



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

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

(11)



1-0051124

(51)^{2006.01} H04N 7/26; H04N 7/50

(13) B

(21) 1-2024-05151

(22) 11/04/2008

(62) 1-2022-01281

(86) PCT/US2008/004747 11/04/2008

(87) WO2008/127676 23/10/2008

(30) 60/923,014 12/04/2007 US; 60/925,400 20/04/2007 US

(45) 25/09/2025 450

(43) 03/02/2025 442ASC

(73) Dolby International AB (NL)

Apollo Building, 3E, Herikerbergweg 1-35, 1101 CN Amsterdam Zuidoost, the Netherlands

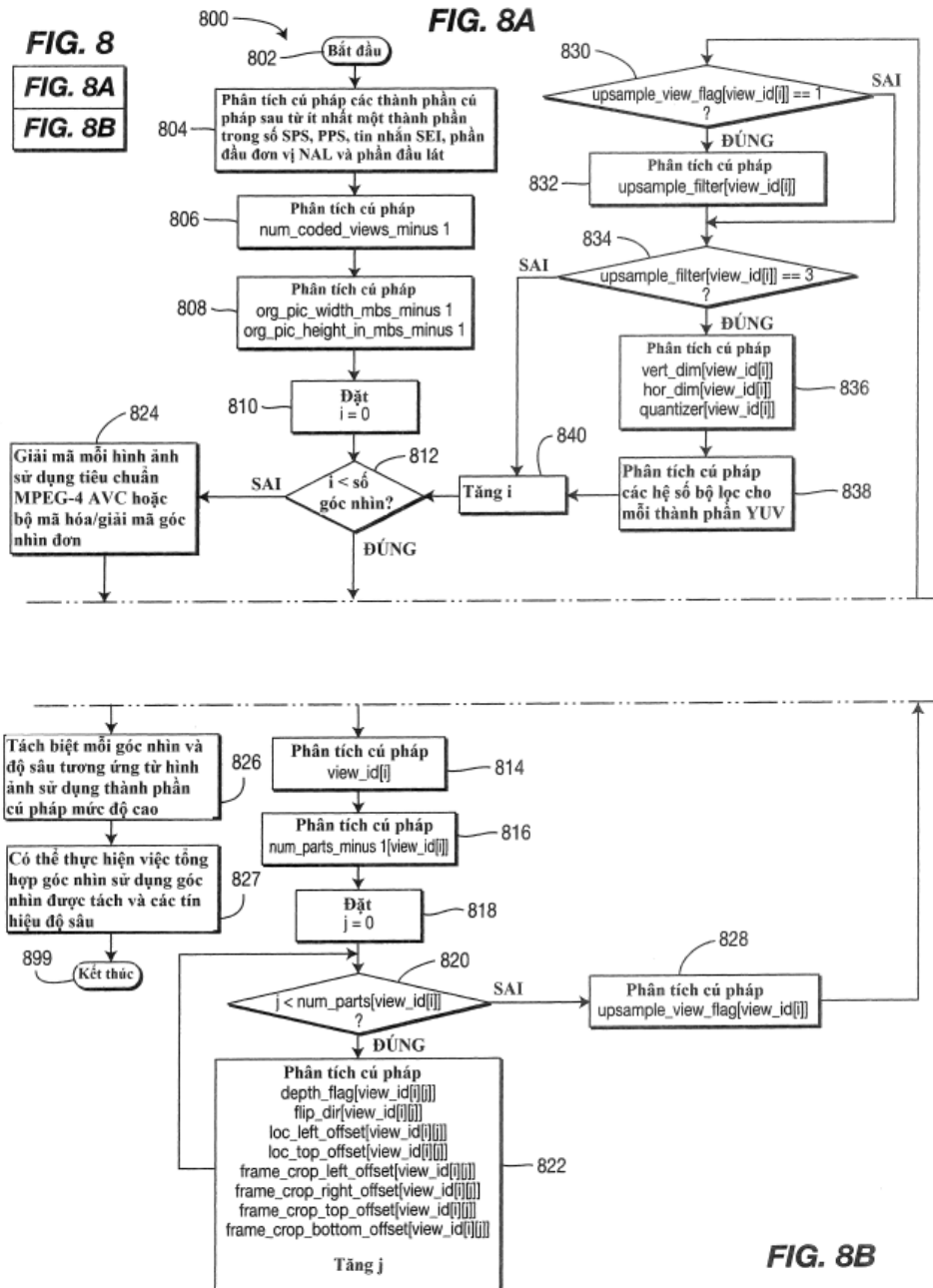
(72) PANDIT, Purvin Bibhas (IN); YIN, Peng (CN); TIAN, Dong (CN).

(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP ĐỂ GIẢI MÃ CÁC HÌNH VIDEO ĐƯỢC MÃ HÓA, VÀ
PHƯƠNG PHÁP ĐỂ GHI MÃ CÁC HÌNH VIDEO THÀNH LUỒNG BÍT ĐƯỢC
MÃ HÓA

(21) 1-2024-05151

(57) Sáng chế đề cập tới phương pháp để giải mã các hình video được mã hóa, và phương pháp để ghi mã các hình video thành luồng bit được mã hóa. Phương pháp áp dụng trong thiết bị này bao gồm các bước truy xuất hình video chứa các hình kết hợp thành một hình (826), truy xuất thông tin biểu thị cách thức các hình trong hình video truy xuất được kết hợp (806, 808, 822), giải mã hình video để tạo ra mẫu giải mã của ít nhất một trong số các hình (824, 826), và tạo ra thông tin truy xuất và hình video giải mã là đầu ra (824, 826). Trong một số quy trình khác, việc định dạng hoặc xử lý thông tin mà biểu thị cách thức các hình có trong hình video đơn được kết hợp thành một hình video, và định dạng và xử lý mẫu được mã hoá của nhiều hình kết hợp được. Sáng chế còn đề cập tới vật ghi phi chuyển tiếp đọc được bởi bộ xử lý.



Lĩnh vực kỹ thuật được đề cập

Nói chung, sáng chế đề cập đến kỹ thuật mã hoá và giải mã, và cụ thể hơn nữa là đến kỹ thuật mã hoá và/hoặc giải mã video.

Tình trạng kỹ thuật của sáng chế

Các hãng sản xuất màn hình video có thể sử dụng khung sắp xếp hoặc xếp lát các cảnh khác nhau trên một khung. Sau đó các cảnh này có thể được trích từ các vị trí tương ứng của chúng và được hiển thị.

Bản chất kỹ thuật của sáng chế

Theo một khía cạnh, sáng chế đề xuất phương pháp truy xuất hình video, hình video này bao gồm nhiều hình kết hợp thành một hình. Thông tin được truy xuất biểu thị cách thức nhiều hình trong hình video được truy xuất được kết hợp. Hình video được giải mã để tạo ra phần biểu diễn được giải mã của nhiều hình kết hợp. Thông tin truy xuất và hình video giải mã này được tạo ra là đầu ra.

Theo một khía cạnh khác, thông tin được tạo ra biểu thị cách thức nhiều hình có trong hình video được kết hợp thành một hình. Hình video được mã hoá để tạo ra phần biểu diễn được mã hóa của nhiều hình kết hợp. Thông tin tạo ra và hình video mã hóa này được cung cấp là đầu ra.

Theo một khía cạnh khác, sáng chế đề xuất tín hiệu hoặc cấu trúc tín hiệu bao gồm thông tin biểu thị cách thức nhiều hình có trong hình video được kết hợp thành một hình video. Tín hiệu hoặc cấu trúc tín hiệu này còn bao gồm biểu diễn mã hoá của nhiều hình video kết hợp.

Theo một khía cạnh khác, sáng chế đề xuất phương pháp truy xuất hình video, hình

video này bao gồm nhiều hình kết hợp thành một hình. Thông tin được truy xuất biểu thị cách thức nhiều hình trong hình video truy xuất được kết hợp. Hình video được giải mã để tạo ra phần biểu diễn được giải mã của ít nhất một trong số các hình. Thông tin truy xuất và hình video giải mã này được tạo ra như đầu ra.

Theo một khía cạnh khác, sáng chế đề xuất phương pháp truy xuất hình video, hình video này bao gồm nhiều hình kết hợp thành một hình. Thông tin được truy xuất biểu thị cách thức nhiều hình trong hình video truy xuất được kết hợp. Hình video được giải mã để tạo ra phần biểu diễn được giải mã của nhiều hình kết hợp. Đầu vào của người dùng được nhận mà lựa chọn ít nhất một trong số nhiều hình để hiển thị. Đầu ra giải mã của ít nhất một hình được chọn được tạo ra, đầu ra giải mã này được tạo ra dựa vào thông tin truy xuất, phần biểu diễn được giải mã, và đầu vào của người dùng.

