số áp dụng cho tất cả các lát 727 trong một ảnh. Do vậy, tiêu đề ảnh và tiêu đề lát 717 có thể được sử dụng qua lại trong một số ngữ cảnh. Chẳng hạn, các tham số cụ thể có thể được di chuyển giữa tiêu đề lát 717 và tiêu đề ảnh phụ thuộc vào liệu các tham số này có chung cho tất cả các lát 727 trong ảnh 725 hay không.

Dữ liệu ảnh 720 chứa dữ liệu video được mã hóa theo dự báo ngoài và/hoặc Dự báo trong cũng như dữ liệu dư được biến đổi và được lượng tử tương ứng. Chẳng hạn, dữ liệu ảnh 720 có thể bao gồm các lớp 723 của các ảnh 725. Các lớp 723 có thể được tổ chức thành các OLS 721. OLS 721 có thể gần như tương tự với OLS 525, 620, 621, và/hoặc 622. Cụ thể là, OLS 721 là tập hợp lớp cho 723 mà một hoặc nhiều lớp 723 được xác định dưới dạng (các) lớp đầu ra. Chẳng hạn, dòng bit 700 có thể được tạo mã để bao gồm vài OLS 721 với video được tạo mã ở các độ phân giải khác nhau, các tốc độ khung, các kích thước ảnh 725, v.v.. Khi yêu cầu bằng bộ giải mã, quá trình trích xuất dòng bit thành phần có thể loại bỏ tất cả trừ OLS 721 được yêu cầu từ dòng bit 700. Sau đó, bộ mã hóa có thể truyền dòng bit 700 chứa chỉ OLS 721 được yêu cầu, và do vậy chỉ video thỏa mãn các tiêu chí được yêu cầu, đến bộ giải mã.

Lớp 723 có thể gần như tương tự với lớp N 531, lớp N+1 532, và/hoặc các lớp 631, 632, và/hoặc 633. Nói chung, lớp 723 là tập hợp của các ảnh được mã hóa 725. Lớp 723 có thể được định nghĩa chính thức dưới dạng tập hợp của các đơn vị NAL VCL mà, khi được giải mã, chia sẻ đặc tính cụ thể (chẳng hạn, độ phân giải chung, tốc độ khung, kích thước ảnh, v.v.). Ảnh 725 có thể được tạo mã dưới dạng tập hợp của các đơn vị NAL VCL. Lớp 723 cũng bao gồm các đơn vị NAL phi VCL được liên kết để hỗ trợ giải mã các đơn vị NAL VCL. Các đơn

vị NAL VCL của lớp 723 có thể chia sẻ giá trị cụ thể của `nuh_layer_id`, vốn là ID lớp ví dụ. Lớp 723 có thể là lớp đồng truyền phát được tạo mã không có dự báo ngoài lớp hoặc lớp 723 được tạo mã theo dự báo ngoài lớp dựa trên các lớp khác.

Ảnh 725 là mảng của các mẫu độ sáng và/hoặc mảng của các mẫu sắc độ tạo khung hoặc trường của nó. Chẳng hạn, ảnh 725 có thể là ảnh được tạo mã có thể được xuất ra để hiển thị hoặc được sử dụng để hỗ trợ tạo mã (các) ảnh 725 khác để xuất ra. Ảnh 725 có thể bao gồm tập hợp đơn vị NAL VCL. Ảnh 725 chứa một hoặc nhiều lát 727. Lát 727 có thể được định nghĩa dưới dạng số lượng nguyên các ô hoàn chỉnh hoặc số lượng nguyên các hàng CTU hoàn chỉnh liên tiếp (chẳng hạn, trong ô) của ảnh 725 chỉ được chứa trong một đơn vị NAL, cụ thể là đơn vị NAL VCL. Các lát 727 còn được phân chia thành các CTU và/hoặc các CTB. CTU là nhóm mẫu có kích thước định trước có thể được phân vùng bởi cây mã. CTB là tập con của CTU và chứa các thành phần độ sáng hoặc các thành phần sắc độ của CTU. Các CTU/CTB còn được phân chia thành các CB dựa trên các cây mã. Sau đó, các CB có thể được mã hóa/giải mã theo các cơ chế dự báo.

Sáng chế bao gồm các cơ chế để hỗ trợ tính khả mở không gian và/hoặc SNR, chẳng hạn theo chuỗi video 600. Chẳng hạn, VPS 711 có thể chứa `ols_mode_idc` 735. `ols_mode_idc` 735 là phần tử cú pháp chỉ báo thông tin liên quan đến số lượng OLS 721, các lớp 723 của các OLS 721, và các lớp đầu ra 723 trong các OLS 721. Lớp đầu ra 723 là lớp bất kỳ được chỉ định để xuất ra bởi bộ giải mã ngược với việc được sử dụng dành cho tạo mã dựa trên tham chiếu. `ols_mode_idc` 735 có thể được gán bằng 1 hoặc 2 để tạo mã các loại video khác. `ols_mode_idc` 735 có thể được gán bằng 0 để hỗ trợ tính khả mở không gian và/hoặc SNR. Chẳng hạn, `ols_mode_idc` 735 có thể được gán bằng 0 để chỉ báo rằng tổng số OLS 721 trong chuỗi video bằng tổng số lớp 723 được xác định trong VPS 711, rằng OLS thứ i 721 bao gồm các lớp từ 0 đến i , bao gồm hai giá trị đầu mút, và that đối với mỗi OLS 721 chỉ lớp cao nhất được xuất ra. Chuỗi điều kiện này có thể mô tả chuỗi video 600 với số lượng OLS 721 bất kỳ. Lợi ích sử dụng `ols_mode_idc` 735 chính là `ols_mode_idc` 735 tiết kiệm bit. Bộ giải mã trong hệ

thống ứng dụng nói chung chỉ nhận một OLS. Tuy nhiên, `ols_mode_idc 735` cũng tiết kiệm bit trong các dòng bit được mã hóa mà chứa nhiều OLS, trong số đó nhiều dữ liệu được chia sẻ, do vậy tiết kiệm trong các máy chủ streaming và tiết kiệm băng thông để truyền các dòng bit này.

Trong một số ví dụ, VPS 711 cũng bao gồm các lớp lớn nhất của phần tử cú pháp VPS trừ 1 (`vps_max_layers_minus1 737`). `VPS_max_layers_minus1 737` là phần tử cú pháp báo hiệu số lượng lớp 723 được xác định bởi VPS 711, và do vậy số lượng lớp lớn nhất 723 được phép trong CVS tương ứng trong dòng bit 700. `ols_mode_idc 735` có thể tạo tham chiếu phần tử cú pháp `vps_max_layers_minus1 737`. Chẳng hạn, `ols_mode_idc 735` có thể chỉ báo rằng tổng số OLS 721 bằng số lượng lớp 723 được xác định bởi `vps_max_layers_minus1 737`.

Ngoài ra, VPS 711 có thể bao gồm `each_layer_is_an_ols_flag 733`. `each_layer_is_an_ols_flag 733` là phần tử cú pháp báo hiệu liệu mỗi OLS 721 trong dòng bit 700 chứa một lớp 723. Chẳng hạn, mỗi OLS 721 có thể chứa một lớp đồng phát khi không sử dụng tính khả mở. Do đó, `each_layer_is_an_ols_flag 733` có thể được gán (chẳng hạn, bằng 0) để chỉ báo rằng một hoặc nhiều OLS 721 chứa nhiều hơn một lớp 723 để hỗ trợ tính khả mở. Thực tế, `each_layer_is_an_ols_flag 733` có thể được sử dụng để hỗ trợ tính khả mở. Chẳng hạn, bộ giải mã có thể kiểm tra `each_layer_is_an_ols_flag 733` để xác định rằng một số OLS 721 bao gồm nhiều hơn một lớp 723. Khi `each_layer_is_an_ols_flag 733` được gán bằng 0 và khi `ols_mode_idc 735` được gán bằng 0 (hoặc 1 được sử dụng cho chế độ khác), tổng số OLS (`TotalNumOlss`) có thể được gán bằng `vps_max_layers_minus1 737`. `TotalNumOlss` là biến được sử dụng bằng cả bộ giải mã và bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD) ở bộ mã hóa. `TotalNumOlss` là biến được sử dụng để lưu trữ số lượng OLS 721 dựa trên dữ liệu trong dòng bit 700. Sau đó, `TotalNumOlss` có thể được sử dụng để giải mã ở bộ giải mã hoặc kiểm tra các sai số dòng bit 700 ở HRD ở bộ mã hóa.

VPS 711 có thể cũng bao gồm phần tử cú pháp ID lớp VPS (`vps_layer_id[i]`) 731. `VPS_layer_id[i]` 731 là mảng lưu trữ các ID lớp (chẳng hạn, `nuh_layer_id`) đối với mỗi lớp. Do đó, `VPS_layer_id[i]` 731 chỉ báo ID lớp của lớp thứ *i*.

Bộ giải mã hoặc HRD có thể sử dụng dữ liệu trong VPS 711 để xác định cấu hình của các OLS 721 và các lớp 723. Trong ví dụ cụ thể, số lượng lớp trong OLS thứ *i* (`NumLayersInOls[i]`) và ID lớp trong OLS (`LayerIdInOls[i][j]`) xác định giá trị `nuh_layer_id` của lớp thứ *j* trong OLS thứ *i* được dẫn xuất như sau:

```
NumLayersInOls[0] = 1
LayerIdInOls[0][0] = vps_layer_id[0]
for(i = 1, i < TotalNumOls; i++) {
    if(each_layer_is_an_ols_flag) {
        NumLayersInOls[i] = 1
        LayerIdInOls[i][0] = vps_layer_id[i]
    } else if(ols_mode_idc == 0 || ols_mode_idc == 1) {
        NumLayersInOls[i] = i + 1
        for(j = 0; j < NumLayersInOls[i]; j++)
            LayerIdInOls[i][j] = vps_layer_id[j]
```

trong đó `vps_layer_id[i]` là ID lớp VPS thứ *i*, `TotalNumOls` là tổng số OLS được xác định bởi VPS, và `each_layer_is_an_ols_flag` là cờ mỗi lớp là OLS xác định liệu ít nhất một OLS chứa nhiều hơn một lớp.

Dữ liệu trong VPS 711 có thể được sử dụng để hỗ trợ SNR và/hoặc các lớp có thể mở rộng trong không gian 723. Các lớp 723 có thể được mã hóa và được bao gồm trong các OLS 721. Bộ mã hóa có thể truyền dòng bit 700 bao gồm OLS 721 được yêu cầu và VPS 711 đến bộ giải mã. Sau đó, bộ giải mã có thể sử dụng thông tin trong VPS 711 để giải mã đúng các lớp 723 trong OLS 721. Cách tiếp cận này hỗ trợ hiệu suất mã trong khi hỗ trợ tính khả mở. Cụ thể là, bộ giải mã có thể nhanh chóng xác định số lượng lớp 723 trong OLS 721, xác định lớp đầu ra trong OLS 721, và xác định các lớp 723 cần được giải mã để hỗ trợ giải mã lớp đầu ra theo dự báo ngoài lớp. Do đó, bộ giải mã có thể chỉ nhận các lớp 723 cần để giải mã lớp cao nhất được nhận và lớp cao nhất được nhận có thể được giải mã và được hiển thị. Theo cách này, tổng số lớp 723 được mã hóa có

thể không ảnh hưởng đến quá trình giải mã và một hoặc nhiều sai số như nêu trên có thể được tránh. Thực tế, các cơ chế được bộc lộ làm tăng chức năng của bộ mã hóa và/hoặc bộ giải mã. Ngoài ra, các cơ chế được bộc lộ có thể giảm kích thước dòng bit, và do vậy giảm bộ xử lý, bộ nhớ, và/hoặc tận dụng tài nguyên mạng ở cả bộ mã hóa và bộ giải mã.

Thông tin đứng trước đang được mô tả chi tiết hơn dưới đây. Kỹ thuật tạo mã video được tạo lớp cũng được gọi là tạo mã video khả mở hoặc tạo mã video có tính khả mở. Tính khả mở trong tạo mã video có thể được hỗ trợ nhờ sử dụng các kỹ thuật tạo mã đa lớp. Dòng bit nhiều lớp bao gồm lớp cơ sở (base layer, BL) và một hoặc nhiều các lớp tăng cường (enhancement layer, EL). Ví dụ về tính khả mở bao gồm tính khả mở không gian, chất lượng/tính khả mở SNR, tính khả mở đa khung nhìn, tính khả mở tốc độ khung, v.v.. Khi kỹ thuật tạo mã đa lớp được sử dụng, ảnh hoặc một phần ảnh của nó có thể được tạo mã mà không sử dụng ảnh tham chiếu (Dự báo trong), có thể được tạo mã bằng cách tạo tham chiếu các ảnh tham chiếu that nằm trong cùng lớp (dự báo ngoài), và/hoặc có thể được tạo mã bằng cách tạo tham chiếu các ảnh tham chiếu nằm trong (các) lớp khác (dự báo ngoài lớp). Ảnh tham chiếu được sử dụng để dự báo ngoài lớp của ảnh hiện tại được gọi là ảnh tham chiếu lớp ngoài (Inter-layer Reference picture, ILRP). Fig.5 minh họa ví dụ tạo mã nhiều lớp cho tính khả mở không gian trong đó các ảnh trong các lớp khác nhau có các độ phân giải khác nhau.