Các khía cạnh, dấu hiệu và ưu điểm của sáng chế sẽ trở nên rõ ràng hơn trong phần mô tả chi tiết sau đây có dựa vào các phương án làm ví dụ, các phương án này có các hình vẽ kèm theo.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ thể hiện bốn cảnh được xếp lát trên một khung;

Fig.2 là sơ đồ thể hiện bốn cảnh được lật và được xếp lát trên một khung;

Fig.3 thể hiện sơ đồ khối của bộ mã hoá video mà sáng chế có thể được áp dụng, theo một phương án của sáng chế;

Fig.4 thể hiện sơ đồ khối của bộ giải mã video mà sáng chế có thể được áp dụng, theo một phương án của sáng chế;

Fig.5 là sơ đồ thể hiện phương pháp mã hoá các hình cho nhiều cảnh bằng cách sử dụng chuẩn MPEG-4 AVC, theo một phương án của sáng chế;

Fig.6 là sơ đồ thể hiện phương pháp giải mã các hình cho nhiều cảnh bằng cách sử dụng chuẩn MPEG-4 AVC, theo một phương án của sáng chế;

Fig.7 là sơ đồ thể hiện phương pháp mã hoá các hình và chiều sâu của hình cho

nhiều cảnh bằng cách sử dụng chuẩn MPEG-4 AVC, theo một phương án của sáng chế;

Fig.8 là sơ đồ thể hiện phương pháp giải mã các hình và chiều sâu của hình cho nhiều cảnh bằng cách sử dụng chuẩn MPEG-4 AVC, theo một phương án của sáng chế;

Fig.9 là sơ đồ thể hiện tín hiệu chiều sâu, theo một phương án của sáng chế;

Fig.10 là sơ đồ thể hiện tín hiệu chiều sâu được thêm vào như một lát, theo một phương án của sáng chế;

Fig.11 là sơ đồ thể hiện 5 cảnh được xếp lát trên một khung, theo một phương án của sáng chế;

Fig.12 là sơ đồ khối thể hiện bộ mã hoá mã hoá video đa cảnh (MVC - Multi-view Video Coding) làm ví dụ mà sáng chế có thể áp dụng, theo một phương án của sáng chế;

Fig.13 là sơ đồ khối thể hiện bộ giải mã MVC làm ví dụ mà sáng chế có thể áp dụng, theo một phương án của sáng chế;

Fig.14 là sơ đồ thể hiện phương pháp xử lý các hình cho nhiều cảnh để chuẩn bị mã hoá các hình bằng cách sử dụng phần mở rộng mã hoá video đa cảnh (MVC) của chuẩn MPEG-4 AVC, theo một phương án của sáng chế;

Fig.15 là sơ đồ thể hiện phương pháp mã hoá các hình cho nhiều cảnh bằng cách sử dụng phần mở rộng mã hoá video đa cảnh (MVC) của chuẩn MPEG-4 AVC, theo một phương án của sáng chế;

Fig.16 là sơ đồ thể hiện phương pháp xử lý các hình cho nhiều cảnh để chuẩn bị giải mã các hình bằng cách sử dụng phần mở rộng mã hoá video đa cảnh (MVC) của chuẩn MPEG-4 AVC, theo một phương án của sáng chế;

Fig.17 là sơ đồ thể hiện phương pháp giải mã các hình cho nhiều cảnh bằng cách sử dụng phần mở rộng mã hoá video đa cảnh (MVC) của chuẩn MPEG-4 AVC, theo một phương án của sáng chế;

Fig.18 là sơ đồ thể hiện phương pháp xử lý các hình và chiều sâu của hình cho nhiều cảnh để chuẩn bị mã hoá các hình này bằng cách sử dụng phần mở rộng mã hoá

video đa cảnh (MVC) của chuẩn MPEG-4 AVC, theo một phương án của sáng chế;

Fig.19 là sơ đồ thể hiện phương pháp mã hoá các hình và chiều sâu của hình cho nhiều cảnh bằng cách sử dụng phần mở rộng mã hoá video đa cảnh (MVC) của chuẩn MPEG-4 AVC, theo một phương án của sáng chế;

Fig.20 là sơ đồ thể hiện phương pháp xử lý các hình và chiều sâu của hình cho nhiều cảnh để chuẩn bị giải mã các hình bằng cách sử dụng phần mở rộng mã hoá video đa cảnh (MVC) của chuẩn MPEG-4 AVC, theo một phương án của sáng chế;

Fig.21 là sơ đồ thể hiện phương pháp giải mã các hình và chiều sâu của hình cho nhiều cảnh sâu bằng cách sử dụng phần mở rộng mã hoá video đa cảnh (MVC) của chuẩn MPEG-4 AVC, theo một phương án của sáng chế;

Fig.22 là sơ đồ thể hiện các ví dụ xếp lát tại mức điểm ảnh, theo một phương án của sáng chế; và

Fig.23 là sơ đồ khối thể hiện thiết bị xử lý video mà sáng chế có thể được áp dụng, theo một phương án của sáng chế.

Mô tả chi tiết sáng chế

Sáng chế đề cập đến phương pháp và thiết bị xếp lát cảnh bằng kỹ thuật mã hoá và giải mã video. Do đó cần phải hiểu rằng chuyên gia trong lĩnh vực có thể có khả năng sắp xếp, mặc dù không được mô tả hoặc thể hiện tường minh ở đây, bản thân các nguyên lý của sáng chế cũng được đưa vào tinh thần và phạm vi của sáng chế.

Tất cả các ví dụ và ngôn ngữ có điều kiện mô tả ở đây được dùng cho mục đích truyền đạt kinh nghiệm nhằm giúp người đọc hiểu rõ các nguyên lý và các khái niệm của sáng chế được đóng góp bởi tác giả (các tác giả) sáng chế trong lĩnh vực, và được hiểu là không bị hạn chế bởi các ví dụ và điều kiện mô tả riêng ở đây.

Hơn nữa, tất cả các lệnh ở đây mô tả các nguyên lý, khía cạnh, và các phương án của sáng chế, cũng như các ví dụ riêng của chúng, có ý định bao gồm cả cấu trúc lẫn chức năng tương đương của chúng. Hơn nữa, các phần tương đương này bao gồm cả các

phần tương đương hiện thời đã biết lẫn các phần tương đương triển khai trong tương lai, chẳng hạn, phần tử triển khai bất kỳ thực hiện cùng một chức năng, bất kể là cấu trúc.

Do đó, ví dụ, cần phải hiểu rằng các sơ đồ khối được thể hiện ở đây biểu diễn các hình vẽ theo khái niệm của sáng chế. Tương tự, cần phải hiểu rằng mọi lưu đồ, sơ đồ, sơ đồ chuyển tiếp trạng thái, mã giả và các quy trình biểu diễn tương tự có thể gần như đại diện cho vật ghi đọc được bằng máy tính và được thực thi bằng máy tính hoặc bộ xử lý cho dù máy tính hoặc bộ xử lý này có được thể hiện tường minh hay không.

Các chức năng của các phần tử thể hiện trong các hình vẽ có thể được tạo ra qua việc sử dụng phần cứng dành riêng cũng như phần cứng có khả năng thực thi phần mềm kết hợp với phần mềm thích hợp. Khi được thực thi bởi bộ xử lý, các chức năng này có thể được tạo ra bởi một bộ xử lý dành riêng, bởi một bộ xử lý dùng chung, hoặc bởi các bộ xử lý riêng lẻ, một trong số chúng có thể được dùng chung. Hơn nữa, việc dùng thuật ngữ “bộ xử lý” hoặc “bộ điều khiển” sẽ không chỉ nói đến khả năng thực thi phần phần mềm trong phần cứng, mà có thể bao gồm, nhưng không hạn chế ở, phần cứng bộ xử lý tín hiệu số (DSP-digital signal processor), bộ nhớ chỉ đọc (ROM-read-only memory) để lưu trữ phần mềm, bộ nhớ truy nhập ngẫu nhiên (RAM- random access memory), và phương tiện lưu trữ không khả biến.

Phần cứng khác, thông thường và/hoặc tùy ý, cũng có thể được bao gồm. Tương tự, mọi phần chuyển thể hiện trong các hình vẽ chỉ là khái niệm. Chức năng của chúng có thể được thực hiện qua hoạt động lô-gic chương trình, qua lô-gic dành riêng, qua tương tác điều khiển chương trình với lô-gic dành riêng, hoặc điều khiển bằng tay, một kỹ thuật cụ thể có khả năng được chọn bằng phương tiện để hiểu rõ hơn từ nội dung ngữ cảnh.

Trong phần yêu cầu bảo hộ, mọi phần tử được biểu diễn là phương tiện thực hiện chức năng xác định có ý định bao gồm mọi cách thực hiện chức năng đó bao gồm, ví dụ, a) kết hợp các phần tử mạch thực hiện chức năng đó hoặc b) phần mềm dạng bất kỳ, bao gồm, phần sụn, vi mã hoặc tương tự, kết hợp với mạch thích hợp thực thi phần mềm để thực hiện các chức năng này. Sáng chế được định nghĩa bằng các yêu cầu bảo hộ nằm

trong cơ sở lập luận mà các chức năng được cung cấp bởi các cách mô tả được kết hợp với nhau và cùng mang lại cách giải thích cho các yêu cầu bảo hộ. Do vậy, có thể thấy rằng phương tiện bất kỳ có thể cung cấp các chức năng của chúng mà tương đương với các chức năng thể hiện ở đây.

Các thuật ngữ “một phương án” (hoặc “một thiết lập”) hoặc “phương án” (hoặc “thiết lập”) của sáng chế có nghĩa là một dấu hiệu, cấu trúc, đặc điểm cụ thể, và vv... mô tả ở đây đề cập đến phương án có trong ít nhất một phương án của sáng chế. Do vậy, việc xuất hiện của thuật ngữ “theo một phương án” hoặc “trong một phương án” tại các vị trí trong toàn bộ bản mô tả không nhất thiết để chỉ cùng một phương án.

Cần phải hiểu rằng việc sử dụng các thuật ngữ “và/hoặc” và “ít nhất một”, ví dụ, trong các trường hợp “A và/hoặc B” và “ít nhất một trong số A và B”, có ý định bao gồm việc lựa chọn của duy nhất một tùy chọn liệt kê thứ nhất (A), hoặc lựa chọn của duy nhất một tùy chọn liệt kê thứ hai (B), hoặc lựa chọn của cả hai tùy chọn (A và B). Một ví dụ nữa, trong các trường hợp “A, B và/hoặc C” và “ít nhất một trong số A, B và C”, thuật ngữ này có ý định bao gồm việc lựa chọn của duy nhất một tùy chọn liệt kê thứ nhất (A), hoặc lựa chọn của duy nhất một tùy chọn liệt kê thứ hai (B), hoặc lựa chọn của duy nhất một tùy chọn liệt kê thứ ba (C), hoặc lựa chọn của duy nhất tùy chọn liệt kê thứ nhất và thứ hai (A và B), hoặc lựa chọn của duy nhất tùy chọn liệt kê thứ nhất và thứ ba (A và C), hoặc lựa chọn của duy nhất tùy chọn liệt kê thứ hai và thứ ba (B và C), hoặc lựa chọn của tất cả ba tùy chọn (A và B và C).

Hơn nữa, cần phải hiểu rằng trong khi một hoặc nhiều phương án của sáng chế mô tả ở đây đối với chuẩn MPEG-4 AVC, sáng chế không bị giới hạn bởi chuẩn này, do đó, có thể được sử dụng với các chuẩn, các quy trình, và các phần mở rộng khác của chúng, các chuẩn mã hoá video, các quy trình, và các phần mở rộng cụ thể của chúng bao gồm các phần mở rộng của chuẩn MPEG-4 AVC, trong khi vẫn duy trì tinh thần của sáng chế.