Một số họ tạo mã video hỗ trợ tính khả mở trong (các) tiểu sử riêng rẽ từ (các) tiểu sử để tạo mã một lớp. SVC là phần mở rộng khả mở của AVC hỗ trợ các tính khả mở không gian, thời gian, và chất lượng. Đối với SVC, cờ được báo hiệu trong mỗi khối lớn (macroblock, MB) trong các ảnh EL để chỉ báo liệu MB EL có được dự báo nhờ sử dụng khối cùng vị trí từ lớp thấp hơn. Dự báo từ khối cùng vị trí có thể bao gồm kết cấu, các vectơ chuyển động, và/hoặc các chế độ mã. Các cách triển khai của SVC có thể không sử dụng lại trực tiếp cách triển khai AVC không được chỉnh sửa trong thiết kế của nó. Cú pháp MB EL SVC và quá trình giải mã khác với cú pháp AVC và quá trình giải mã.

HEVC khả mở (Scalable HEVC, SHVC) là phần mở rộng của HEVC hỗ trợ các tính khả mở không gian và chất lượng. MV-HEVC là phần mở rộng của

HEVC hỗ trợ tính khả mở nhiều khung nhìn. 3D HEVC (3D-HEVC) là phần mở rộng của HEVC hỗ trợ để tạo mã video 3D được cải tiến hơn và hiệu quả hơn MV-HEVC. Tính khả mở thời gian có thể được bao gồm dưới dạng phần toàn vẹn của codec HEVC một lớp. Trong phần mở rộng nhiều lớp của HEVC, các ảnh được giải mã được sử dụng để dự báo ngoài lớp chỉ đến từ cùng AU và được xử lý dưới dạng các ảnh tham chiếu dài hạn (long term reference picture, LTRP). Các ảnh này là các chỉ số tham chiếu được gán trong (các) danh sách ảnh tham chiếu cùng với các ảnh tham chiếu thời gian khác trong lớp hiện tại. Dự báo ngoài lớp (Interlayer prediction, ILP) được thực hiện ở mức đơn vị dự báo (prediction unit, PU) bằng cách thiết lập giá trị của chỉ số tham chiếu để viện dẫn đến (các) ILRP trong (các) RPL. Tính khả mở không gian lấy mẫu lại ảnh tham chiếu hoặc một phần của nó khi ILRP có độ phân giải không gian khác so với ảnh hiện tại được mã hóa hoặc giải mã. Kỹ thuật lấy mẫu lại ảnh tham chiếu có thể được thực hiện ở mức ảnh hoặc mức CB.

VVC có thể cũng hỗ trợ kỹ thuật tạo mã video được tạo lớp. Dòng bit VVC có thể bao gồm nhiều lớp. Các lớp có thể độc lập với nhau. Chẳng hạn, mỗi lớp có thể được tạo mã không sử dụng dự báo ngoài lớp. Trong trường hợp này, các lớp cũng được gọi là các lớp đồng phát. Trong một số trường hợp, một số lớp được tạo mã nhờ sử dụng ILP. Cờ trong VPS có thể chỉ báo liệu các lớp là các lớp đồng phát hoặc liệu một số lớp có sử dụng ILP hay không. Khi một số lớp sử dụng ILP, mối quan hệ lệ thuộc lớp giữa các lớp cũng được báo hiệu trong VPS. Không giống SHVC và MV-HEVC, VVC có thể không xác định các OLS. OLS bao gồm tập hợp lớp cụ thể, trong đó một hoặc nhiều lớp trong tập hợp lớp được xác định là các lớp đầu ra. Lớp đầu ra là lớp OLS được xuất ra. Theo một số cách triển khai của VVC, chỉ một lớp có thể được chọn để giải mã và được xuất ra khi các lớp là các lớp đồng phát. Theo một số cách triển khai của VVC, toàn bộ dòng bit bao gồm tất cả các lớp được xác định là được giải mã khi bất kỳ lớp nào sử dụng ILP. Ngoài ra, các lớp cụ thể trong số các lớp được xác định là các lớp đầu ra. Các lớp đầu ra có thể được chỉ báo là riêng lớp cao nhất, tất cả các lớp, hoặc lớp cao nhất cộng tập hợp lớp thấp hơn được chỉ báo.

Các khía cạnh trước đó có các vấn đề cụ thể. Trong một số hệ thống tạo mã video khi dự báo ngoài lớp được sử dụng, toàn bộ dòng bit và tất cả các lớp được xác định được giải mã và các lớp cụ thể trong số các lớp được xác định là các lớp đầu ra. Các lớp đầu ra có thể được chỉ báo là riêng lớp cao nhất, tất cả các lớp, hoặc lớp cao nhất cộng tập hợp lớp thấp hơn được chỉ báo. Để đơn giản trong việc mô tả vấn đề, hai lớp có thể được sử dụng với lớp trên sử dụng lớp dưới cho tham chiếu dự báo ngoài lớp. Đối với tính khả mở không gian và/hoặc tính khả mở SNR, hệ thống cần xác định sử dụng chỉ lớp thấp hơn (giải mã và đầu ra của riêng lớp thấp hơn). Hệ thống cũng xác định sử dụng cả hai lớp (giải mã cả hai lớp và xuất ra lớp cao hơn). Không may mắn, điều này là không thể đối với một số hệ thống tạo mã video.

Nói chung, sáng chế mô tả các cách tiếp cận để báo hiệu đơn giản và hiệu quả các OLS cho các tính khả mở không gian và SNR. Các phần mô tả của các kỹ thuật dựa trên VVC bởi JVET của ITU-T và ISO/IEC. Tuy nhiên, các kỹ thuật cũng áp dụng cho việc tạo mã video được tạo lớp dựa trên các đặc tả codec video khác.

Một hoặc nhiều vấn đề nêu trên có thể được giải quyết như sau. Cụ thể là, sáng chế bao gồm phương pháp đơn giản và hiệu quả để báo hiệu các OLS đối với các tính khả mở không gian và SNR. Hệ thống tạo mã video có thể sử dụng VPS để chỉ báo rằng một số lớp sử dụng ILP, rằng tổng số OLS được xác định bởi VPS bằng số lượng lớp, rằng OLS thứ i bao gồm các lớp có các chỉ số lớp từ 0 đến i , bao gồm hai giá trị đầu mút, và đối với mỗi OLS chỉ lớp cao nhất trong OLS được xuất ra.

Cách thực hiện ví dụ của các cơ chế trước đó như sau. Cú pháp VPS lấy làm ví dụ như sau.

video_parameter_set_rbsp() {	Mô tả
vps_video_parameter_set_id	u(4)
vps_max_layers_minus1	u(6)
vps_max_sub_layers_minus1	u(3)
if(vps_max_layers_minus1 > 0)	
vps_all_independent_layers_flag	u(1)
for(i = 0; i <= vps_max_layers_minus1; i++) {	
vps_layer_id[i]	u(6)

if(i > 0 && !vps_all_independent_layers_flag) {	
vps_independent_layer_flag[i]	u(1)
if(!vps_independent_layer_flag[i])	
for(j = 0; j < i; j++)	
vps_direct_dependency_flag[i][j]	u(1)
}	
}	
if(vps_max_layers_minus1 > 0) {	
if(vps_all_independent_layers_flag)	
each_layer_is_an_ols_flag	u(1)
if(!each_layer_is_an_ols_flag) {	
if(!vps_all_independent_layers_flag)	
ols_mode_idc	u(2)
if(ols_mode_idc == 2) {	
num_output_layer_sets_minus1	u(8)
for(i = 1; i < num_output_layer_sets_minus1 + 1; i++) {	
for(j = 0; j <= vps_max_layers_minus1; j++)	
layer_bao_gòmd_flag[i][j]	u(1)
if(!vps_all_independent_layers_flag)	
for(j = 0; j < NumLayersInOls[i] - 1; j++)	
vps_output_layer_flag[i][j]	u(1)
}	
}	
}	
}	
vps_constraint_info_present_flag	u(1)
vps_reserved_zero_7bits	u(7)
if(vps_constraint_info_present_flag)	
general_constraint_info()	
general_hrd_params_present_flag	u(1)
if(general_hrd_params_present_flag) {	
num_units_in_tick	u(32)
time_scale	u(32)
general_hrd_parameters()	
}	
vps_extension_flag	u(1)
if(vps_extension_flag)	
trong khi(more_rbsp_data())	
vps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

Ngữ nghĩa VPS lấy làm ví dụ như sau. RBSP VPS nên khả dụng cho quá trình giải mã trước khi được tạo tham chiếu, cần được bao gồm trong ít nhất một QU có TemporalId bằng 0 hoặc được tạo qua các cơ chế bên ngoài, và đơn vị NAL VPS chứa RBSP VPS nên có nuh_layer_id bằng vps_layer_id[0]. Tất cả các đơn vị NAL VPS có giá trị cụ thể của vps_video_parameter_set_id trong CVS nên có cùng nội dung. vps_video_parameter_set_id tạo ID cho VPS để tham chiếu bởi các phần tử cú pháp khác. vps_max_layers_minus1 cộng 1 xác định số lượng lớp được phép lớn nhất trong mỗi CVS viện dẫn đến VPS. vps_max_sub_layers_minus1 cộng 1 xác định số lượng lớn nhất của các lớp con thời gian có thể xuất hiện trong mỗi CVS viện dẫn đến VPS. Giá trị của vps_max_sub_layers_minus1 sẽ nằm trong khoảng từ 0 đến 6, bao gồm hai giá trị đầu mút.

vps_all_independent_layers_flag có thể được gán bằng 1 để xác định rằng tất cả các lớp trong CVS được tạo mã độc lập mà không sử dụng dự báo ngoài lớp. vps_all_independent_layers_flag có thể được gán bằng 0 để xác định rằng một hoặc nhiều lớp trong CVS có thể use dự báo ngoài lớp. Khi không xuất hiện, giá trị của vps_all_independent_layers_flag được suy ra bằng 1. Khi vps_all_independent_layers_flag bằng 1, giá trị của vps_independent_layer_flag[i] được suy ra bằng 1. Khi vps_all_independent_layers_flag bằng 0, giá trị của vps_independent_layer_flag[0] được suy ra bằng 1. vps_layer_id[i] xác định giá trị nuh_layer_id của lớp thứ i. Đối với hai giá trị nguyên không âm bất kỳ của m và n, khi m nhỏ hơn n, giá trị của vps_layer_id[m] cần nhỏ hơn vps_layer_id[n]. vps_independent_layer_flag[i] có thể được gán bằng 1 để xác định rằng lớp có chỉ số i không sử dụng dự báo ngoài lớp. vps_independent_layer_flag[i] có thể được gán bằng 0 để xác định rằng lớp có chỉ số i có thể sử dụng dự báo ngoài lớp và vps_layer_dependency_flag[i] xuất hiện trong VPS. Khi không xuất hiện, giá trị của vps_independent_layer_flag[i] được suy ra bằng 1.

vps_direct_dependency_flag[i][j] có thể được gán bằng 0 để xác định rằng lớp có chỉ số j không phải là lớp tham chiếu trực tiếp cho lớp có chỉ số i. vps_direct_dependency_flag[i][j] có thể được gán bằng 1 để xác định rằng lớp

có chỉ số j là lớp tham chiếu trực tiếp cho lớp có chỉ số i . Khi $vps_direct_dependency_flag[i][j]$ không xuất hiện với i và j nằm trong khoảng từ 0 đến $vps_max_layers_minus1$, bao gồm hai giá trị đầu mút, $VPS_direct_dependency_flag[i][j]$ được suy ra bằng 0. Biến $DirectDependentLayerIdx[i][j]$, xác định lớp lệ thuộc trực tiếp thứ j của lớp thứ i , được dẫn xuất như sau:

```
for(i = 1; i < vps_max_layers_minus1; i++)
    if(!vps_independent_layer_flag[i])
        for(j = i, k = 0; j >= 0; j--)
            if(vps_direct_dependency_flag[i][j])
                DirectDependentLayerIdx[i][k++] = j
```

Biến $GeneralLayerIdx[i]$, xác định chỉ số lớp của lớp có nuh_layer_id bằng $vps_layer_id[i]$, được dẫn xuất như sau:

```
for(i = 0; i <= vps_max_layers_minus1; i++)
    GeneralLayerIdx[vps_layer_id[i]] = i
```

$each_layer_is_an_ols_flag$ có thể được gán bằng 1 để xác định rằng mỗi OLS chứa chỉ một lớp và chính mỗi lớp trong dòng bit là OLS có một lớp được bao gồm là chỉ một lớp đầu ra. $each_layer_is_an_ols_flag$ có thể được gán bằng 0 để xác định rằng OLS có thể chứa nhiều hơn một lớp. Nếu $vps_max_layers_minus1$ bằng 0, giá trị của $each_layer_is_an_ols_flag$ được suy ra bằng 1. Ngược lại, khi $vps_all_independent_layers_flag$ bằng 0, giá trị của $each_layer_is_an_ols_flag$ được suy ra bằng 0.

ols_mode_idc có thể được gán bằng 0 để xác định rằng tổng số OLS được xác định bởi VPS bằng $vps_max_layers_minus1 + 1$, OLS thứ i bao gồm các lớp có các chỉ số lớp từ 0 đến i , bao gồm hai giá trị đầu mút, và đối với mỗi OLS chỉ lớp cao nhất trong OLS được xuất ra. ols_mode_idc có thể được gán bằng 1 để xác định rằng tổng số OLS được xác định bởi VPS bằng $vps_max_layers_minus1 + 1$, OLS thứ i bao gồm các lớp có các chỉ số lớp từ 0 đến i , bao gồm hai giá trị đầu mút, và đối với mỗi OLS, tất cả các lớp trong OLS được xuất ra. ols_mode_idc có thể được gán bằng 2 để xác định rằng tổng số OLS được xác định bởi VPS được báo hiệu tường minh và đối với mỗi OLS, lớp

cao nhất và tập hợp lớp thấp hơn được báo hiệu tường minh trong OLS được xuất ra. Giá trị của `ols_mode_idc` sẽ nằm trong khoảng từ 0 đến 2, bao gồm hai giá trị đầu mút. Giá trị 3 của `ols_mode_idc` được dành riêng. Khi `vps_all_independent_layers_flag` bằng 1 và `each_layer_is_an_ols_flag` bằng 0, giá trị của `ols_mode_idc` được suy ra bằng 2.

`num_output_layer_sets_minus1` cộng 1 xác định tổng số OLS được xác định bởi VPS khi `ols_mode_idc` bằng 2. Biến `TotalNumOlss`, xác định tổng số OLSs được xác định bởi VPS, được dẫn xuất như sau:

```
if(vps_max_layers_minus1 == 0)
    TotalNumOlss = 1
else if(each_layer_is_an_ols_flag || ols_mode_idc == 0 || ols_mode_idc
== 1)
    TotalNumOlss = vps_max_layers_minus1 + 1
else if(ols_mode_idc == 2)
    TotalNumOlss = num_output_layer_sets_minus1 + 1
```

Lớp bao gồm `md_flag[i][j]` xác định liệu lớp thứ `j` (chẳng hạn, lớp có `nuh_layer_id` bằng `vps_layer_id[j]`) được bao gồm trong OLS thứ `i` khi `ols_mode_idc` bằng 2. `layer_included_flag[i][j]` có thể được gán bằng 1 để xác định rằng lớp thứ `j` được bao gồm trong OLS thứ `i`. `layer_included_flag[i][j]` có thể được gán bằng 0 để xác định rằng lớp thứ `j` không được bao gồm trong OLS thứ `i`.

Biến `NumLayersInOls[i]`, xác định số lượng lớp trong OLS thứ `i`, và biến `LayerIdInOls[i][j]`, xác định giá trị `nuh_layer_id` của lớp thứ `j` trong OLS thứ `i`, có thể được dẫn xuất như sau:

```
NumLayersInOls[0] = 1
LayerIdInOls[0][0] = vps_layer_id[0]
for(i = 1, i < TotalNumOlss; i++) {
    if(each_layer_is_an_ols_flag) {
        NumLayersInOls[i] = 1
        LayerIdInOls[i][0] = vps_layer_id[i]
    } else if(ols_mode_idc == 0 || ols_mode_idc == 1) {
```

```

    NumLayersInOls[i] = i + 1
    for(j = 0; j < NumLayersInOls[i]; j++)
        LayerIdInOls[i][j] = vps_layer_id[j]
    } else if(ols_mode_idc == 2) {
        for(k = 0, j = 0; k <= vps_max_layers_minus1; k++)
            if(layer_bao_gomd_flag[i][k])
                LayerIdInOls[i][j++] = vps_layer_id[k]
        NumLayersInOls[i] = j
    }
}

```

Biến `OlsLayerIdx[i][j]`, xác định chỉ số lớp OLS của lớp có `nuh_layer_id` bằng `LayerIdInOls[i][j]`, có thể được dẫn xuất như sau:

```

for(i = 0, i < TotalNumOlss; i++)
    for j = 0; j < NumLayersInOls[i]; j++)
        OlsLayerIdx[i][LayerIdInOls[i][j]] = j

```

Lớp thấp nhất trong mỗi OLS nên là lớp độc lập. Nói theo cách khác, đối với mỗi i trong khoảng từ 0 đến $TotalNumOlss - 1$, bao gồm hai giá trị đầu mút, giá trị của `vps_independent_layer_flag[GeneralLayerIdx[LayerIdInOls[i][0]]]` nên bằng 1. Mỗi lớp có thể được bao gồm trong ít nhất một OLS được xác định bởi VPS. Nói theo cách khác, đối với mỗi lớp có giá trị cụ thể của `nuh_layer_id` (chẳng hạn, `nuhLayerId` bằng một trong `vps_layer_id[k]` với k nằm trong khoảng từ 0 đến `vps_max_layers_minus1`, bao gồm hai giá trị đầu mút) nên có ít nhất một cặp giá trị i và j , trong đó i nằm trong khoảng từ 0 đến $TotalNumOlss - 1$, bao gồm hai giá trị đầu mút, và j nằm trong khoảng từ `NumLayersInOls[i] - 1`, bao gồm hai giá trị đầu mút, sao cho giá trị của `lớpIdInOls[i][j]` bằng `nuhLayerId`. Lớp trong OLS sẽ là lớp đầu ra của OLS hoặc lớp tham chiếu (trực tiếp hoặc gián tiếp) của lớp đầu ra của OLS.

`vps_output_layer_flag[i][j]` xác định liệu lớp thứ j trong OLS thứ i được xuất ra khi `ols_mode_idc` bằng 2. `VPS_output_layer_flag[i]` có thể được gán bằng 1 để xác định rằng lớp thứ j trong OLS thứ i được xuất ra. `VPS_output_layer_flag[i]` có thể được gán bằng 0 để xác định rằng lớp thứ j

trong OLS thứ i không được xuất ra. Khi `vps_all_independent_layers_flag` bằng 1 và `each_layer_is_an_ols_flag` bằng 0, giá trị của `vps_output_layer_flag[i]` có thể được suy ra bằng 1.

Biến `OutputLayerFlag[i][j]`, mà giá trị 1 của nó xác định rằng lớp thứ j trong OLS thứ i được xuất ra và giá trị 0 xác định rằng lớp thứ j trong OLS thứ i không được xuất ra, có thể được dẫn xuất như sau:

```
for(i = 0, i < TotalNumOlss; i++) {
    OutputLayerFlag[i][NumLayersInOls[i] - 1] = 1
    for(j = 0; j < NumLayersInOls[i] - 1; j++)
        if(ols_mode_idc[i] == 0)
            OutputLayerFlag[i][j] = 0
        else if(ols_mode_idc[i] == 1)
            OutputLayerFlag[i][j] = 1
        else if(ols_mode_idc[i] == 2)
            OutputLayerFlag[i][j] = vps_output_layer_flag[i][j]
}
```

OLS thứ 0 chứa chỉ lớp thấp nhất (chẳng hạn, lớp có `nuh_layer_id` bằng `vps_layer_id[0]`) và với OLS thứ 0, lớp được bao gồm duy nhất được xuất ra. `vps_constraint_info_present_flag` có thể được gán bằng 1 để xác định rằng cấu trúc cú pháp `general_constraint_info()` xuất hiện trong VPS. `VPS_constraint_info_present_flag` có thể được gán bằng 0 để xác định rằng cấu trúc cú pháp `general_constraint_info()` không xuất hiện trong VPS. `VPS_reserved_zero_7bits` cần bằng 0 trong các dòng bit tương hợp. Các giá trị khác cho `vps_reserved_zero_7bits` được dành riêng. Bộ giải mã nên bỏ qua giá trị của `vps_reserved_zero_7bits`.

`general_hrd_params_present_flag` có thể được gán bằng 1 để xác định rằng các phần tử cú pháp `num_units_in_tick` và `time_scale` và cấu trúc cú pháp `general_hrd_parameters()` hiện có trong cấu trúc cú pháp SPS RBSP. `general_hrd_params_present_flag` có thể được gán bằng 0 để xác định rằng các phần tử cú pháp `num_units_in_tick` và `time_scale` và cấu trúc cú pháp `general_hrd_parameters()` không xuất hiện trong cấu trúc cú pháp SPS RBSP.

`num_units_in_tick` là số lượng đơn vị thời gian của đồng hồ hoạt động ở tần số `time_scale` hertz (Hz) tương ứng với một số gia (được gọi là tích tắc đồng hồ) của bộ đếm tích tắc đồng hồ. `num_units_in_tick` nên lớn hơn 0. Tiếng tích tắc đồng hồ, theo các đơn vị giây, bằng thương số của `num_units_in_tick` chỉ cho `time_scale`. Chẳng hạn, khi tốc độ ảnh của tín hiệu video bằng 25Hz, `time_scale` có thể bằng 27000000 và `num_units_in_tick` có thể bằng 1080000, và kết quả là tích tắc đồng hồ có thể bằng 0,04 giây.

`time_scale` là số lượng đơn vị thời gian trôi qua trong một giây. Chẳng hạn, hệ thống tọa độ thời gian đo thời gian nhờ sử dụng đồng hồ 27megahertz (MHz) có `time_scale` 27000000. Giá trị của `time_scale` nên lớn hơn 0. `vps_extension_flag` có thể được gán bằng 0 để xác định rằng không phần tử cú pháp `vps_extension_data_flag` nào hiện có trong cấu trúc cú pháp VPS RBSP. `VPS_extension_flag` có thể được gán bằng 1 để xác định rằng có các phần tử cú pháp `vps_extension_data_flag` hiện có trong cấu trúc cú pháp VPS RBSP. `vps_extension_data_flag` có thể có giá trị bất kỳ. Sự có mặt và giá trị của `VPS_extension_data_flag` không ảnh hưởng đến sự tương hợp của bộ giải mã với các tiêu sử. Việc tương hợp của các bộ giải mã sẽ bỏ qua tất cả các phần tử cú pháp `vps_extension_data_flag`.

Fig.8 là sơ đồ của thiết bị tạo mã video 800 ví dụ. Thiết bị tạo mã video 800 thích hợp để thực hiện các ví dụ/phương án thực hiện được bộc lộ như được nêu ở đây. Thiết bị tạo mã video 800 bao gồm các cổng ra 820, các cổng vào 850, và/hoặc bộ thu phát (Tx/Rx) 810, bao gồm bộ truyền và/hoặc bộ nhận để truyền dữ liệu vào và/hoặc ra trên mạng. Thiết bị tạo mã video 800 cũng bao gồm bộ xử lý 830 bao gồm đơn vị logic và/hoặc khối xử lý trung tâm (central processing unit, CPU) để xử lý dữ liệu và bộ nhớ 832 để lưu trữ dữ liệu. Thiết bị tạo mã video 800 có thể cũng bao gồm linh kiện điện, linh kiện quang sang điện (optical-to-electrical, OE), linh kiện điện sang quang (electrical-to-optical, EO), và/hoặc các thành phần truyền thông không dây được ghép nối với các cổng vào 850 và/hoặc các cổng ra 820 để truyền thông dữ liệu qua điện, quang, hoặc mạng truyền thông không dây. Thiết bị tạo mã video 800 có thể cũng bao gồm các bộ phận nhập và/hoặc xuất (Input/Output, I/O) 860 để truyền thông dữ liệu đến và

từ người dùng. Các bộ phận I/O 860 có thể bao gồm các bộ phận xuất chẳng hạn bộ phận hiển thị để hiển thị dữ liệu video, loa để xuất dữ liệu audio, v.v.. Các bộ phận I/O 860 có thể cũng bao gồm các bộ phận nhập liệu, chẳng hạn bàn phím, chuột, bi xoay, v.v., và/hoặc các giao diện tương ứng để tương tác với các bộ phận xuất này.

Bộ xử lý 830 được thực hiện bằng phần cứng và phần mềm. Bộ xử lý 830 có thể được thực hiện dưới dạng một hoặc nhiều chip CPU, lõi (chẳng hạn, dưới dạng bộ xử lý nhiều lõi), mảng cổng dạng trường lập trình được (field-programmable gate array, FPGA), mạch tích hợp ứng dụng cụ thể (application specific integrated circuit, ASIC), và bộ xử lý tín hiệu số (digital signal processor, DSP). Bộ xử lý 830 truyền thông với các cổng ra 820, Tx/Rx 810, các cổng vào 850, và bộ nhớ 832. Bộ xử lý 830 bao gồm môđun tạo mã 814. Môđun tạo mã 814 thực hiện các phương án thực hiện được bộc lộ được mô tả ở đây, chẳng hạn các phương pháp 100, 900, và 1000, có thể sử dụng chuỗi video nhiều lớp 500, chuỗi video 600, và/hoặc dòng bit 700. Môđun tạo mã 814 có thể cũng thực hiện phương pháp/cơ chế khác bất kỳ được mô tả ở đây. Ngoài ra, môđun tạo mã 814 có thể thực hiện hệ thống codec 200, bộ mã hóa 300, và/hoặc bộ giải mã 400. Chẳng hạn, môđun tạo mã 814 có thể được sử dụng để tạo mã chuỗi video vào các lớp và/hoặc các OLS để hỗ trợ SNR và/hoặc tính khả mở không gian. Chẳng hạn, môđun tạo mã 814 có thể mã hóa và/hoặc giải mã phần tử cú pháp `ols_mode_idc` thành/từ VPS trong dòng bit. Phần tử cú pháp `ols_mode_idc` có thể chỉ báo rằng tổng số OLS trong chuỗi video bằng tổng số lớp được xác định trong VPS, rằng OLS thứ *i* bao gồm các lớp từ 0 đến *i*, bao gồm hai giá trị đầu mút, và đối với mỗi OLS chỉ lớp cao nhất được xuất ra. Do vậy, môđun tạo mã 814 có thể sử dụng phần tử cú pháp `ols_mode_idc` để chỉ báo/xác định rằng chỉ lớp cao nhất được nhận có thể được giải mã và được hiển thị và tất cả các lớp được nhận khác cần được giải mã để hỗ trợ giải mã lớp được nhận cao nhất. Thực tế, môđun tạo mã 814 khiến thiết bị tạo mã video 800 tạo chức năng bổ sung và/hoặc hiệu suất tạo mã khi tạo mã dữ liệu video. Thực tế, môđun tạo mã 814 cải thiện chức năng của thiết bị tạo mã video 800 cũng như giải quyết các vấn đề riêng cho kỹ thuật tạo mã video. Ngoài ra, môđun tạo mã 814 biến đổi thiết bị

tạo mã video 800 sang trạng thái khác. Theo cách khác, môđun tạo mã 814 có thể được thực hiện dưới dạng các lệnh được lưu trữ trong bộ nhớ 832 và được thực thi bằng bộ xử lý 830 (chẳng hạn, dưới dạng sản phẩm chương trình máy tính được lưu trữ trên phương tiện bất biến).