Hơn nữa, cần phải hiểu rằng trong khi một hoặc nhiều phương án khác của sáng chế được mô tả ở đây đối với phần mở rộng mã hoá video đa cảnh của chuẩn MPEG-4 AVC,

sáng chế không bị giới hạn bởi phần mở rộng và/hoặc chuẩn này và, do đó, có thể được sử dụng với các chuẩn mã hoá, các quy trình, và các phần mở rộng khác của chúng liên quan đến việc mã hoá video đa cảnh, trong khi vẫn duy trì tinh thần của sáng chế. Việc mã hoá video đa cảnh là khung nén cho việc mã hoá các chuỗi đa cảnh. Chuỗi MVC là tập hợp hai hoặc nhiều chuỗi video mà giữ cùng một cảnh từ một điểm cảnh khác.

Hơn nữa, cần phải hiểu rằng trong khi một hoặc nhiều phương án khác của sáng chế được mô tả ở đây sử dụng thông tin chiều sâu đối với nội dung video, sáng chế không bị giới hạn bởi các phương án này và, do đó, các phương án khác có thể được thực hiện mà không sử dụng thông tin chiều sâu, trong khi vẫn duy trì tinh thần của sáng chế.

Hơn nữa, như được dùng ở đây, “cú pháp mức cao” được dùng để chỉ cú pháp có trong luồng bit mà thường trú phân cấp trên lớp khối mac-ro. Ví dụ, cú pháp mức cao, như dùng ở đây, có thể đề cập đến, nhưng không bị hạn chế bởi, cú pháp tại mức phân đầu phiên, mức thông tin nâng cao phụ (SEI - Supplemental Enhancement Information), mức tập tham số hình (PPS - Picture Parameter Set), mức tập tham số chuỗi (SPS - Sequence Parameter Set), mức tập tham số cảnh (VPS - View Parameter Set), và mức phân đầu đơn vị lớp trừu tượng mạng (NAL - Network Abstraction Layer).

Trong quá trình thực hiện mã hoá video đa cảnh dựa vào Tổ chức tiêu chuẩn hoá quốc tế/Hội đồng kỹ thuật điện quốc tế (ISO/IEC - International Organization for Standardization/International Electrotechnical Commission) Phần 10 Hiệp hội chuyên gia điện ảnh (MPEG-4 -Moving Picture Experts Group-4) Chuẩn mã hoá video tiên tiến (AVC - Advanced Video Coding)/Hiệp hội truyền thông quốc tế, sec-tơ truyền thông (ITU-T - International Telecommunication Union, Telecommunication Sector) khuyến nghị H.264 (ở dưới đây là "chuẩn MPEG-4 AVC "), phần mềm liên quan thu được dự báo đa cảnh bằng cách mã hoá mỗi cảnh với một bộ mã hoá và tính đến các tham chiếu giữa các cảnh. Mỗi cảnh được mã hoá bằng bộ mã hoá là một luồng bit riêng theo độ phân giải ban đầu của nó và sau đó tất cả các luồng bit này được kết hợp để tạo thành một luồng bit mà sau này được giải mã. Mỗi cảnh tạo ra một đầu ra giải mã YUV riêng.

Theo một phương pháp khác, việc dự báo đa cảnh bao gồm việc nhóm tập các cảnh thành các cảnh giả. Theo một ví dụ của phương pháp này, chúng ta có thể xếp lát các hình từ mỗi N cảnh trong số toàn bộ M cảnh (ví dụ tại cùng một thời điểm) trên một khung lớn hơn hoặc một siêu khung có thể có hoạt động lấy mẫu xuống hoặc các hoạt động khác. Fig.1 là một ví dụ về bốn cảnh được xếp lát trên một khung được biểu thị chung bởi số chỉ dẫn 100. Toàn bộ bốn cảnh này nằm trong sự định hướng chuẩn của chúng.

Fig.2 là sơ đồ thể hiện bốn cảnh được lật và được xếp lát trên một khung được biểu thị chung bởi số chỉ dẫn 200. Cảnh trên bên trái nằm theo hướng chuẩn của nó. Cảnh trên bên phải được lật theo chiều ngang. Cảnh dưới bên trái được lật theo chiều dọc. Cảnh dưới bên phải được lật theo cả hai chiều dọc và ngang. Do đó, nếu có bốn cảnh, thì hình từ mỗi cảnh này được sắp xếp trong siêu khung giống như xếp lát. Việc này dẫn đến một chuỗi đầu vào không được mã hoá có độ phân giải lớn.

Cách khác, có thể lấy mẫu xuống ảnh để tạo ra độ phân giải nhỏ hơn. Do đó, có thể tạo ra nhiều chuỗi mỗi chuỗi này bao gồm các cảnh khác nhau mà được xếp lát cùng nhau. Mỗi chuỗi này sau đó tạo ra một cảnh giả, trong đó mỗi cảnh giả này bao gồm N cảnh được xếp lát khác nhau. Fig.1 thể hiện một cảnh giả, Fig.2 thể hiện một cảnh giả khác. Các cảnh giả này sau đó có thể được mã hoá bằng cách sử dụng các chuẩn mã hoá video hiện thời như chuẩn ISO/IEC MPEG-2 và chuẩn MPEG-4 AVC.

Theo một phương pháp khác nữa, việc dự báo đa cảnh thường bao gồm việc mã hoá độc lập các cảnh khác nhau bằng cách sử dụng chuẩn mới và, sau đó giải mã, xếp lát các cảnh theo yêu cầu của người chơi.

Hơn nữa, theo một phương pháp khác, các cảnh cũng có thể được xếp lát theo cách hiểu biết điểm ảnh. Ví dụ, trong một siêu cảnh mà bao gồm bốn cảnh, điểm ảnh (x, y) có thể từ cảnh 0, trong khi điểm ảnh $(x+1, y)$ có thể từ cảnh 1, điểm ảnh $(x, y+1)$ có thể từ cảnh 2, điểm ảnh $(x+1, y+1)$ có thể từ cảnh 3.

Rất nhiều hãng màn hình sử dụng khung này để sắp xếp hoặc xếp lát các cảnh khác

nhau trên một khung và sau đó trích các cảnh này từ các vị trí tương ứng của chúng và trình bày chúng. Trong những trường hợp này, không có một chuẩn nào để xác định xem luồng bit đã có đặc tính này hay chưa. Do đó, nếu hệ thống sử dụng phương pháp xếp lát các hình của các cảnh khác nhau trong một khung lớn, thì phương pháp trích các cảnh khác nhau thuộc quyền sở hữu riêng.

Tuy nhiên, không có một chuẩn nào để xác định xem luồng bit đã có đặc tính này hay chưa. Sáng chế đề xuất cú pháp mức cao để hỗ trợ người trình bày hoặc người chơi trích thông tin này để hỗ trợ việc hiển thị hoặc xử lý sau khác. Các hình con cũng có thể có độ phân giải khác nhau và một số hình con lấy mẫu lên có thể được cần đến để trình bày cảnh. Người dùng có thể muốn có phương pháp lấy mẫu lên được biểu thị trong cú pháp mức cao. Hơn nữa, các tham số để thay đổi tiêu điểm chiều sâu cũng có thể được truyền.

Theo một phương án, sáng chế đề xuất thông báo SEI mới để báo hiệu thông tin đa cảnh trong luồng bit tương thích chuẩn MPEG-4 AVC tại đó mỗi hình bao gồm các hình con mà thuộc về một cảnh khác. Phương án này được dành, ví dụ, cho việc hiển thị các luồng video đa cảnh trên các màn hình ba chiều (3D - three-dimensional) mà có thể sử dụng khung này dễ dàng và thuận tiện. Khái niệm có thể được mở rộng cho các chuẩn và các quy định mã hoá video khác báo hiệu thông tin này bằng cách sử dụng cú pháp mức cao.

Hơn nữa, theo một phương án, sáng chế đề xuất phương pháp báo hiệu cách thức sắp xếp các cảnh trước khi chúng được gửi cho bộ mã hoá và/hoặc bộ giải mã video đa cảnh. Có lợi là, phương án có thể làm cho việc cài đặt của tiến trình mã hoá video đa cảnh được đơn giản. Các cảnh có thể được đặt cùng nhau và tạo ra cảnh giả hoặc siêu cảnh và sau đó siêu cảnh được xếp lát này được xử lý như một cảnh thông thường bằng bộ mã hoá và/hoặc bộ giải mã video đa cảnh chung, ví dụ theo chuẩn MPEG-4 AVC hiện thời dựa vào việc thực hiện mã hoá video đa cảnh. Cờ mới được đưa ra trong phần mở rộng SPS của tiến trình mã hoá video đa cảnh để thông báo việc sử dụng kỹ thuật các cảnh giả. Phương án được dành để hiển thị các luồng video đa cảnh trên các màn hình

3D mà có thể sử dụng khung này dễ và thuận tiện.

Mã hoá/giải mã bằng cách sử dụng chuẩn/quy định mã hoá/giải mã video một cảnh

Trong quá trình thực hiện mã hoá video đa cảnh dựa vào Tổ chức tiêu chuẩn hoá quốc tế/Hội đồng kỹ thuật điện quốc tế (ISO/IEC - International Organization for Standardization/International Electrotechnical Commission) Phần 10 Hiệp hội chuyên gia điện ảnh (MPEG-4 -Moving Picture Experts Group-4) Chuẩn mã hoá video tiên tiến (AVC - Advanced Video Coding)/Hiệp hội truyền thông quốc tế, sec-tơ truyền thông (ITU-T - International Telecommunication Union, Telecommunication Sector) khuyến nghị H.264 (ở dưới đây là "chuẩn MPEG-4 AVC "), phần mềm liên quan đạt được dự báo đa cảnh bằng cách mã hoá mỗi cảnh với một bộ mã hoá và tính đến các tham chiếu giữa các cảnh. Mỗi cảnh được mã hoá bằng bộ mã hoá là một luồng bit riêng theo độ phân giải ban đầu của nó và sau đó tất cả các luồng bit này được kết hợp để tạo thành một luồng bit mà sau này được giải mã. Mỗi cảnh tạo ra một đầu ra giải mã YUV riêng.

Theo một phương pháp khác, việc dự báo đa cảnh bao gồm việc xếp lát các hình từ mỗi cảnh (ví dụ tại cùng một thời điểm) trên một khung lớn hơn hoặc một siêu khung có thể hoạt động lấy mẫu xuống. Fig.1 là một ví dụ thể hiện bốn cảnh được xếp lát trên một khung được biểu thị chung bởi số chỉ dẫn 100. Fig.2 là sơ đồ thể hiện bốn cảnh được lật và được xếp lát trên một khung được biểu thị chung bởi số chỉ dẫn 200. Do đó, nếu có bốn cảnh, thì hình từ mỗi cảnh này được sắp xếp trong siêu khung giống như xếp lát. Việc này dẫn đến một chuỗi đầu vào không được mã hoá có độ phân giải lớn. Tín hiệu này sau đó có thể được mã hoá bằng cách sử dụng các chuẩn mã hoá video hiện thời như chuẩn ISO/IEC MPEG-2 và chuẩn MPEG-4 AVC.

Theo một phương pháp khác nữa, việc dự báo đa cảnh thường bao gồm việc mã hoá độc lập các cảnh khác nhau bằng cách sử dụng chuẩn mới và, sau đó giải mã, xếp lát các cảnh theo yêu cầu của người chơi.

Nhiều hãng màn hình sử dụng khung này để sắp xếp hoặc xếp lát các cảnh khác nhau trên một khung và sau đó trích các cảnh này từ các vị trí tương ứng của chúng và

trình bày chúng. Trong những trường hợp này, không có một chuẩn nào để xác định xem luồng bit đã có đặc tính này hay chưa. Do đó, nếu hệ thống sử dụng phương pháp xếp lát các hình của các cảnh khác nhau trong khung lớn, thì phương pháp trích các cảnh khác nhau thuộc quyền sở hữu riêng.

Fig.3 là sơ đồ thể hiện bộ mã hoá video có khả năng thực hiện mã hoá video theo chuẩn MPEG-4 AVC được biểu thị chung bằng số chỉ dẫn 300.

Bộ mã hoá video 300 bao gồm bộ đệm sắp xếp khung 310 có đầu ra truyền thông tin hiệu với đầu vào không đảo của bộ kết hợp 385. Đầu ra của bộ kết hợp 385 này được nối truyền thông tin hiệu với đầu vào thứ nhất của bộ biến đổi và bộ lượng tử hoá 325. Đầu ra của bộ biến đổi và bộ lượng tử hoá 325 này được nối truyền thông tin hiệu với đầu vào thứ nhất của bộ mã hoá entropy 345 và đầu vào thứ nhất của bộ biến đổi ngược và bộ lượng tử hoá ngược 350. Đầu ra của bộ mã hoá entropy 345 này được nối truyền thông tin hiệu với đầu vào không đảo thứ nhất của bộ kết hợp 390. Đầu ra của bộ kết hợp 390 này được nối truyền thông tin hiệu với đầu vào thứ nhất của bộ đệm đầu ra 335.

Đầu ra thứ nhất của bộ điều khiển bộ mã hoá 305 được nối truyền thông tin hiệu với đầu vào thứ hai của bộ đệm sắp xếp khung 310, đầu vào thứ hai của bộ biến đổi ngược và bộ lượng tử hoá ngược 350, đầu vào của mô-đun phân xử loại hình 315, và đầu vào của mô-đun phân xử loại khối macro (MB-type) 320, đầu vào thứ hai của mô-đun dự báo bên trong 360, đầu vào thứ hai của bộ lọc tách khối 365, đầu vào thứ nhất của bộ bù chuyển động 370, đầu vào thứ nhất của bộ đánh giá chuyển động 375, và đầu vào thứ hai bộ đệm hình tham chiếu 380.

Đầu ra thứ hai của bộ điều khiển bộ mã hoá 305 được nối truyền thông tin hiệu với đầu vào thứ nhất của bộ chèn SEI 330, đầu vào thứ hai của bộ biến đổi và bộ lượng tử hoá 325, đầu vào thứ hai của bộ mã hoá entropy 345, đầu vào thứ hai của bộ đệm đầu ra 335, và đầu vào của bộ chèn SPS và PPS 340.

Đầu ra thứ nhất của mô-đun phân xử kiểu hình 315 được nối truyền thông tin hiệu với đầu vào thứ ba của bộ đệm sắp xếp khung 310. Đầu ra thứ hai của mô-đun phân xử

kiểu hình 315 được nối truyền thông tín hiệu với đầu vào thứ hai của mô-đun phân xử kiểu khối macro 320.

Đầu ra của bộ chèn SPS và PPS 340 được nối truyền thông tín hiệu với đầu vào không đảo thứ ba của bộ kết hợp 390. Đầu ra của bộ chèn SEI 330 được nối truyền thông tín hiệu với đầu vào không đảo thứ hai của bộ kết hợp 390.

Đầu ra của bộ lượng tử hoá ngược và bộ biến đổi ngược 350 được nối truyền thông tín hiệu với đầu vào không đảo thứ nhất của bộ kết hợp 319. Đầu ra của bộ kết hợp 319 này được nối truyền thông tín hiệu với đầu vào thứ nhất của mô-đun dự báo bên trong 360 và đầu vào thứ nhất của bộ lọc tách khối 365. Đầu ra của bộ lọc tách khối 365 được nối truyền thông tín hiệu với đầu vào thứ nhất của bộ đệm hình tham chiếu 380. Đầu ra của bộ đệm hình tham chiếu 380 này được nối truyền thông tín hiệu với đầu vào thứ hai của bộ đánh giá chuyển động 375 và với đầu vào thứ nhất của bộ bù chuyển động 370. Đầu vào thứ nhất của bộ bù chuyển động 370 này được nối truyền thông tín hiệu với đầu vào thứ hai của bộ bù chuyển động 370. Đầu ra thứ hai của bộ đánh giá chuyển động 375 được nối truyền thông tín hiệu với đầu vào thứ ba của bộ mã hoá entropy 345.

Đầu ra của bộ bù chuyển động 370 được nối truyền thông tín hiệu với đầu vào thứ nhất của bộ chuyển 397. Đầu ra của mô-đun dự báo bên trong 360 được nối truyền thông tín hiệu với đầu vào thứ hai của bộ chuyển 397. Đầu ra của mô-đun phân xử kiểu khối macro 320 được nối truyền thông tín hiệu với đầu vào thứ ba của bộ chuyển 397 để cung cấp đầu vào điều khiển cho bộ chuyển 397. Đầu vào thứ ba của bộ chuyển 397 này xác định xem đầu vào “dữ liệu” của bộ chuyển (so với đầu vào điều khiển, ví dụ, đầu vào thứ ba) có được cung cấp bởi bộ bù chuyển động 370 hoặc mô-đun dự báo bên trong 360 hay không. Đầu ra của bộ chuyển 397 được nối truyền thông tín hiệu với đầu vào không đảo thứ hai của bộ kết hợp 319 và với đầu vào đảo của bộ kết hợp 385.

Đầu vào của bộ đệm sắp xếp khung 310 và bộ điều khiển bộ mã hoá 105 khả dụng như đầu vào của bộ mã hóa 300, để nhận hình đầu vào 301. Hơn nữa, đầu vào của bộ

chèn SEI 330 khả dụng như đầu vào của bộ mã hóa 300, để nhận siêu dữ liệu. Đầu ra của bộ đệm đầu ra 335 khả dụng như đầu ra của bộ mã hóa 300, để đưa ra luồng bit.

Fig.4 là sơ đồ thể hiện bộ giải mã video có khả năng thực hiện giải mã video theo chuẩn MPEG-4 AVC được biểu thị chung bằng số chỉ dẫn 400.

Bộ giải mã video 400 bao gồm bộ đệm đầu vào 410 có đầu ra được nối truyền thông tín hiệu với đầu vào thứ nhất của bộ giải mã entropy 445. Đầu ra thứ nhất của bộ giải mã entropy 445 này được nối truyền thông tín hiệu với đầu vào thứ nhất của bộ biến đổi ngược và bộ lượng tử hoá ngược 450. Đầu ra của bộ biến đổi ngược và bộ lượng tử hoá ngược 450 này được nối truyền thông tín hiệu với đầu vào không đảo thứ hai của bộ kết hợp 425. Đầu ra của bộ kết hợp 425 này được nối truyền thông tín hiệu với đầu vào thứ hai của bộ lọc tách khối 465 và đầu vào thứ nhất của mô-đun dự báo bên trong 460. Đầu ra thứ hai của bộ lọc tách khối 465 được nối truyền thông tín hiệu với đầu vào thứ nhất của bộ đệm hình tham chiếu 480. Đầu ra của bộ đệm hình tham chiếu 480 được nối truyền thông tín hiệu với tín hiệu đầu vào thứ hai của bộ bù chuyển động 470.

Đầu ra thứ hai của bộ giải mã entropy 445 được nối truyền thông tín hiệu với đầu vào thứ ba của bộ bù chuyển động 470 và đầu vào thứ nhất của bộ lọc tách khối 465. Đầu ra thứ ba của bộ giải mã entropy 445 được nối truyền thông tín hiệu với đầu vào của bộ điều khiển bộ giải mã 405. Đầu ra thứ nhất của bộ điều khiển bộ giải mã 405 này được nối truyền thông tín hiệu với đầu vào thứ hai của bộ giải mã entropy 445. Đầu ra thứ hai của bộ điều khiển bộ giải mã 405 được nối truyền thông tín hiệu với đầu vào thứ hai của bộ biến đổi ngược và bộ lượng tử hoá ngược 450. Đầu ra thứ ba của bộ điều khiển bộ giải mã 405 được nối truyền thông tín hiệu với đầu vào thứ ba của bộ lọc tách khối 465. Đầu ra thứ tư của bộ điều khiển bộ giải mã 405 được nối truyền thông tín hiệu với đầu vào thứ hai của mô-đun dự báo bên trong 460, với đầu vào thứ nhất của bộ bù chuyển động 470, và với đầu vào thứ hai của bộ đệm hình tham chiếu 480.