Bộ nhớ 832 bao gồm một hoặc nhiều loại bộ nhớ, chẳng hạn, đĩa, ổ băng, ổ trạng thái rắn, bộ nhớ chỉ đọc (read only memory, ROM), bộ nhớ truy nhập ngẫu nhiên (random access memory, RAM), bộ nhớ nhanh, bộ nhớ có nội dung đánh địa chỉ tam phân (ternary content-addressable memory, TCAM), RAM tĩnh (static random-access memory, SRAM), v.v.. Bộ nhớ 832 có thể được sử dụng làm bộ phận lưu trữ dữ liệu tràn, để lưu trữ chương trình khi các chương trình này được chọn để thực thi, và để lưu trữ các lệnh và dữ liệu được đọc khi thực thi chương trình.

Fig.9 là lưu đồ của phương pháp 900 mã hóa chuỗi video ví dụ với các OLS được tạo cấu hình cho tính khả mở, chẳng hạn chuỗi video nhiều lớp 500 và/hoặc chuỗi video 600 trong dòng bit 700. Phương pháp 900 có thể được sử dụng bằng bộ mã hóa, chẳng hạn hệ thống codec 200, bộ mã hóa 300, và/hoặc thiết bị tạo mã video 800 khi thực hiện phương pháp 100.

Phương pháp 900 có thể bắt đầu khi bộ mã hóa nhận chuỗi video và xác định để mã hóa chuỗi video đó dưới dạng chuỗi video khả mở trong tập hợp lớp và các OLS, chẳng hạn dựa trên đầu vào người dùng. Chuỗi video có thể được tạo mã để hỗ trợ tính khả mở SNR, tính khả mở không gian, tính khả mở theo các đặc tính khác được nêu ở đây, hoặc các tổ hợp khác của nó. Ở bước 901, bộ mã hóa có thể mã hóa dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã. Cụ thể là, bộ mã hóa có thể mã hóa lớp cơ sở có ID lớp thấp nhất và các lớp tăng cường khác nhau với việc tăng các ID lớp. Mỗi lớp tăng cường với ID lớp j có thể được tạo mã theo dự báo ngoài lớp dựa trên lớp cơ sở và bất kỳ các lớp tăng cường với ID lớp nhỏ hơn j . Các lớp cũng có thể được tổ chức thành các OLS với các ID OLS, có thể được ký hiệu bằng i để phân biệt với các ID lớp j . Chẳng hạn, có thể có một OLS trên lớp. Thực tế, OLS với ID OLS i có thể bao gồm lớp đầu ra với ID lớp j trong đó i bằng j . OLS với ID OLS i có thể cũng bao gồm tất cả các lớp có các ID lớp giữa 0 và $j-1$, bao gồm hai giá trị đầu mút. Lấy

làm ví dụ, OLS có ID OLS 5 có thể bao gồm các lớp từ 0 đến 5 trong đó lớp 5 là lớp đầu ra.

Ở bước 903, bộ mã hóa mã hóa VPS thành dòng bit. VPS bao gồm phần tử cú pháp `ols_mode_idc`. `ols_mode_idc` có thể được gán để xác định rằng tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS. Ngoài ra, `ols_mode_idc` có thể được gán để xác định rằng OLS thứ *i* bao gồm các lớp có các chỉ số lớp từ 0 đến *i* và/hoặc *j* (chẳng hạn, *i* bằng *j* trong trường hợp này), bao gồm hai giá trị đầu mút. `ols_mode_idc` có thể cũng được gán để xác định rằng đối với mỗi OLS chỉ lớp cao nhất trong mỗi OLS là lớp đầu ra. Chẳng hạn, `ols_mode_idc` có thể được gán bằng một trong vài chế độ. Chế độ nêu trên có thể được báo hiệu khi `ols_mode_idc` được gán bằng 0. Trong một số ví dụ, VPS có thể cũng bao gồm `vps_max_layers_minus1` xác định số lượng lớp được xác định bởi VPS. Đây cũng là số lượng lớp được phép lớn nhất trong mỗi CVS viện dẫn đến VPS. `ols_mode_idc` có thể tạo tham chiếu `VPS_max_layers_minus1`.

Lấy làm ví dụ, chuỗi video có thể được giải mã ở bộ giải mã và/hoặc ở HRD ở bộ mã hóa để kiểm chứng các chuẩn. Khi giải mã chuỗi video, tổng số biến OLS (`TotalNumOls`) cho chuỗi video có thể được gán bằng `vps_max_layers_minus1` cộng 1 khi `each_layer_is_an_ols_flag` trong VPS được gán bằng 0, khi `ols_mode_idc` được gán bằng 0, hoặc khi `ols_mode_idc` được gán bằng 1. Như là ví dụ cụ thể, số lượng lớp trong OLS thứ *i* (`NumLayersInOls[i]`) và ID lớp trong OLS (`LayerIdInOls[i][j]`) xác định giá trị `layer_id` của lớp thứ *j* trong OLS thứ *i* có thể được dẫn xuất như sau:

```
NumLayersInOls[0] = 1
LayerIdInOls[0][0] = vps_layer_id[0]
for(i = 1, i < TotalNumOls; i++) {
    if(each_layer_is_an_ols_flag) {
        NumLayersInOls[i] = 1
        LayerIdInOls[i][0] = vps_layer_id[i]
    } else if(ols_mode_idc == 0 || ols_mode_idc == 1) {
        NumLayersInOls[i] = i + 1
```

```
for(j = 0; j < NumLayersInOls[i]; j++)
```

```
    LayerIdInOls[i][j] = vps_layer_id[j]
```

trong đó `vps_layer_id[i]` là ID lớp VPS thứ *i*, `TotalNumOls` là tổng số OLS được xác định bởi VPS, và `each_layer_is_an_ols_flag` là cờ mỗi lớp là OLS xác định liệu ít nhất một OLS chứa nhiều hơn một lớp.

Ở bước 905, bộ mã hóa có thể lưu trữ dòng bit để truyền thông về phía bộ giải mã. Chẳng hạn, bộ giải mã có thể nhận thức về các OLS khả dụng (chẳng hạn, qua truyền thông và/hoặc các giao thức khác chẳng hạn giao thức phát trực tuyến thích ứng động trên truyền siêu văn bản (dynamic adaptive streaming over hypertext transfer protocol, DASH)). Bộ giải mã có thể chọn và yêu cầu OLS có ID cao nhất có thể được giải mã/hiển thị đúng bởi bộ giải mã. Chẳng hạn, trong trường hợp tính khả mở không gian, bộ giải mã có thể yêu cầu OLS có kích thước ảnh được liên kết với màn hình được kết nối với bộ giải mã. Trong trường hợp tính khả mở SNR, bộ giải mã có thể yêu cầu OLS có ID cao nhất có thể được giải mã theo các điều kiện mạng hiện tại (chẳng hạn, theo băng thông truyền thông khả dụng). Sau đó, bộ mã hóa và/hoặc cache hoặc máy chủ nội dung trung gian có thể truyền OLS và (các) lớp liên kết với bộ giải mã để giải mã. Thực tế, bộ mã hóa có thể tạo chuỗi video có thể mở rộng hoặc thu hẹp dựa trên các yêu cầu của bộ giải mã.

Fig.10 là lưu đồ của phương pháp 1000 giải mã chuỗi video lấy làm ví dụ bao gồm OLS được tạo cấu hình cho tính khả mở, chẳng hạn chuỗi video nhiều lớp 500 và/hoặc chuỗi video 600 trong dòng bit 700. Phương pháp 1000 có thể được sử dụng bởi bộ giải mã, chẳng hạn hệ thống codec 200, bộ giải mã 400, và/hoặc thiết bị tạo mã video 800 khi thực hiện phương pháp 100.

Phương pháp 1000 có thể bắt đầu khi bộ giải mã bắt đầu nhận dòng bit chứa OLS có tập hợp lớp của chuỗi video khả mở, chẳng hạn như là kết quả của phương pháp 900. Chuỗi video có thể được tạo mã để hỗ trợ tính khả mở SNR, tính khả mở không gian, tính khả mở theo các đặc tính khác được nêu ở đây, hoặc các kết hợp của nó. Ở bước 1001, bộ giải mã có thể nhận dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã và VPS. Các lớp có thể được bao gồm trong OLS. Chẳng hạn, các lớp có thể bao gồm lớp cơ sở với ID lớp thấp

nhất và các lớp tăng cường khác nhau có tăng các ID lớp. Mỗi lớp tăng cường có ID lớp j có thể được tạo mã theo dự báo ngoài lớp dựa trên lớp cơ sở và bất kỳ lớp tăng cường nào có ID lớp nhỏ hơn j . OLS có thể bao gồm ID OLS, có thể được ký hiệu bằng i để phân biệt với các ID lớp j . Chẳng hạn, có thể có một OLS trong dòng bit được mã hóa theo lớp. Thực tế, OLS có ID OLS i có thể bao gồm lớp đầu ra có ID lớp j trong đó i bằng j . OLS được nhận với ID OLS i có thể cũng bao gồm tất cả các lớp có các ID lớp giữa 0 và $j-1$, bao gồm hai giá trị đầu mút. Lấy làm ví dụ, OLS được nhận với ID OLS ID 5 có thể bao gồm các lớp 0 đến 5 trong đó lớp 5 là lớp đầu ra. Cấu hình của OLS và các lớp có thể được chỉ báo bởi VPS.

Chẳng hạn, VPS bao gồm phần tử cú pháp `ols_mode_idc`. `ols_mode_idc` có thể được gán để xác định rằng tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS. Ngoài ra, `ols_mode_idc` có thể được gán để xác định rằng OLS thứ i bao gồm các lớp có các chỉ số lớp từ 0 đến i và/hoặc j (chẳng hạn, i bằng j trong trường hợp này), bao gồm hai giá trị đầu mút. `ols_mode_idc` cũng có thể được gán để xác định rằng đối với mỗi OLS chỉ lớp cao nhất trong mỗi OLS là lớp đầu ra. Chẳng hạn, `ols_mode_idc` có thể được gán bằng một trong vài chế độ. Chế độ nêu trên có thể được báo hiệu khi `ols_mode_idc` được gán bằng 0. Trong một số ví dụ, VPS có thể cũng bao gồm `vps_max_layers_minus1` xác định số lượng lớp được xác định bởi VPS. Đây cũng là số lượng lớp lớn nhất được phép trong mỗi CVS viện dẫn đến VPS. `ols_mode_idc` có thể tạo tham chiếu `VPS_max_layers_minus1`.

Ở bước 1003, bộ giải mã có thể xác định lớp đầu ra dựa trên `ols_mode_idc` trong VPS. Như là ví dụ, khi giải mã chuỗi video, biến tổng số OLS (`TotalNumOlss`) cho chuỗi video có thể được gán bằng `vps_max_layers_minus1` cộng 1 khi `each_layer_is_an_ols_flag` trong VPS được gán bằng 0, khi `ols_mode_idc` được gán bằng 0, hoặc khi `ols_mode_idc` được gán bằng 1. Như là ví dụ cụ thể, số lượng lớp trong OLS thứ i (`NumLayersInOls[i]`) và ID lớp trong OLS (`LayerIdInOls[i][j]`) xác định giá trị `nuh_layer_id` của lớp thứ j trong OLS thứ i có thể được dẫn xuất như sau:

```

NumLayersInOls[0] = 1
LayerIdInOls[0][0] = vps_layer_id[0]
for(i = 1, i < TotalNumOls; i++) {
    if(each_layer_is_an_ols_flag) {
        NumLayersInOls[i] = 1
        LayerIdInOls[i][0] = vps_layer_id[i]
    } else if(ols_mode_idc == 0 || ols_mode_idc == 1) {
        NumLayersInOls[i] = i + 1
        for(j = 0; j < NumLayersInOls[i]; j++)
            LayerIdInOls[i][j] = vps_layer_id[j]
    }
}

```

trong đó `vps_layer_id[i]` là ID lớp VPS thứ `i`, `TotalNumOls` là tổng số OLS được xác định bởi VPS, và `each_layer_is_an_ols_flag` là cờ mỗi lớp là OLS xác định liệu ít nhất một OLS chứa nhiều hơn một lớp. Quá trình cho phép bộ giải mã để xác định cấu hình của OLS cũng như ID lớp đầu ra và các ID lớp hỗ trợ.

Khi ID lớp đầu ra và các ID của các lớp hỗ trợ được biết, bộ giải mã có thể bắt đầu giải mã các ảnh được tạo mã trong lớp đầu ra dựa trên các lớp hỗ trợ nhờ sử dụng dự báo ngoài lớp. Ở bước 1005, bộ giải mã có thể giải mã ảnh được tạo mã từ lớp đầu ra để tạo ảnh được giải mã. Ở bước 1007, bộ giải mã có thể chuyển tiếp ảnh được giải mã để hiển thị dưới dạng một phần của chuỗi video được giải mã.

Chẳng hạn, bộ giải mã có thể nhận thức về các OLS khả dụng (chẳng hạn, qua truyền thông và/hoặc các giao thức khác chẳng hạn DASH). Bộ giải mã có thể chọn và yêu cầu OLS có ID cao nhất có thể được giải mã/hiển thị đúng bởi bộ giải mã. Chẳng hạn, trong trường hợp tính khả mở không gian, bộ giải mã có thể yêu cầu OLS có kích thước ảnh được liên kết với màn hình được kết nối với bộ giải mã. Trong trường hợp tính khả dụng SNR, bộ giải mã có thể yêu cầu OLS có ID cao nhất có thể được giải mã theo các điều kiện mạng hiện tại (chẳng hạn, theo băng thông truyền thông khả dụng). Bộ mã hóa và/hoặc cache hoặc máy chủ nội dung trung gian sau đó có thể truyền OLS và (các) lớp liên kết đến bộ giải mã để giải mã. Thực tế, bộ mã hóa có thể tạo chuỗi video có thể tăng hoặc

giảm dựa trên các nhu cầu của bộ giải mã. Sau đó, bộ giải mã có thể giải mã chuỗi video được yêu cầu khi nhận nhờ sử dụng phương pháp 1000.