Đầu ra của bộ bù chuyển động 470 được nối truyền thông tín hiệu với đầu vào thứ nhất của bộ chuyển 497. Đầu ra của mô-đun dự báo bên trong 460 được nối truyền thông

tín hiệu với đầu vào thứ hai của bộ chuyển 497. Đầu ra của bộ chuyển 497 này được nối truyền thông tín hiệu với đầu vào không đảo thứ nhất của bộ kết hợp 425.

Đầu vào của bộ đệm đầu vào 410 khả dụng như đầu vào của bộ giải mã 400, để nhận luồng bit đầu vào. Đầu ra thứ nhất của bộ lọc tách khối 465 khả dụng như đầu ra của bộ giải mã 400, để xuất ra hình đầu ra.

Fig.5 là sơ đồ thể hiện phương pháp mã hoá các hình cho nhiều cảnh bằng cách sử dụng chuẩn MPEG-4 AVC được biểu thị chung bằng số chỉ dẫn 500.

Phương pháp 500 bao gồm khối bắt đầu 502 chuyển quyền điều khiển cho khối chức năng 504. Khối chức năng 504 này sắp xếp mỗi cảnh tại một nấc thời gian cụ thể như là một hình con theo định dạng lát, và chuyển điều khiển cho khối chức năng 506. Khối chức năng 506 này thiết lập phần tử cú pháp `num_coded_views_minus1` và chuyển điều khiển cho khối chức năng 508. Khối chức năng 508 này thiết lập phần tử cú pháp `org_pic_width_in_mbs_minus1` và `org_pic_height_in_mbs_minus1`, và chuyển điều khiển cho khối chức năng 510. Khối chức năng 510 này đặt biến `i` bằng 0, và chuyển điều khiển cho khối phân xử 512. Khối phân xử 512 này xác định xem biến `i` có nhỏ hơn số lượng cảnh hay không. Nếu có, quyền điều khiển được chuyển cho khối chức năng 514. Ngược lại, điều khiển được chuyển cho khối chức năng 524.

Khối chức năng 514 thiết lập phần tử cú pháp `view_id[i]`, và chuyển điều khiển cho khối chức năng 516. Khối chức năng 516 này thiết lập phần tử cú pháp `num_parts[view_id[i]]`, và chuyển điều khiển cho khối chức năng 518. Khối chức năng 518 này đặt biến `j` bằng 0, và chuyển điều khiển cho khối phân xử 520. Khối phân xử 520 này xác định xem giá trị hiện thời của biến `j` có nhỏ hơn giá trị hiện thời của phần tử cú pháp `num_parts[view_id[i]]` hay không. Nếu có, điều khiển được chuyển cho khối chức năng 522. Nếu không, điều khiển được chuyển cho khối chức năng 528.

Khối chức năng 522 thiết lập các phần tử cú pháp sau đây, tăng biến `j`, và sau đó trả quyền điều khiển cho khối phân xử 520:

depth_flag[view_id[i]][j]; flip_dir[view_id[i]][j]; loc_left_offset[view_id[i]][j];
 loc_top_offset[view_id[i]][j]; frame_crop_left_offset[view_id[i]][j];
 frame_crop_right_offset [view_id [i]][j]; frame_crop_top_offset[view_id[i]][j]; và
 frame_crop_bottom_offset[view_id[i]][j].

Khối chức năng 528 thiết lập phần tử cú pháp upsample_view_flag[view_id [i]], và chuyển điều khiển cho khối phân xử 530. Khối phân xử 530 này xác định xem giá trị hiện thời của phần tử cú pháp upsample_view_flag[view_id[i]] có bằng 1 hay không. Nếu có, điều khiển được chuyển cho khối chức năng 532. Nếu không, điều khiển được chuyển cho khối phân xử 534.

Khối chức năng 532 thiết lập phần tử cú pháp upsample_filter[view_id [i]], và chuyển quyền điều khiển cho khối phân xử 534.

Khối phân xử 534 này xác định xem giá trị hiện thời của phần tử cú pháp upsample_filter[view_id[i]] có bằng 3 hay không. Nếu có, điều khiển được chuyển cho khối chức năng 536. Nếu không, điều khiển được chuyển cho khối chức năng 540.

Khối chức năng 536 thiết lập các phần tử cú pháp sau đây và chuyển điều khiển cho khối chức năng 538: vert_dim[view_id[i]]; hor_dim[view_id[i]]; và quantizer[view_id[i]].

Khối chức năng 538 này thiết lập các hệ số bộ lọc cho mỗi thành phần YUV, và chuyển điều khiển cho khối chức năng 540.

Khối chức năng 540 tăng biến i, và trả quyền điều khiển cho khối phân xử 512.

Khối chức năng 524 viết các phần tử cú pháp này vào ít nhất một thông báo SPS, PPS, SEI, phần đầu đơn vị NAL, và phần đầu phiên, và chuyển điều khiển cho khối chức năng 526. Khối chức năng 526 này mã hoá mỗi hình bằng cách sử dụng chuẩn MPEG-4 AVC hoặc bộ mã hoá - giải mã một cảnh khác, và chuyển điều khiển cho khối kết thúc 599.

Fig.6 là sơ đồ thể hiện phương pháp giải mã các hình cho nhiều cảnh bằng cách sử dụng chuẩn MPEG-4 AVC được biểu thị chung bằng số chỉ dẫn 600.

Phương pháp 600 bao gồm khối bắt đầu mà chuyển quyền điều khiển cho khối chức năng 604. Khối chức năng 604 này phân tích các phần tử cú pháp sau đây từ ít nhất một thông báo SPS, PPS, SEI, phần đầu đơn vị NAL, và phần đầu phiên, và chuyển điều khiển cho khối chức năng 606. Khối chức năng 606 này phân tích phần tử cú pháp `num_coded_views_minus1`, và chuyển điều khiển cho khối chức năng 608. Khối chức năng 608 này phân tích phần tử cú pháp `org_pic_width_in_mbs_minus1` và `org_pic_height_in_mbs_minus1`, và chuyển điều khiển cho khối chức năng 610. Khối chức năng 610 này đặt biến `i` bằng số 0, và chuyển điều khiển cho khối phân xử 612. Khối phân xử 612 này xác định xem biến `i` có nhỏ hơn số lượng các cảnh hay không. Nếu có, quyền điều khiển được chuyển cho khối chức năng 614. Nếu không, điều khiển được chuyển cho khối chức năng 624.

Khối chức năng 614 phân tích phần tử cú pháp `view_id[i]`, và chuyển quyền điều khiển cho khối chức năng 616. Khối chức năng 616 này phân tích phần tử cú pháp `num_parts_minus1[view_id[i]]`, và chuyển quyền điều khiển cho khối chức năng 618. Khối chức năng 618 này đặt biến `j` bằng 0 và chuyển quyền điều khiển cho khối phân xử 620. Khối phân xử 620 này xác định xem giá trị hiện thời của biến `j` có nhỏ hơn giá trị hiện thời của phần tử cú pháp `num_parts[view_id[i]]` hay không. Nếu có, quyền điều khiển được chuyển cho khối chức năng 622. Nếu không, điều khiển được chuyển cho khối chức năng 628.

Khối chức năng 622 phân tích các phần tử cú pháp sau, tăng biến `j`, và sau đó trả quyền điều khiển cho khối phân xử 620:

`depth_flag[view_id[i]][j]`; `flip_dir[view_id[i]][j]`; `loc_left_offset[view_id[i]][j]`;
`loc_top_offset[view_id[i]][j]`; `frame_crop_left_offset[view_id[i]][j]`;
`frame_crop_right_offset [view_id [i]][j]`; `frame_crop_top_offset [view_id[i]][j]`; và
`frame_crop_bottom_offset[view_id[i]][j]`.

Khối chức năng 628 phân tích phần tử cú pháp `upsample_view_flag[view_id[i]]`, và chuyển điều khiển cho khối phân xử 630. Khối phân xử 630 này xác định xem giá trị hiện thời của phần tử cú pháp `upsample_view_flag[view_id[i]]` có bằng 1 hay không. Nếu có, điều khiển được chuyển cho khối chức năng 632. Nếu không, điều khiển được chuyển cho khối phân xử 634.

Khối chức năng 632 phân tích phần tử cú pháp `upsample_filter[view_id[i]]`, và chuyển quyền điều khiển cho khối phân xử 634.

Khối phân xử 634 này xác định xem giá trị hiện thời của phần tử cú pháp `upsample_filter[view_id[i]]` có bằng 3 hay không. Nếu có, điều khiển được chuyển cho khối chức năng 636. Nếu không, điều khiển được chuyển cho khối chức năng 640.

Khối chức năng 636 phân tích các phần tử cú pháp sau và chuyển quyền điều khiển cho khối chức năng 638: `vert_dim[view_id[i]]`; `hor_dim[view_id[i]]`; và `quantizer[view_id[i]]`.

Khối chức năng 638 phân tích các hệ số bộ lọc cho mỗi thành phần YUV, và chuyển điều khiển cho khối chức năng 640.

Khối chức năng 640 tăng biến `i`, và trả lại quyền điều khiển cho khối phân xử 612.

Khối chức năng 624 giải mã mỗi hình bằng cách sử dụng chuẩn MPEG-4 AVC hoặc bộ mã hoá-giải mã cảnh đơn khác, và chuyển quyền điều khiển cho khối chức năng 626. Khối chức năng 626 này tách mỗi cảnh từ hình bằng cách sử dụng cú pháp mức cao, và chuyển điều khiển cho khối kết thúc 699.

Fig.7 là sơ đồ thể hiện phương pháp mã hoá các hình và chiều sâu của hình cho nhiều cảnh bằng cách sử dụng chuẩn MPEG-4 AVC được biểu thị chung bằng số chỉ dẫn 700.

Phương pháp 700 bao gồm khối bắt đầu 702 mà chuyển quyền điều khiển cho khối chức năng 704. Khối chức năng 704 này sắp xếp mỗi cảnh và chiều sâu tương ứng tại một nấc thời gian cụ thể như là một hình con theo dạng xếp lát, và chuyển quyền

điều khiển cho khối chức năng 706. Khối chức năng 706 này thiết lập phần tử cú pháp `num_coded_views_minus1`, và chuyển điều khiển cho khối chức năng 708. Khối chức năng 708 này thiết lập các phần tử cú pháp `org_pic_width_in_mbs_minus1` và `org_pic_height_in_mbs_minus1`, và chuyển điều khiển cho khối chức năng 710. Khối chức năng 710 này đặt biến `i` bằng 0, và chuyển điều khiển cho khối phân xử 712. Khối phân xử 712 này xác định xem biến `i` có nhỏ hơn số lượng các cảnh hay không. Nếu có, quyền điều khiển được chuyển cho khối chức năng 714. Nếu không, điều khiển được chuyển cho khối chức năng 724.

Khối chức năng 714 này thiết lập phần tử cú pháp `view_id[i]`, và chuyển điều khiển cho khối chức năng 716. Khối chức năng 716 này thiết lập phần tử cú pháp `num_parts[view_id[i]]`, và chuyển điều khiển cho khối chức năng 718. Khối chức năng 718 này đặt biến `j` bằng 0, và chuyển điều khiển cho khối phân xử 720. Khối phân xử 720 này xác định xem giá trị hiện thời của biến `j` có nhỏ hơn giá trị hiện thời của phần tử cú pháp `num_parts[view_id[i]]` hay không. Nếu có, quyền điều khiển được chuyển cho khối chức năng 722. Nếu không, điều khiển được chuyển cho khối chức năng 728.

Khối chức năng 722 này thiết lập các phần tử cú pháp sau, tăng biến `j`, và sau đó trả lại quyền điều khiển cho khối phân xử 720:

```
depth_flag[view_id[i]][j]; flip_dir[view_id[i]][j]; loc_left_offset[view_id[i]][j];
loc_top_offset[view_id[i]][j]; frame_crop_left_offset[view_id[i]][j];
frame_crop_right_offset[view_id[i]][j]; frame_crop_top_offset[view_id[i]][j]; và
frame_crop_bottom_offset[view_id[i]][j].
```

Khối chức năng 728 thiết lập phần tử cú pháp `upsample_view_flag[view_id[i]]`, và chuyển quyền điều khiển cho khối phân xử 730. Khối phân xử 730 này xác định xem giá trị hiện thời của phần tử cú pháp `upsample_view_flag[view_id[i]]` có bằng 1 hay không. Nếu có, điều khiển được chuyển cho khối chức năng 732. Nếu không, điều khiển được chuyển cho khối phân xử 734.

Khối chức năng 732 thiết lập phần tử cú pháp `upsample_filter[view_id[i]]`, và

chuyển quyền điều khiển khối phân xử 734.

Khối phân xử 734 này xác định xem giá trị hiện thời của phần tử cú pháp `upsample_filter[view_id[i]]` có bằng 3 hay không. Nếu có, quyền điều khiển được chuyển cho khối chức năng 736. Nếu không, điều khiển được chuyển cho khối chức năng 740.

Khối chức năng 736 thiết lập các phần tử cú pháp sau và chuyển quyền điều khiển cho khối chức năng 738: `vert_dim[view_id[i]]`; `hor_dim[view_id[i]]`; và `quantizer[view_id[i]]`.

Khối chức năng 738 thiết lập các hệ số bộ lọc cho mỗi thành phần YUV, và chuyển quyền điều khiển cho khối chức năng 740.

Khối chức năng 740 này tăng biến `i`, và trả lại quyền điều khiển cho khối phân xử 712.

Khối chức năng 724 viết các phần tử cú pháp này vào ít nhất một thông báo SPS, PPS, SEI, phần đầu đơn vị NAL, và phần đầu phiên, và chuyển quyền điều khiển cho khối chức năng 726. Khối chức năng 726 này mã hoá mỗi hình bằng cách sử dụng chuẩn MPEG-4 AVC hoặc bộ mã hoá-giải mã một cảnh khác, chuyển điều khiển cho khối kết thúc 799.

Fig.8 là sơ đồ thể hiện phương pháp giải mã các hình và chiều sâu của hình cho nhiều cảnh bằng cách sử dụng chuẩn MPEG-4 AVC được biểu thị chung bằng số chỉ dẫn 800.

Phương pháp 800 bao gồm khối bắt đầu 802 mà chuyển quyền điều khiển cho khối chức năng 804. Khối chức năng 804 này phân tích các phần tử cú pháp sau đây từ ít nhất một thông báo SPS, PPS, SEI, phần đầu đơn vị NAL, và phần đầu phiên, và chuyển điều khiển cho khối chức năng 806. Khối chức năng 806 này phân tích phần tử cú pháp `num_coded_views_minus1`, và chuyển điều khiển cho khối chức năng 808. Khối chức năng 808 này phân tích các phần tử cú pháp `org_pic_width_in_mbs_minus1` và `org_pic_height_in_mbs_minus1`, và chuyển điều khiển cho khối chức năng 810. Khối

chức năng 810 này đặt biến i bằng 0, và chuyển quyền điều khiển cho khối phân xử 812. Khối phân xử 812 này xác định xem biến I có nhỏ hơn số lượng cảnh hay không. Nếu có, quyền điều khiển được chuyển cho khối chức năng 814. Nếu không, điều khiển được chuyển cho khối chức năng 824.

Khối chức năng 814 phân tích phần tử cú pháp $view_id[i]$, và chuyển quyền điều khiển cho khối chức năng 816. Khối chức năng 816 này phân tích phần tử cú pháp $num_parts_minus1[view_id[i]]$, và chuyển quyền điều khiển cho khối chức năng 818. Khối chức năng 818 này đặt biến j bằng 0, và chuyển điều khiển cho khối phân xử 820. Khối phân xử 820 này xác định xem giá trị hiện thời của biến j có thấp hơn giá trị hiện thời của phần tử cú pháp $num_parts[view_id[i]]$ hay không. Nếu có, quyền điều khiển được chuyển cho khối chức năng 822. Nếu không, điều khiển được chuyển cho khối chức năng 828.

Khối chức năng 822 phân tích các phần tử cú pháp sau, tăng biến j , và sau đó trả lại quyền điều khiển cho khối phân xử 820:

```
depth_flag[view_id[i]][j]; flip_dir[view_id[i]][j]; loc_left_offset[view_id[i]][j];
loc_top_offset[view_id[i]][j]; frame_crop_left_offset[view_id[i]][j];
frame_crop_right_offset[view_id[i]][j]; frame_crop_top_offset[view_id[i]][j]; và
frame_crop_bottom_offset[view_id[i]][j].
```

Khối chức năng 828 phân tích phần tử cú pháp $upsample_view_flag[view_id[i]]$, và chuyển quyền điều khiển cho khối phân xử 830. Khối phân xử 830 này xác định xem giá trị hiện thời của phần tử cú pháp $upsample_view_flag[view_id[i]]$ có bằng 1 hay không. Nếu có, quyền điều khiển được chuyển cho khối chức năng 832. Nếu không, điều khiển được chuyển cho khối phân xử 834.

Khối chức năng 832 phân tích phần tử cú pháp $upsample_filter[view_id[i]]$, và chuyển quyền điều khiển cho khối phân xử 834.

Khối phân xử 834 này xác định xem giá trị hiện thời của phần tử cú pháp $upsample_filter[view_id[i]]$ có bằng 3 hay không. Nếu có, thì điều khiển được chuyển

cho khối chức năng 836. Nếu không, thì điều khiển được chuyển cho khối chức năng 840.

Khối chức năng 836 phân tích các phần tử cú pháp sau đây và chuyển quyền điều khiển cho khối chức năng 838: `vert_dim[view_id[i]]`; `hor_dim[view_id[i]]`; và `quantizer[view_id[i]]`.

Khối chức năng 838 phân tích các hệ số bộ lọc cho mỗi thành phần YUV và chuyển điều khiển cho khối chức năng 840.

Khối chức năng 840 tăng thêm biến *i*, và trả quyền điều khiển cho khối chức năng 812.

Khối chức năng 824 giải mã mỗi hình bằng cách sử dụng chuẩn MPEG-4 AVC hoặc bộ mã hoá-giải mã một cảnh khác, và chuyển điều khiển cho khối chức năng 826. Khối chức năng 826 tách mỗi cảnh và chiều sâu tương ứng từ hình bằng cách sử dụng cú pháp mức cao, và chuyển quyền điều khiển cho khối chức năng 827. Khối chức năng 827 này thực hiện tổng hợp cảnh tiềm tàng bằng cách sử dụng cảnh được trích, và chuyển điều khiển cho khối kết thúc 899.

Đối với chiều sâu được dùng trong các Fig.7 và Fig.8, Fig.9 thể hiện một ví dụ về tín hiệu chiều sâu 900, tại đó chiều sâu được tạo ra như giá trị điểm ảnh cho mỗi vị trí tương ứng của ảnh (không thể hiện). Hơn nữa, Fig.10 thể hiện một ví dụ về hai tín hiệu chiều sâu có trong lát 1000. Phần trên bên phải lát 1000 là tín hiệu chiều sâu có các giá trị chiều sâu tương ứng với ảnh phía trên bên trái lát 1000. Phần dưới bên phải lát 1000 là tín hiệu chiều sâu có các giá trị chiều sâu tương ứng với ảnh phía dưới bên trái lát 1000.

Fig.11 là sơ đồ thể hiện một ví dụ về 5 cảnh được xếp lát trên một khung được biểu thị chung bằng số chỉ dẫn 1100. Bốn cảnh trên nằm theo hướng chuẩn. Cảnh thứ năm cũng nằm theo hướng chuẩn, nhưng được tách thành hai phần dọc theo phía dưới của lát 1100. Phần bên trái của cảnh thứ năm thể hiện “phía trên” của cảnh thứ năm, và phần bên phải của cảnh thứ năm thể hiện “phía dưới” của cảnh thứ năm này.