Fig.11 là sơ đồ của hệ thống 1100 ví dụ để tạo mã chuỗi video với các OLS được tạo cấu hình cho tính khả mở, chẳng hạn chuỗi video nhiều lớp 500 và/hoặc chuỗi video 600 trong dòng bit 700. Hệ thống 1100 có thể được thực hiện bằng bộ mã hóa và bộ giải mã, chẳng hạn, hệ thống codec 200, bộ mã hóa 300, bộ giải mã 400, và/hoặc thiết bị tạo mã video 800. Ngoài ra, hệ thống 1100 có thể được sử dụng khi thực hiện phương pháp 100, 900, và/hoặc 1000.

Hệ thống 1100 bao gồm bộ mã hóa video 1102. Bộ mã hóa video 1102 bao gồm môđun mã hóa 1105 để mã hóa dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã. Môđun mã hóa 1105 còn để mã hóa VPS thành dòng bit, trong đó VPS bao gồm `ols_mode_idc` xác định rằng tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS. Bộ mã hóa video 1102 còn bao gồm môđun lưu trữ 1106 để lưu trữ dòng bit để truyền thông về phía bộ giải mã. Bộ mã hóa video 1102 còn bao gồm môđun truyền 1107 để truyền dòng bit về phía bộ giải mã video 1110. Bộ mã hóa video 1102 có thể còn được tạo cấu hình để thực hiện bước bất kỳ trong các bước của phương pháp 900.

Hệ thống 1100 cũng bao gồm bộ giải mã video 1110. Bộ giải mã video 1110 bao gồm môđun nhận 1111 để nhận dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã và VPS, trong đó VPS bao gồm `ols_mode_idc` xác định rằng tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS. Bộ giải mã video 1110 còn bao gồm môđun xác định 1113 để xác định lớp đầu ra dựa trên `ols_mode_idc` trong VPS. Bộ giải mã video 1110 còn bao gồm môđun giải mã 1115 để giải mã ảnh được tạo mã từ lớp đầu ra để tạo ảnh được giải mã. Bộ giải mã video 1110 còn bao gồm môđun chuyển tiếp 1117 để chuyển tiếp ảnh được giải mã để hiển thị dưới dạng một phần của chuỗi video được giải mã. Bộ giải mã video 1110 có thể còn được tạo cấu hình để thực hiện bước bất kỳ trong các bước của phương pháp 1000.

Thành phần thứ nhất được ghép nối trực tiếp với thành phần thứ hai khi không có các thành phần xen kẽ, ngoại trừ đường thẳng, dấu vết, hoặc phương tiện khác giữa thành phần thứ nhất và thành phần thứ hai. Thành phần thứ nhất được ghép

nối gián tiếp với thành phần thứ hai khi có thành phần đan xen khác ngoài đường thẳng, dấu vết, hoặc phương tiện khác giữa thành phần thứ nhất và thành phần thứ hai. Cụm từ “được ghép nối” và các biến thể của nó bao gồm cả ghép nối trực tiếp lẫn ghép nối gián tiếp. Việc sử dụng cụm từ “khoảng” nghĩa là khoảng bao gồm $\pm 10\%$ của số tiếp theo trừ khi có thông báo ngược lại.

Cũng cần lưu ý rằng các bước của các phương pháp lấy làm ví dụ được nêu ở đây không cần được yêu cầu được thực hiện theo thứ tự được mô tả, và thứ tự của các bước của các phương pháp này cần được hiểu là chỉ lấy làm ví dụ. Tương tự, các bước bổ sung có thể được bao gồm trong các phương pháp này, và các bước cụ thể có thể được bỏ qua hoặc kết hợp, trong các phương pháp nhất quán với các phương án thực hiện khác nhau của sáng chế.

Trong khi vài phương án thực hiện đã được nêu theo sáng chế, có thể hiểu rằng các hệ thống và các phương pháp được bộc lộ có thể được thực hiện ở nhiều dạng cụ thể khác mà không xa rời tinh thần hoặc phạm vi của sáng chế. Các ví dụ này được xem là minh họa và không giới hạn, và mục đích sẽ không bị giới hạn ở các chi tiết được nêu ở đây. Chẳng hạn, các phần tử hoặc các thành phần khác nhau có thể được kết hợp hoặc được tích hợp trong hệ thống khác hoặc các dấu hiệu cụ thể có thể bị bỏ qua hoặc không được thực hiện.

Ngoài ra, kỹ thuật, hệ thống, hệ thống phụ, và các phương pháp được mô tả và minh họa theo các phương án thực hiện khác nhau dưới dạng rời rạc có thể được kết hợp hoặc được tích hợp với các hệ thống, các thành phần, các kỹ thuật, hoặc các phương pháp khác mà không xa rời phạm vi của sáng chế. Các ví dụ về các thay đổi, thay thế, và cải biến có thể được hiểu bởi người có hiểu biết trung bình trong lĩnh vực và có thể được thực hiện mà không xa rời tinh thần và phạm vi được bộc lộ ở đây.

YÊU CẦU BẢO HỘ

1. Phương pháp tạo mã được triển khai bởi bộ giải mã, trong đó phương pháp bao gồm các bước:

nhận, bởi bộ nhận của bộ giải mã, dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã và tập hợp tham số video (video parameter set, VPS), trong đó VPS bao gồm mã nhận diện chế độ tập hợp lớp đầu ra (output layer set, OLS) (ols_mode_idc), và ols_mode_idc được gán bằng 0 để xác định ít nhất một trong:

tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS,

OLS thứ i bao gồm các lớp có các chỉ số lớp từ 0 đến i, bao gồm cả 0 và i, và

đối với mỗi OLS chỉ lớp cao nhất trong mỗi OLS là lớp đầu ra;

xác định, bởi bộ xử lý của bộ giải mã, lớp đầu ra dựa trên ols_mode_idc trong VPS; và

giải mã, bởi bộ xử lý của bộ giải mã, ảnh được tạo mã từ lớp đầu ra để tạo ảnh được giải mã.

2. Phương pháp theo điểm 1, trong đó VPS bao gồm các lớp lớn nhất VPS trừ đi 1 (vps_max_layers_minus1), và VPS_max_layers_minus1 cộng 1 xác định số lượng lớp lớn nhất được phép trong mỗi CVS chỉ đến VPS .

3. Phương pháp theo điểm 1 hoặc 2, trong đó tổng số OLS (TotalNumOlss) bằng vps_max_layers_minus1 cộng 1 khi ols_mode_idc bằng 0 hoặc khi ols_mode_idc bằng 1.

4. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó số lượng lớp trong OLS thứ i (NumLayersInOls[i]) và định danh (identifier, ID) lớp trong OLS (LayerIdInOLS[i][j]) xác định giá trị định danh lớp tiêu đề đơn vị lớp trừu tượng mạng (network abstraction layer, NAL) (nuh_layer_id) của lớp thứ j trong OLS thứ i được suy ra như sau:

```
NumLayersInOls[0] = 1
LayerIdInOls[0][0] = vps_layer_id[0]
for(i = 1, i < TotalNumOlss; i++) {
```

```

if(each_layer_is_an_ols_flag) {
    NumLayersInOls[i] = 1
    LayerIdInOls[i][0] = vps_layer_id[i]
} else if(ols_mode_idc == 0 || ols_mode_idc == 1) {
    NumLayersInOls[i] = i + 1
    for(j = 0; j < NumLayersInOls[i]; j++)
        LayerIdInOls[i][j] = vps_layer_id[j]

```

trong đó `vps_layer_id[i]` là định danh lớp VPS thứ *i*, `TotalNumOlss` là tổng số OLS được xác định bởi VPS, và `each_layer_is_an_ols_flag` chính là mỗi lớp là cờ OLS mà xác định liệu ít nhất một OLS chứa nhiều hơn một lớp.

5. Phương pháp tạo mã được triển khai bởi bộ mã hóa, trong đó phương pháp bao gồm các bước:

mã hóa, bởi bộ xử lý của bộ mã hóa, dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã;

mã hóa thành dòng bit, bởi bộ xử lý, tập hợp tham số video (video parameter set, VPS), trong đó VPS bao gồm mã nhận diện chế độ tập hợp lớp đầu ra (OLS) (`ols_mode_idc`), và `ols_mode_idc` được gán bằng 0 để xác định ít nhất một trong:

tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS,

OLS thứ *i* bao gồm các lớp có các chỉ số lớp từ 0 đến *i*, inclusive, và

đối với mỗi OLS chỉ lớp cao nhất trong mỗi OLS là lớp đầu ra; và lưu trữ, bởi bộ nhớ được ghép nối với bộ xử lý, dòng bit để truyền thông về phía bộ giải mã.

6. Phương pháp theo điểm 5, trong đó VPS bao gồm các lớp lớn nhất VPS trừ 1 (`vps_max_layers_minus1`), và `vps_max_layers_minus1` cộng 1 xác định số lượng lớp lớn nhất được phép trong mỗi CVS chỉ đến VPS .

7. Phương pháp theo điểm 5 hoặc 6, trong đó tổng số các OLS (`TotalNumOlss`) bằng `vps_max_layers_minus1` cộng 1 khi `ols_mode_idc` bằng 0 hoặc khi `ols_mode_idc` bằng 1.

8. Phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 7, trong đó số lượng lớp trong OLS thứ i ($\text{NumLayersInOls}[i]$) và định danh lớp (ID) trong OLS ($\text{LayerIdInOls}[i][j]$) xác định giá trị định danh lớp tiêu đề khối lớp trừu tượng mạng (network abstraction layer, NAL) (nuh_layer_id) của lớp thứ j trong OLS thứ i được suy ra như sau:

```

NumLayersInOls[0] = 1
LayerIdInOls[0][0] = vps_layer_id[0]
for(i = 1, i < TotalNumOlss; i++) {
    if(each_layer_is_an_ols_flag) {
        NumLayersInOls[i] = 1
        LayerIdInOls[i][0] = vps_layer_id[i]
    } else if(ols_mode_idc == 0 || ols_mode_idc == 1) {
        NumLayersInOls[i] = i + 1
        for(j = 0; j < NumLayersInOls[i]; j++)
            LayerIdInOls[i][j] = vps_layer_id[j]
    }
}

```

trong đó $\text{vps_layer_id}[i]$ là định danh lớp VPS thứ i , TotalNumOlss là tổng số OLS được xác định bởi VPS, và $\text{each_layer_is_an_ols_flag}$ chính là mỗi lớp là cờ OLS mà xác định liệu ít nhất một OLS chứa nhiều hơn một lớp.

9. Thiết bị tạo mã video bao gồm:

bộ xử lý, bộ nhận được ghép nối với bộ xử lý, bộ nhớ được ghép nối với bộ xử lý, và bộ truyền được ghép nối với bộ xử lý, trong đó bộ xử lý, bộ nhận, bộ nhớ, và bộ truyền được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 4.

10. Thiết bị tạo mã video bao gồm:

bộ xử lý, bộ nhận được ghép nối với bộ xử lý, bộ nhớ được ghép nối với bộ xử lý, và bộ truyền được ghép nối với bộ xử lý, trong đó bộ xử lý, bộ nhận, bộ nhớ, và bộ truyền được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 8.

11. Vật ghi máy tính đọc được bất biến bao gồm chương trình máy tính để sử dụng bởi thiết bị tạo mã video, chương trình máy tính bao gồm các lệnh máy tính có thể thực thi được lưu trữ trên vật ghi máy tính đọc được bất biến such that khi

được thực thi bởi bộ xử lý khiến thiết bị tạo mã video thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 4.

12. Vật ghi máy tính đọc được bất biến bao gồm chương trình máy tính để sử dụng bởi thiết bị tạo mã video, chương trình máy tính bao gồm các lệnh máy tính có thể thực thi được lưu trữ trên vật ghi máy tính đọc được bất biến sao cho khi được thực thi bởi bộ xử lý khiến thiết bị tạo mã video thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 8.

13. Bộ giải mã bao gồm:

phương tiện nhận để nhận dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã và VPS, trong đó VPS bao gồm mã nhận diện chế độ OLS (`ols_mode_idc`), và `ols_mode_idc` được gán bằng 0 để xác định ít nhất một trong:

tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS,

OLS thứ i bao gồm các lớp có các chỉ số lớp từ 0 đến i , bao gồm cả 0 lần i , và

đối với mỗi OLS chỉ lớp cao nhất trong mỗi OLS là lớp đầu ra;

phương tiện xác định để xác định lớp đầu ra dựa trên `ols_mode_idc` trong VPS;

phương tiện giải mã để giải mã ảnh được tạo mã từ lớp đầu ra để tạo ảnh được giải mã; và

phương tiện chuyển tiếp để chuyển tiếp ảnh được giải mã để hiển thị dưới dạng một phần của chuỗi video được giải mã.

14. Bộ giải mã theo điểm 13, trong đó bộ giải mã còn được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 2 đến 4.

15. Bộ mã hóa bao gồm:

phương tiện mã hóa để:

mã hóa dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã;

và

mã hóa VPS thành dòng bit, trong đó VPS bao gồm mã nhận diện chế độ OLS (`ols_mode_idc`), và `ols_mode_idc` được gán bằng 0 để xác định ít nhất một trong:

tổng số OLS được xác định bởi VPS bằng số lượng lớp được xác định bởi VPS,

OLS thứ i bao gồm các lớp có các chỉ số lớp từ 0 đến i , bao gồm cả 0 và i , và

đối với mỗi OLS chỉ lớp cao nhất trong mỗi OLS là lớp đầu ra; và phương tiện lưu trữ để lưu trữ dòng bit để truyền thông về phía bộ giải mã.