Mã hoá/giải mã bằng cách sử dụng chuẩn/quy định mã hoá/giải mã video đa cảnh

Fig.12 là sơ đồ khối thể hiện bộ mã hoá mã hoá video đa cảnh (MVC - Multi-view Video Coding) được biểu thị chung bằng số chỉ dẫn 1200. Bộ mã hóa 1200 bao gồm bộ kết hợp 1205 có đầu ra được nối truyền thông tín hiệu với đầu vào của bộ biến đổi 1210. Đầu ra của bộ biến đổi 1210 này được nối truyền thông tín hiệu với đầu vào của bộ lượng tử hoá 1215. Đầu ra của bộ lượng tử hoá 1215 này được nối truyền thông tín hiệu với đầu vào của bộ mã hoá entropy 1220 và đầu vào của bộ lượng tử hoá ngược 1225. Đầu ra của bộ lượng tử hoá ngược 1225 này được nối truyền thông tín hiệu với đầu vào của bộ biến đổi ngược 1230. Đầu ra của bộ biến đổi ngược 1230 này được nối truyền thông tín hiệu với đầu vào không đảo thứ nhất của bộ kết hợp 1235. Đầu ra của bộ kết hợp 1235 này được nối truyền thông tín hiệu với đầu vào của bộ dự báo bên trong 1245 và đầu vào của bộ lọc tách khối 1250. Đầu ra của bộ lọc tách khối 1250 được nối truyền thông tín hiệu với đầu vào của kho hình tham chiếu 1255 (đối với cảnh i). Đầu ra của kho hình tham chiếu 1255 này được nối truyền thông tín hiệu với đầu vào thứ nhất của bộ bù chuyển động 1275 và đầu vào thứ nhất của bộ đánh giá chuyển động 1280. Đầu ra của bộ đánh giá chuyển động 1280 này được nối truyền thông tín hiệu với đầu vào thứ hai của bộ bù chuyển động 1275.

Đầu ra của kho hình tham chiếu 1260 (cho các cảnh khác) được nối truyền thông tín hiệu với đầu vào thứ nhất của bộ đánh giá chênh lệch 1270 và đầu vào thứ nhất của bộ bù chênh lệch 1265. Đầu ra của bộ đánh giá chênh lệch 1270 này được nối truyền thông tín hiệu với đầu vào thứ hai của bộ bù chênh lệch 1265.

Đầu ra của bộ giải mã entropy 1220 khả dụng như đầu ra của bộ mã hóa 1200. Đầu vào không đảo của bộ kết hợp 1205 khả dụng như đầu vào của bộ mã hóa 1200, và được nối truyền thông tín hiệu với đầu vào thứ hai của bộ đánh giá chênh lệch 1270, và đầu vào thứ hai của bộ đánh giá chuyển động 1280. Đầu ra của bộ chuyển 1285 được nối truyền thông tín hiệu với đầu vào không đảo thứ hai của bộ kết hợp 1235 và với đầu vào ngược của bộ kết hợp 1205. Bộ chuyển 1285 bao gồm đầu vào thứ nhất được nối truyền thông tín hiệu với đầu ra của bộ bù chuyển động 1275, đầu vào thứ hai được nối truyền

thông tin hiệu với đầu ra của bộ bù chênh lệch 1265, và đầu vào thứ ba được nối truyền thông tin hiệu với đầu ra của bộ dự báo bên trong 1245.

Mô-đun phân xử chế độ 1240 có đầu ra nối với bộ chuyển 1285 để điều khiển đầu vào nào được lựa chọn bởi bộ chuyển 1285 này.

Fig.13 là sơ đồ khối thể hiện bộ giải mã MVC làm ví dụ được biểu thị chung bằng số chỉ dẫn 1300. Bộ giải mã 1300 bao gồm bộ giải mã entropy 1305 có đầu ra được nối truyền thông tin hiệu với đầu vào của bộ lượng tử hoá ngược 1310. Đầu ra của bộ lượng tử hoá ngược này được nối truyền thông tin hiệu với đầu vào của bộ biến đổi ngược 1315. Đầu ra của bộ biến đổi ngược 1315 này được nối truyền thông tin hiệu với đầu vào không ngược thứ nhất của bộ kết hợp 1320. Đầu ra của bộ kết hợp 1320 này được nối truyền thông tin hiệu với đầu vào của bộ lọc tách khối 1325 và đầu vào của bộ dự báo bên trong 1330. Đầu ra của bộ lọc tách khối 1325 được nối truyền thông tin hiệu với đầu vào của kho hình tham chiếu 1340 (cho cảnh i). Đầu ra của kho hình tham chiếu 1340 này được nối truyền thông tin hiệu với đầu vào thứ nhất của bộ bù chuyển động 1335.

Đầu ra của kho hình tham chiếu 1345 (cho các cảnh khác) được nối truyền thông tin hiệu với đầu vào thứ nhất của bộ bù chênh lệch 1350.

Đầu vào của bộ mã hoá entropy 1305 khả dụng như đầu vào của bộ giải mã 1300, để nhận luồng bit dư. Hơn nữa, đầu vào của mô-đun chế độ 1360 cũng khả dụng như đầu vào của bộ giải mã 1300, để nhận cú pháp điều khiển để điều khiển đầu vào nào được lựa chọn bởi bộ chuyển 1355. Hơn nữa, đầu vào thứ hai của bộ bù chuyển động 1335 khả dụng như đầu vào của bộ giải mã 1300, để nhận các vector chuyển động. Ngoài ra, đầu vào thứ hai của bộ bù chênh lệch 1350 khả dụng như đầu vào của bộ giải mã 1300, để nhận các vector chênh lệch.

Đầu ra của bộ chuyển 1355 được nối truyền thông tin hiệu với đầu vào không ngược thứ hai của bộ kết hợp 1320. Đầu vào thứ nhất của bộ chuyển 1355 được nối truyền thông tin hiệu với đầu ra của bộ bù chênh lệch 1350. Đầu vào thứ hai của bộ chuyển 1355 được nối truyền thông tin hiệu với đầu ra của bộ bù chuyển động 1335.

Đầu vào thứ ba của bộ chuyển 1355 này được nối truyền thông tín hiệu với đầu ra của bộ dự báo bên trong 1330. Đầu ra của mô-đun chế độ 1360 được nối truyền thông tín hiệu với bộ chuyển 1355 để điều khiển đầu vào nào được chọn bởi bộ chuyển 1355 này. Đầu ra của bộ lọc tách khối 1325 khả dụng như đầu ra của bộ giải mã 1300.

Fig.14 là sơ đồ thể hiện phương pháp xử lý các hình cho nhiều cảnh để chuẩn bị mã hoá các hình này bằng cách sử dụng phần mở rộng MVC của chuẩn MPEG-4 AVC được biểu thị chung bằng số chỉ dẫn 1400.

Phương pháp 1400 bao gồm khối bắt đầu 1405 mà chuyển quyền điều khiển cho khối chức năng 1410. Khối chức năng 1410 này sắp xếp N cảnh, trong toàn bộ M cảnh, tại một lúc thời gian cụ thể như một siêu hình theo kiểu xếp lát, và chuyển điều khiển cho khối chức năng 1415. Khối chức năng 1415 này thiết lập phần tử cú pháp `num_coded_views_minus1` và chuyển quyền điều khiển cho khối chức năng 1420. Khối chức năng 1420 này thiết lập phần tử cú pháp `view_id[i]` cho tất cả các cảnh (`num_coded_views_minus1 + 1`), và chuyển điều khiển cho khối chức năng 1425. Khối chức năng 1425 này thiết lập thông tin phụ thuộc tham chiếu giữa các cảnh cho các hình neo, và chuyển điều khiển cho khối chức năng 1430. Khối chức năng 1430 này thiết lập thông tin phụ thuộc tham chiếu giữa các cảnh cho các hình không neo, và chuyển điều khiển cho khối chức năng 1435. Khối chức năng 1435 này thiết lập phần tử cú pháp `pseudo_view_present_flag`, và chuyển quyền điều khiển cho khối phân xử 1440. Khối phân xử 1440 này xác định xem giá trị hiện thời của phần tử cú pháp `pseudo_view_present_flag` có đúng hay không. Nếu đúng, quyền điều khiển được chuyển cho khối chức năng 1445. Nếu sai, điều khiển được chuyển cho khối kết thúc 1499.

Khối chức năng 1445 thiết lập các phần tử cú pháp sau đây, và chuyển quyền điều khiển cho khối chức năng 1450: `tiling_mode`; `org_pic_width_in_mbs_minus1`; và `org_pic_height_in_mbs_minus1`. Khối chức năng 1450 gọi phần tử cú pháp `pseudo_view_info(view_id)` cho mỗi cảnh mã hóa, và chuyển điều khiển cho khối kết thúc 1499.

Fig.15 là sơ đồ thể hiện phương pháp mã hoá các hình cho nhiều cảnh bằng cách sử dụng phần mở rộng MVC của chuẩn MPEG-4 AVC và được biểu thị chung bằng số chỉ dẫn 1500.

Phương pháp 1500 bao gồm khối bắt đầu 1502 với tham số đầu vào `pseudo_view_id` và chuyển quyền điều khiển cho khối chức năng 1504. Khối chức năng 1504 này thiết lập phần tử cú pháp `num_sub_views_minus1`, và chuyển điều khiển cho khối chức năng 1506. Khối chức năng 1506 này đặt biến `i` bằng 0, và chuyển điều khiển cho khối phân xử 1508. Khối phân xử 1508 này xác định xem biến `i` có nhỏ hơn số lượng các `sub_view` hay không. Nếu có, thì chuyển quyền điều khiển cho khối chức năng 1510. Nếu không, điều khiển được chuyển cho khối chức năng 1520.

Khối chức năng 1510 thiết lập phần tử cú pháp `sub_view_id[i]`, và chuyển điều khiển cho khối chức năng 1512. Khối chức năng 1512 này thiết lập phần tử cú pháp `num_parts_minus1[sub_view_id[i]]`, và chuyển điều khiển cho khối chức năng 1514. Khối chức năng 1514 này đặt biến `j` bằng 0, và chuyển quyền điều khiển cho khối phân xử 1516. Khối phân xử 1516 này xác định xem biến `j` có nhỏ hơn phần tử cú pháp `num_parts_minus1[sub_view_id[i]]` hay không. Nếu có, quyền điều khiển được chuyển cho khối chức năng 1518. Nếu không, điều khiển được chuyển cho khối phân xử 1522.