16. Bộ mã hóa theo điểm 15, trong đó bộ mã hóa còn được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 6 đến 8.

1/11

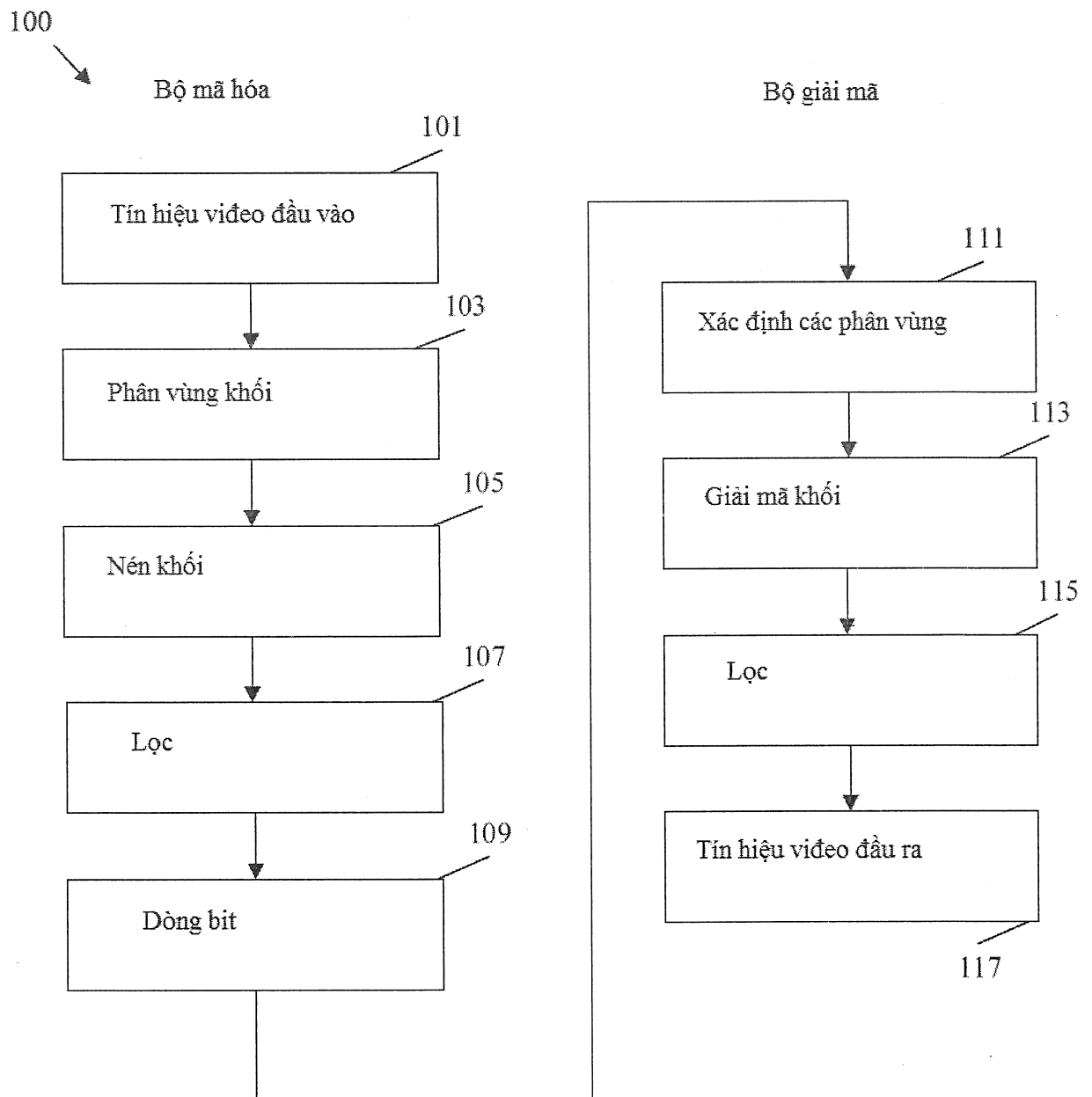


Fig.1

2/11

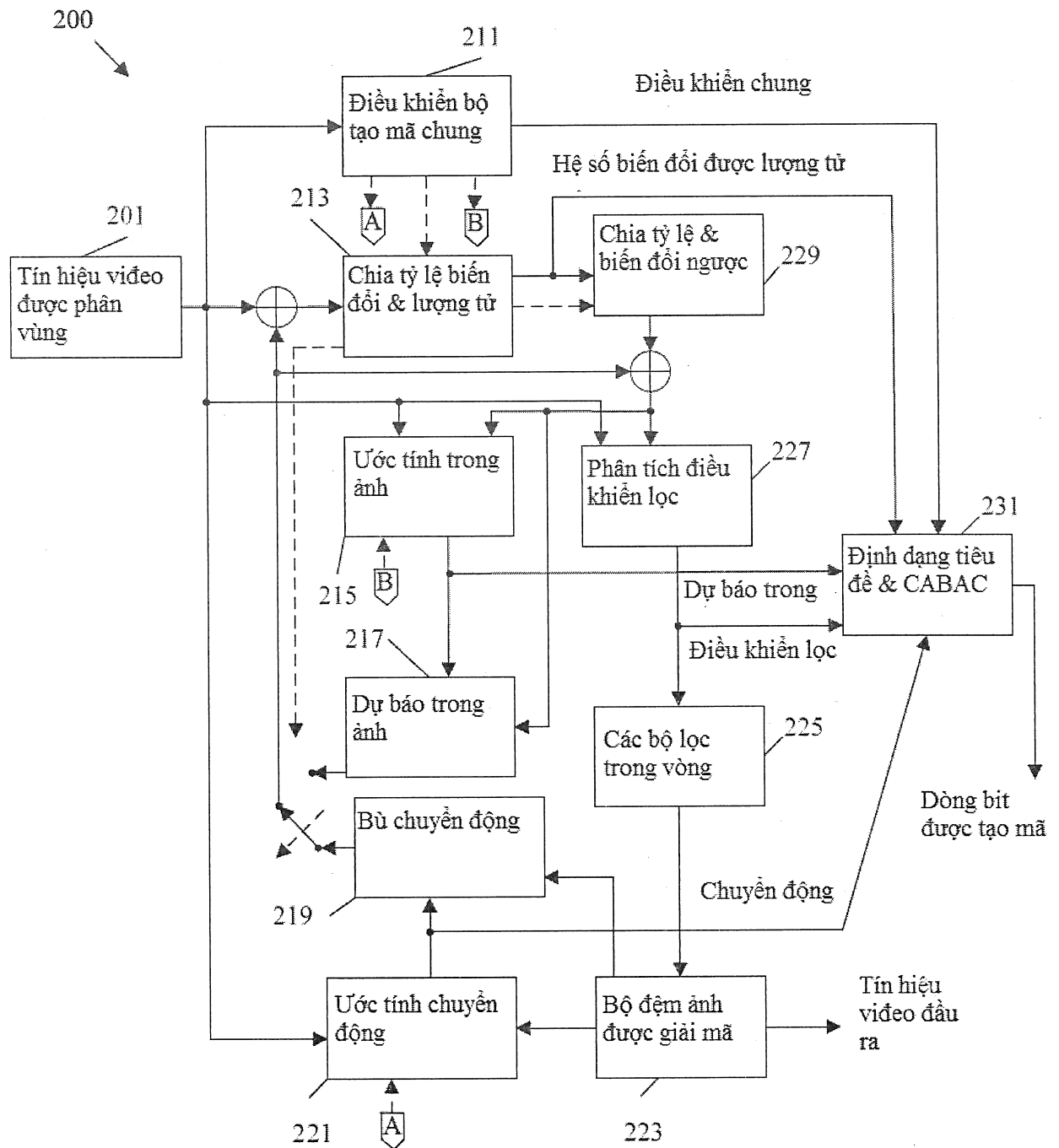


Fig.2

3/11

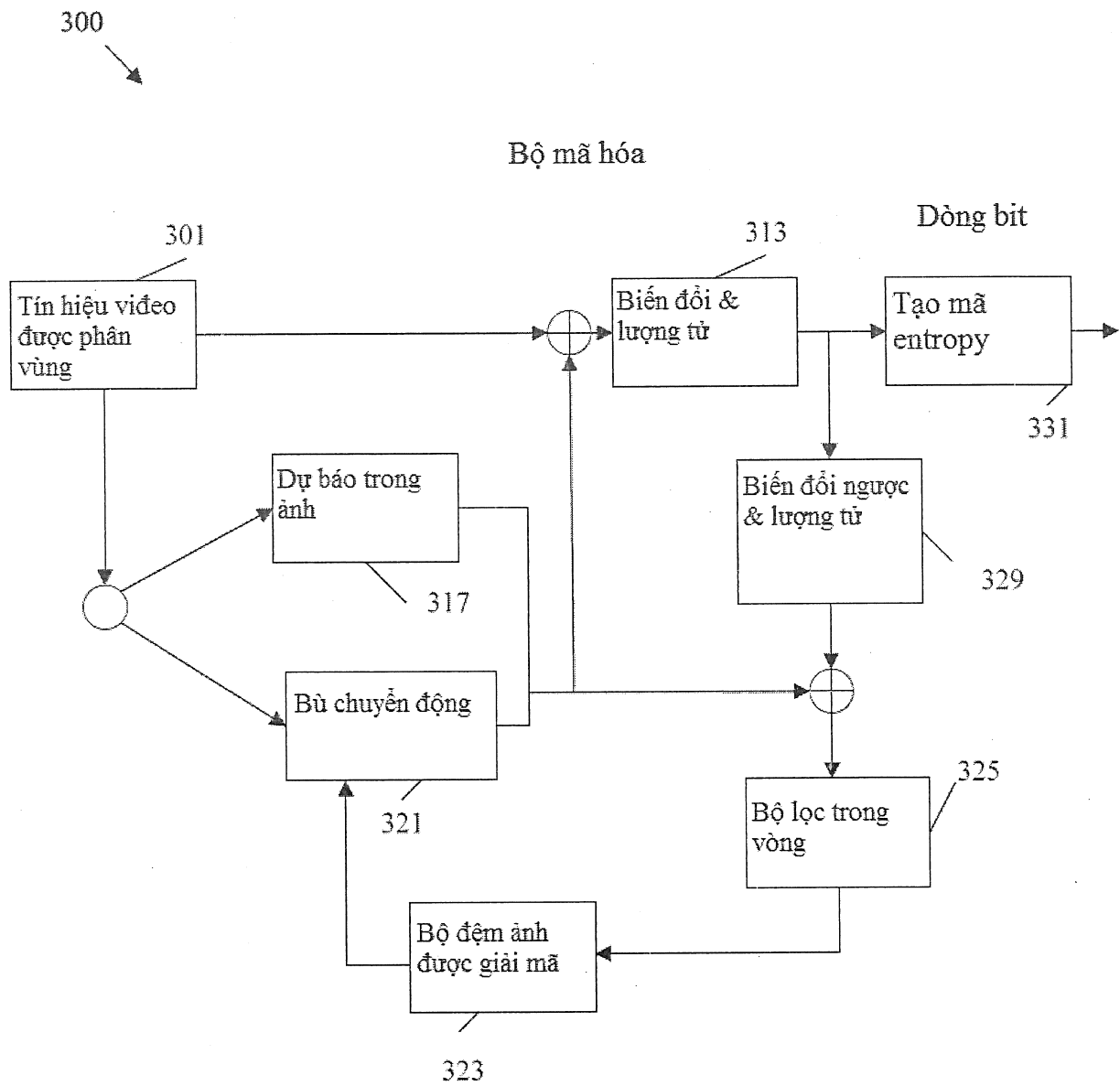


Fig.3

4/11

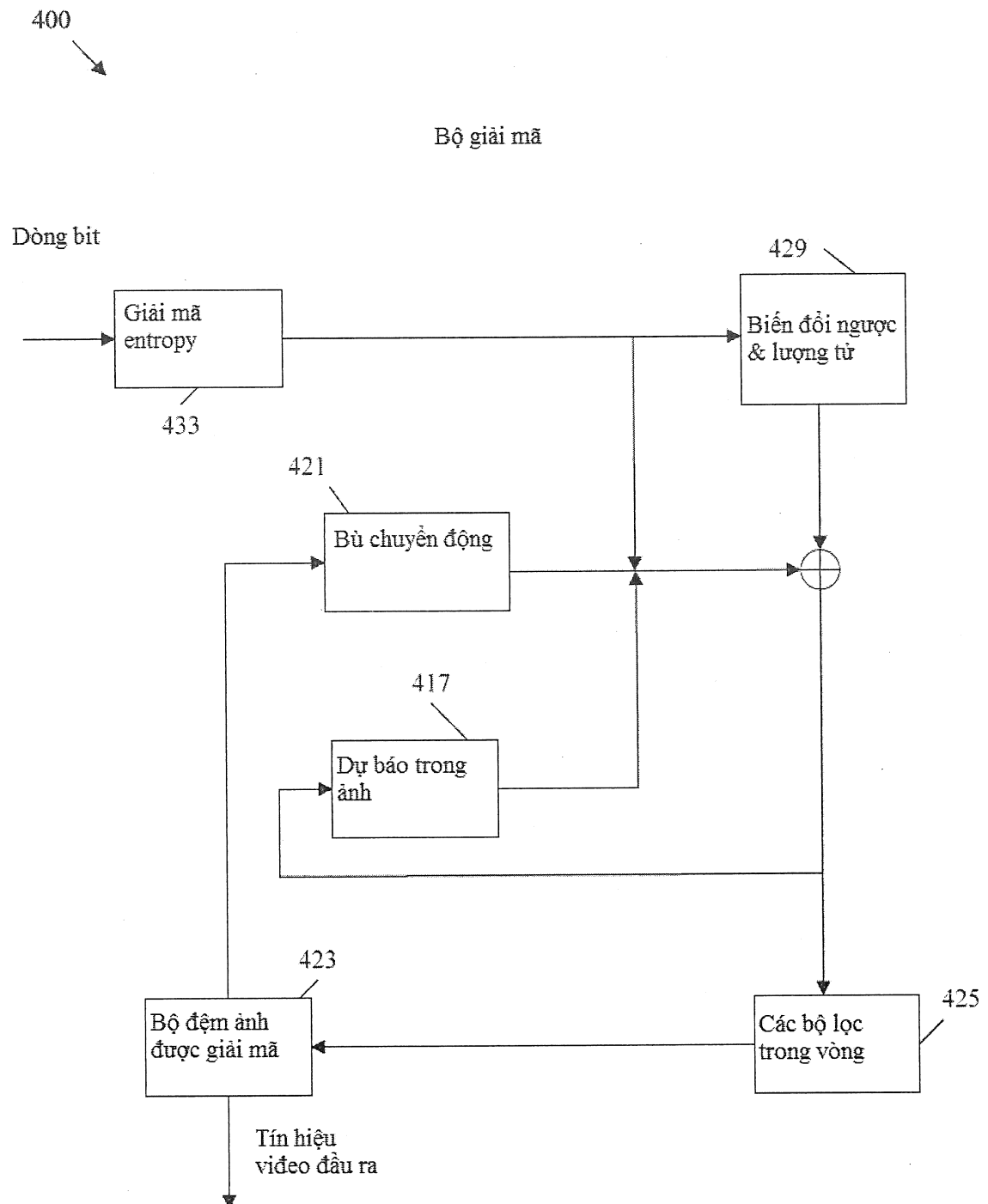


Fig.4

5/11

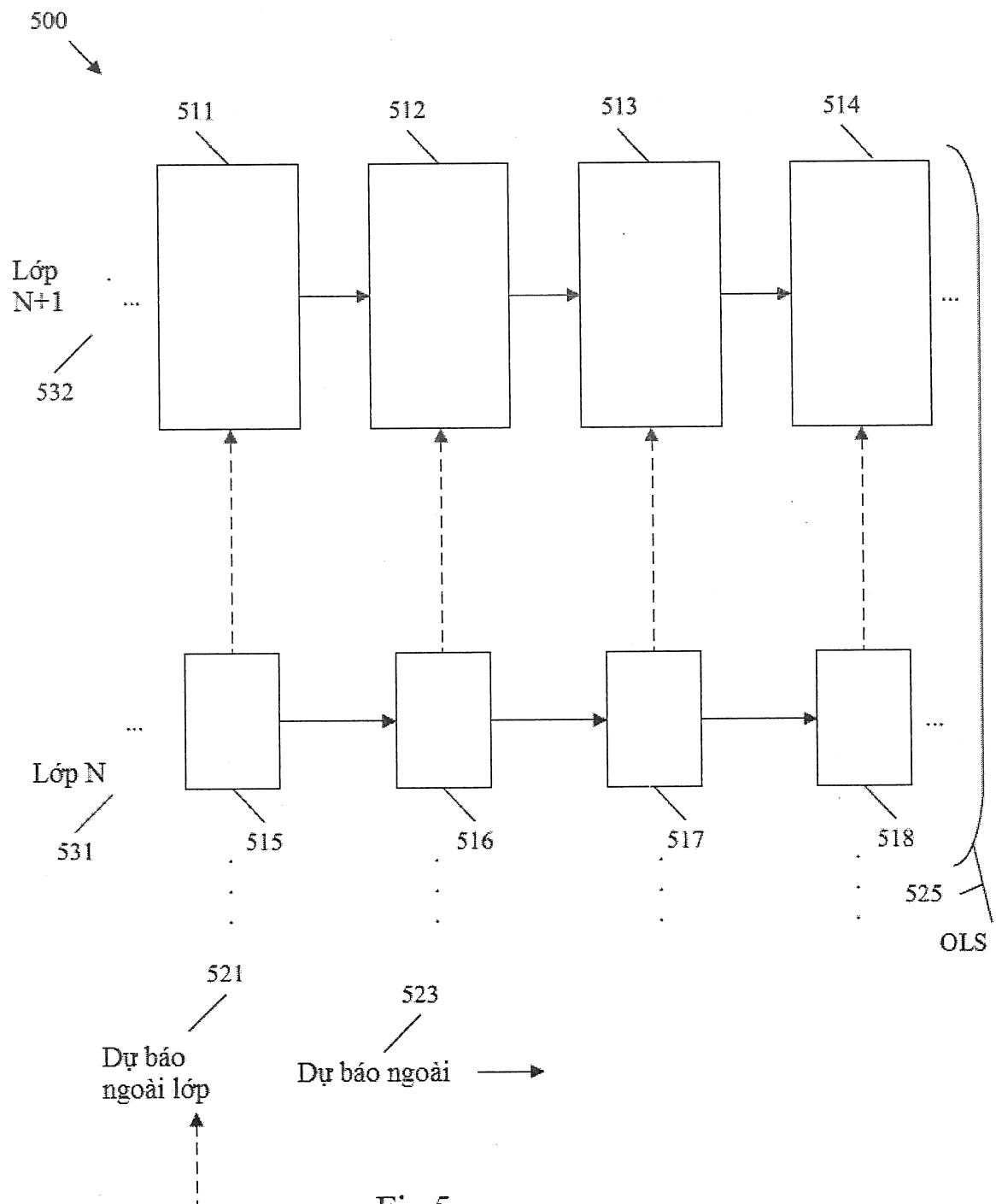


Fig.5

6/11

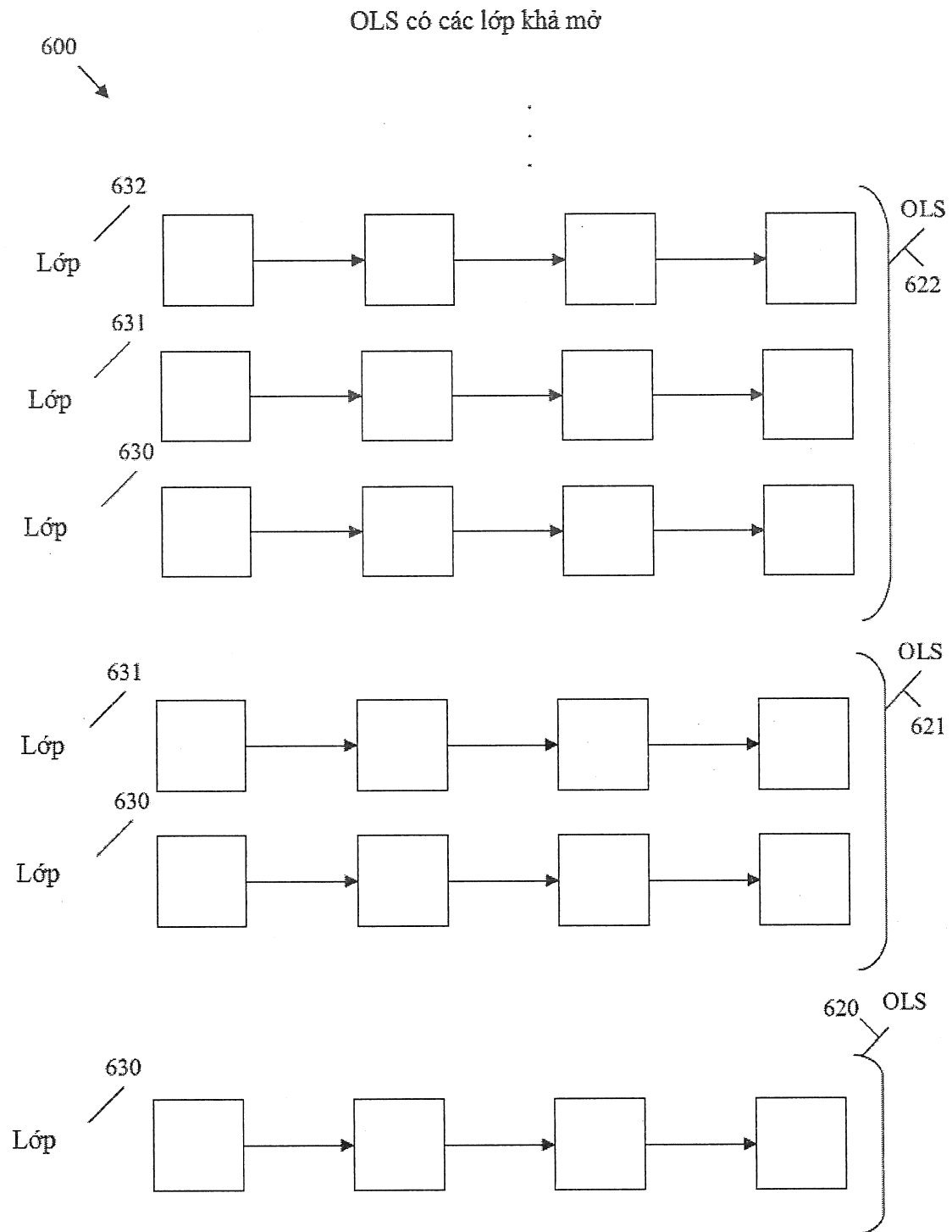


Fig.6

7/11

Dòng bit

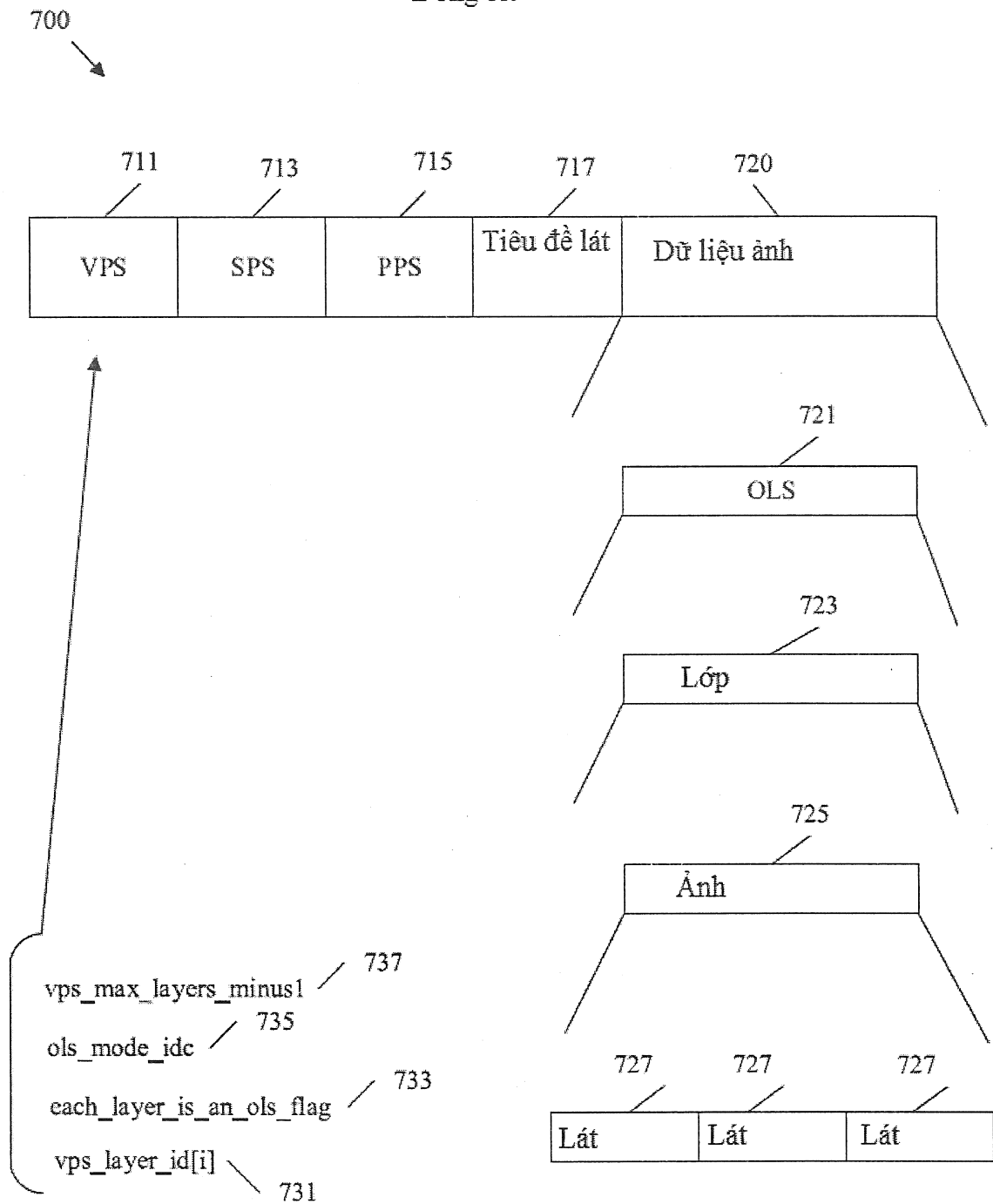


Fig.7

8/11

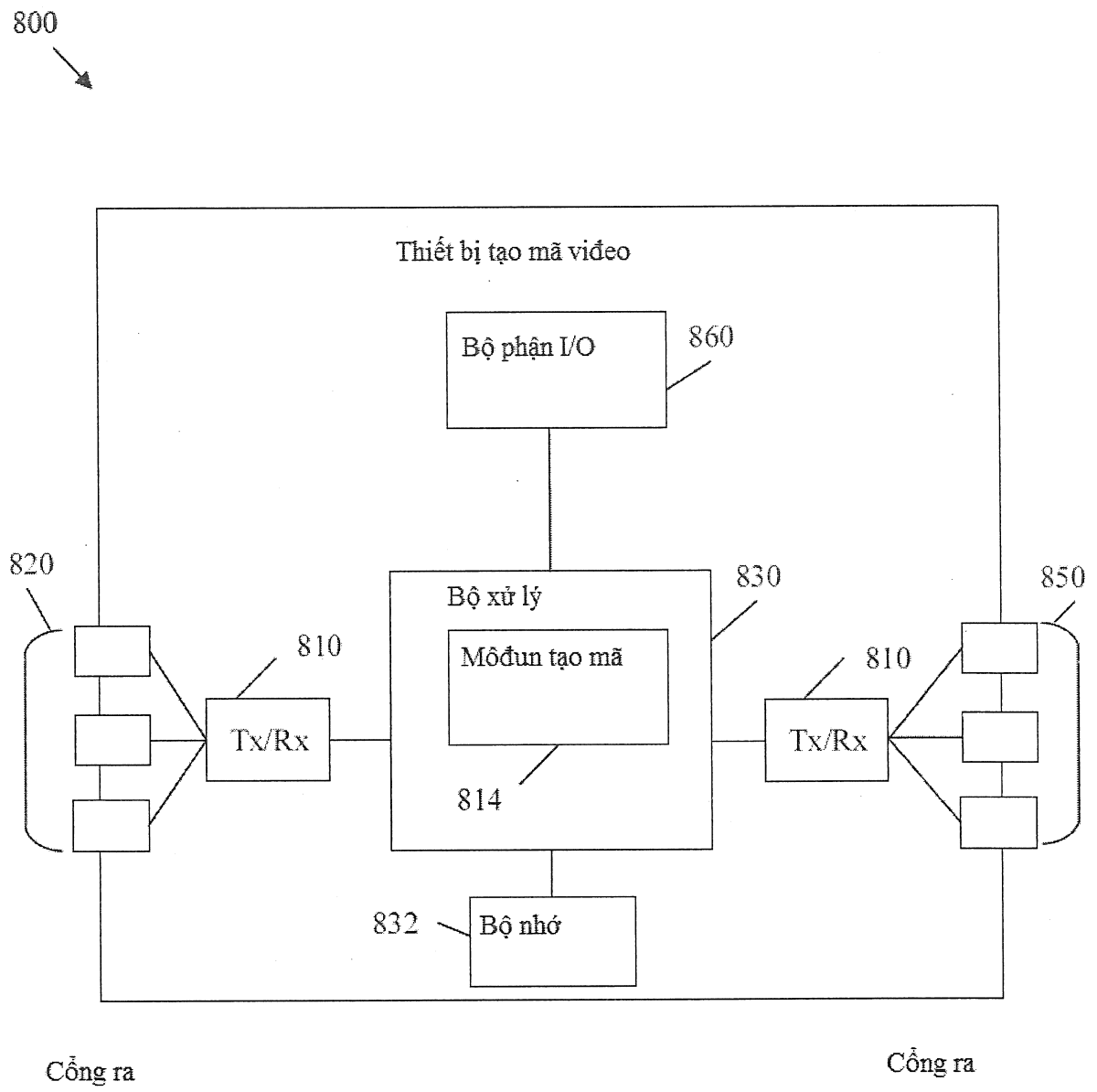


Fig.8

9/11

900

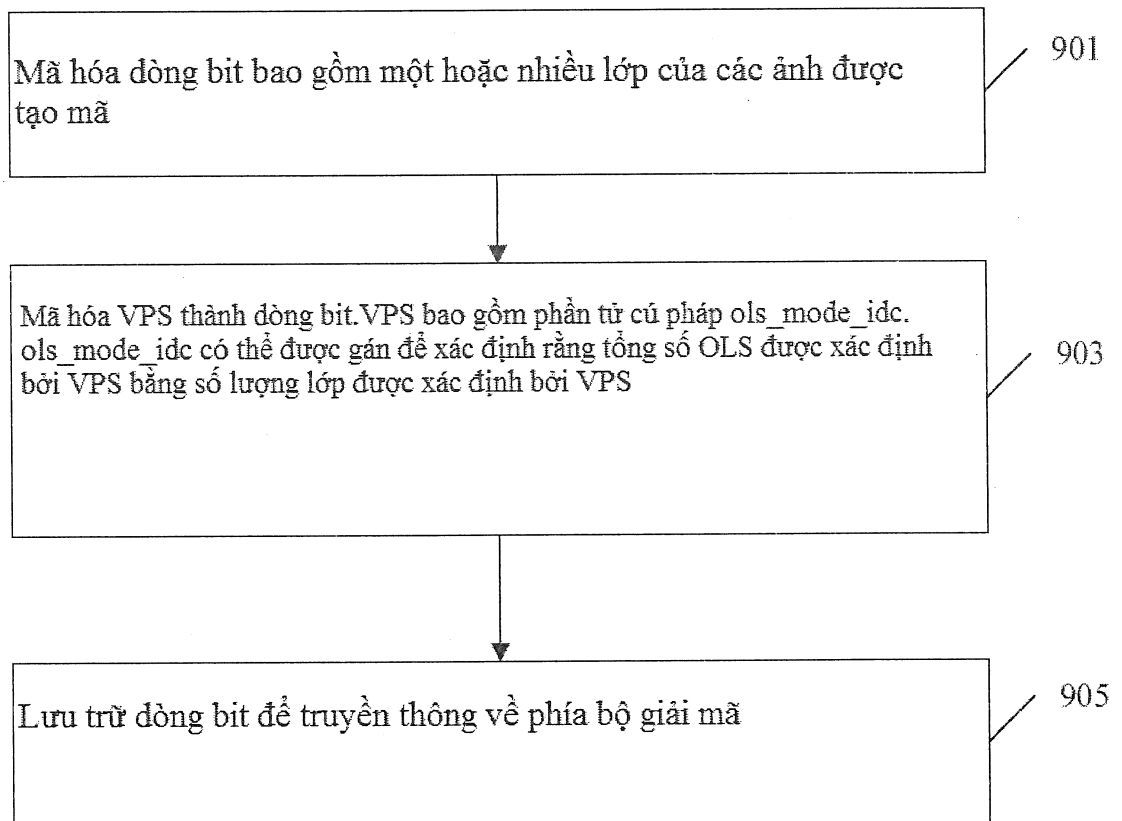


Fig.9

10/11

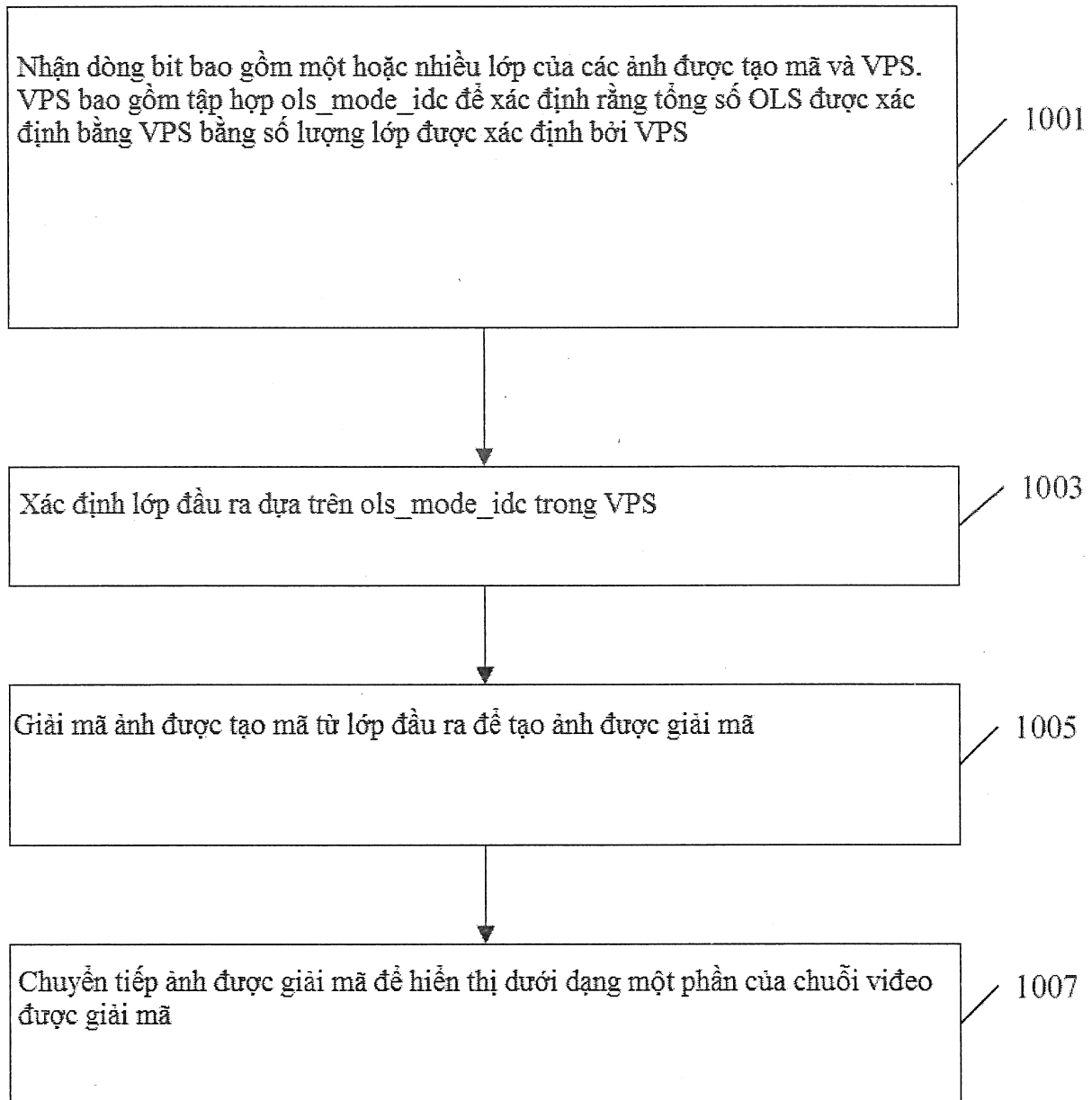


Fig.10

11/11

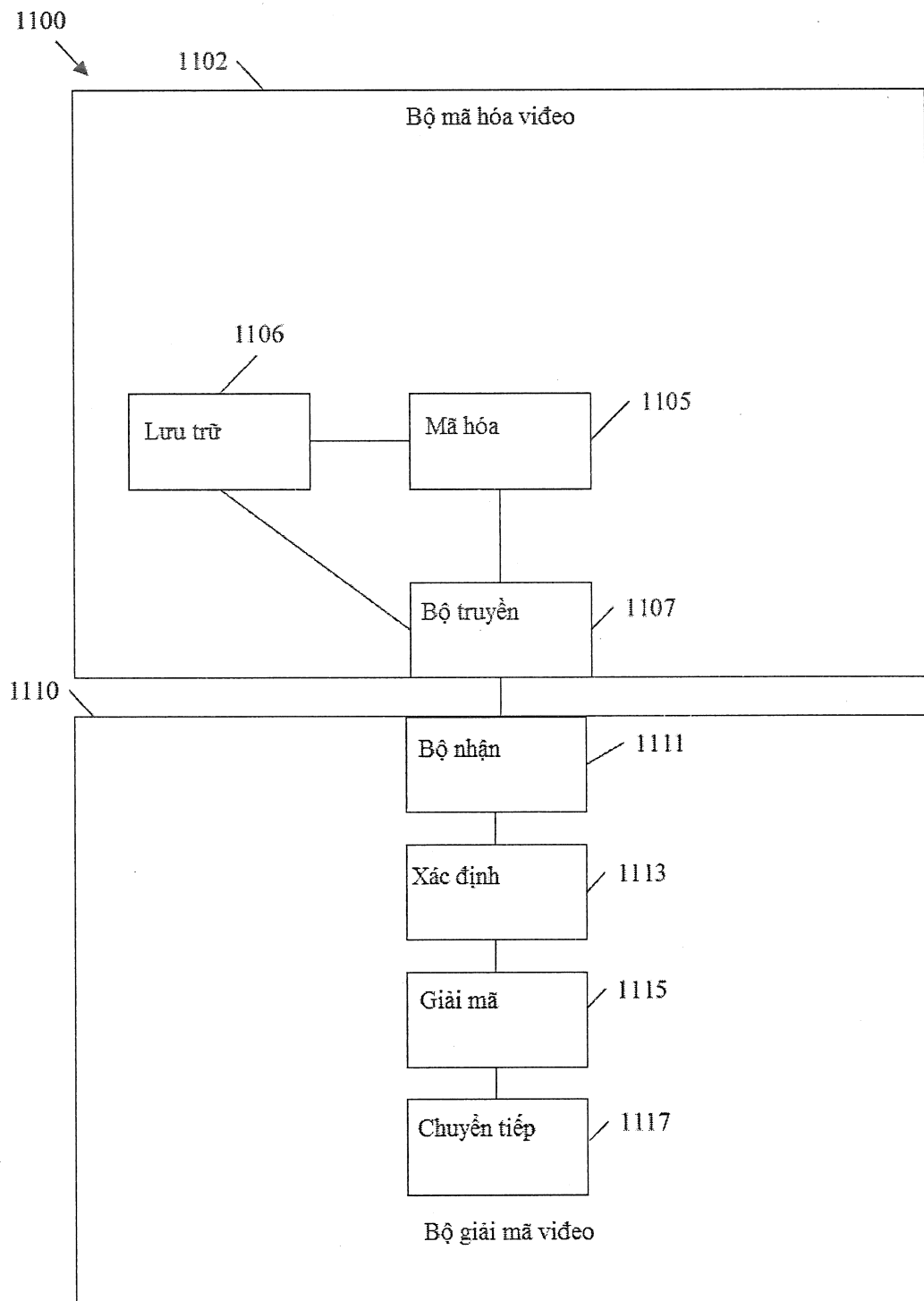


Fig.11