Khối chức năng 1518 thiết lập các phần tử cú pháp sau, tăng biến `j`, và trả lại quyền điều khiển cho khối phân xử 1516:

```
loc_left_offset[sub_view_id[i]][j]; loc_top_offset[sub_view_id[i]][j];
frame_crop_left_offset[sub_view_id[i]][j]; frame_crop_right_offset[sub_view_id[i]][j];
frame_crop_top_offset[sub_view_id[i]][j]; và frame_crop_bottom_offset
[sub_view_id[i]][j].
```

Khối chức năng 1520 mã hóa hình hiện thời cho cảnh hiện thời bằng cách sử dụng kỹ thuật MVC, và chuyển điều khiển cho khối kết thúc 1599.

Khối phân xử 1522 xác định xem phần tử cú pháp `tiling_mode` có bằng 0 hay không. Nếu có, quyền điều khiển được chuyển cho khối chức năng 1524. Nếu

không, điều khiển được chuyển cho khối chức năng 1538.

Khối chức năng 1524 thiết lập phần tử cú pháp flip_dir[sub_view_id[i]] và phần tử cú pháp upsample_view_flag[sub_view_id[i]], và chuyển điều khiển cho khối phân xử 1526. Khối phân xử 1526 này xác định xem giá trị hiện thời của phần tử cú pháp upsample_view_flag[sub_view_id[i]] có bằng 1 hay không. Nếu có, điều khiển được chuyển cho khối chức năng 1528. Nếu không, điều khiển được chuyển cho khối phân xử 1530.

Khối chức năng 1528 thiết lập phần tử cú pháp upsample_filter[sub_view_id[i]], và chuyển quyền điều khiển cho khối phân xử 1530. Khối phân xử 1530 này xác định xem giá trị phần tử cú pháp upsample_filter[sub_view_id[i]] có bằng 3 hay không. Nếu có, điều khiển được chuyển cho khối chức năng 1532. Nếu không, điều khiển được chuyển cho khối chức năng 1536.

Khối chức năng 1532 thiết lập các phần tử cú pháp sau đây, và chuyển quyền điều khiển cho khối chức năng 1534: vert_dim[sub_view_id[i]]; hor_dim[sub_view_id[i]]; và quantizer[sub_view_id[i]]. Khối chức năng 1534 thiết lập các hệ số bộ lọc cho mỗi thành phần YUV, và chuyển điều khiển cho khối chức năng 1536.

Khối chức năng 1536 tăng biến i, và trả lại quyền điều khiển cho khối phân xử 1508.

Khối chức năng 1538 thiết lập phần tử cú pháp pixel_dist_x[sub_view_id[i]] và phần tử cú pháp flip_dist_y[sub_view_id[i]], và chuyển điều khiển cho khối chức năng 1540. Khối chức năng 1540 này đặt biến j bằng 0, và chuyển điều khiển cho khối phân xử 1542. Khối phân xử 1542 này xác định xem giá trị hiện thời của biến j có nhỏ hơn giá trị hiện thời của phần tử cú pháp num_parts[sub_view_id[i]] hay không. Nếu có, quyền điều khiển được chuyển cho khối chức năng 1544. Nếu không, điều khiển được chuyển cho khối chức năng 1536.

Khối chức năng 1544 thiết lập phần tử cú pháp num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]], và chuyển điều khiển cho khối

chức năng 1546. Khối chức năng 1546 này thiết lập các hệ số cho tất cả các bộ lọc xếp lát điểm ảnh, và chuyển điều khiển cho khối chức năng 1536.

Fig.16 là sơ đồ thể hiện phương pháp xử lý các hình cho nhiều cảnh để chuẩn bị giải mã các hình bằng cách sử dụng phần mở rộng MVC của chuẩn MPEG-4 AVC và được biểu thị chung bằng số chỉ dẫn 1600.

Phương pháp 1600 bao gồm khối bắt đầu 1605 mà chuyển quyền điều khiển cho khối chức năng 1615. Khối chức năng 1615 này phân tích phần tử cú pháp `num_coded_views_minus1`, và chuyển điều khiển cho khối chức năng 1620. Khối chức năng 1620 này phân tích phần tử cú pháp `view_id[i]` cho tất cả các cảnh (`num_coded_views_minus1 + 1`), và chuyển điều khiển cho khối chức năng 1625. Khối chức năng 1625 này phân tích thông tin phụ thuộc tham chiếu giữa các cảnh cho các hình neo, và chuyển quyền điều khiển cho khối chức năng 1630. Khối chức năng 1630 này phân tích thông tin phụ thuộc tham chiếu giữa các cảnh cho các hình không neo, và chuyển điều khiển cho khối chức năng 1635. Khối chức năng 1635 này phân tích phần tử cú pháp `pseudo_view_present_flag`, và chuyển điều khiển cho khối phân xử 1640. Khối phân xử 1640 này xác định xem giá trị hiện thời của phần tử cú pháp `pseudo_view_present_flag` có đúng hay không. Nếu đúng, điều khiển được chuyển cho khối chức năng 1645. Nếu sai, điều khiển được chuyển cho khối kết thúc 1699.

Khối chức năng 1645 phân tích các phần tử cú pháp sau đây, và chuyển quyền điều khiển cho khối chức năng 1650: `tiling_mode`; `org_pic_width_in_mbs_minus1`; và `org_pic_height_in_mbs_minus1`. Khối chức năng 1650 gọi phần tử cú pháp `pseudo_view_info(view_id)` cho mỗi cảnh được mã hoá, và chuyển điều khiển cho khối kết thúc 1699.

Fig.17 là sơ đồ thể hiện phương pháp giải mã các hình cho nhiều cảnh bằng cách sử dụng phần mở rộng MVC của chuẩn MPEG-4 AVC được biểu thị chung bằng số chỉ dẫn 1700.

Phương pháp 1700 bao gồm khối bắt đầu 1702 bắt đầu với tham số đầu vào

pseudo_view_id và chuyển quyền điều khiển cho khối chức năng 1704. Khối chức năng 1704 này phân tích phần tử cú pháp num_sub_views_minus1, và chuyển điều khiển cho khối chức năng 1706. Khối chức năng 1706 này đặt biến i bằng 0, và chuyển điều khiển cho khối 1708. Khối phân xử 1708 này xác định xem biến i có nhỏ hơn số lượng các sub_views hay không. Nếu có, điều khiển được chuyển cho khối chức năng 1710. Nếu không, điều khiển được chuyển cho khối chức năng 1720.

Khối chức năng 1710 phân tích phần tử cú pháp sub_view_id[i], và chuyển quyền điều khiển cho khối chức năng 1712. Khối chức năng 1712 này phân tích phần tử cú pháp num_parts_minus1[sub_view_id[i]], và chuyển điều khiển cho khối chức năng 1714. Khối chức năng 1714 này đặt biến j bằng 0, và chuyển điều khiển cho khối phân xử 1716. Khối phân xử 1716 này xác định xem biến j có nhỏ hơn phần tử cú pháp num_parts_minus1[sub_view_id[i]] hay không. Nếu có, thì chuyển quyền điều khiển cho khối chức năng 1718. Nếu không, điều khiển được chuyển cho khối phân xử 1722.

Khối chức năng 1718 thiết lập các phần tử cú pháp sau, tăng biến j, và trả lại quyền điều khiển cho khối phân xử 1716:

```
loc_left_offset[sub_view_id[i]][j]; loc_top_offset[sub_view_id[i]][j];
frame_crop_left_offset[sub_view_id[i]][j];

frame_crop_right_offset[sub_view_id[i]][j];
frame_crop_top_offset[sub_view_id[i]][j]; và
frame_crop_bottom_offset[sub_view_id[i]][j].
```

Khối chức năng 1720 giải mã hình hiện thời cho cảnh hiện thời bằng cách sử dụng kỹ thuật mã hoá video đa cảnh, và chuyển điều khiển cho khối chức năng 1721. Khối chức năng 1721 này tách mỗi cảnh từ hình bằng cách sử dụng cú pháp mức cao, và chuyển điều khiển cho khối kết thúc 1799.

Việc tách mỗi cảnh từ hình được giải mã được thực hiện bằng cách sử dụng cú pháp mức cao biểu thị trong luồng bit. Cú pháp mức cao này có thể biểu thị vị trí chính xác và có thể sử dụng hướng của các cảnh (và có thể sử dụng chiều sâu tương ứng) có

trong hình.

Khối phân xử 1722 xác định xem phần tử cú pháp `tiling_mode` có bằng 0 hay không. Nếu bằng, thì quyền điều khiển được chuyển cho khối chức năng 1724. Nếu không, điều khiển được chuyển cho khối chức năng 1738.

Khối chức năng 1724 phân tích phần tử cú pháp `flip_dir[sub_view_id[i]]` và phần tử cú pháp `upsample_view_flag[sub_view_id[i]]`, và chuyển điều khiển cho khối phân xử 1726. Khối phân xử 1726 này xác định xem giá trị hiện thời của phần tử cú pháp `upsample_view_flag[sub_view_id[i]]` có bằng 1 hay không. Nếu có, thì điều khiển được chuyển cho khối chức năng 1728. Nếu không, điều khiển được chuyển cho khối phân xử 1730.

Khối chức năng 1728 phân tích phần tử cú pháp `upsample_filter[sub_view_id[i]]`, và chuyển quyền điều khiển cho khối phân xử 1730. Khối phân xử 1730 này xác định xem giá trị của phần tử cú pháp `upsample_filter[sub_view_id[i]]` có bằng 3 hay không. Nếu bằng, điều khiển được chuyển cho khối chức năng 1732. Nếu không, điều khiển được chuyển cho khối chức năng 1736.

Khối chức năng 1732 phân tích các phần tử cú pháp sau đây, và chuyển quyền điều khiển cho khối chức năng 1734: `vert_dim[sub_view_id[i]]`; `hor_dim[sub_view_id[i]]`; và `quantizer[sub_view_id[i]]`. Khối chức năng 1734 phân tích các hệ số bộ lọc cho mỗi thành phần YUV, và chuyển điều khiển cho khối chức năng 1736.

Khối chức năng 1736 này tăng biến `i`, và trả lại quyền điều khiển cho khối phân xử 1708.

Khối chức năng 1738 phân tích phần tử cú pháp `pixel_dist_x[sub_view_id[i]]` và phần tử cú pháp `flip_dist_y[sub_view_id[i]]`, và chuyển điều khiển cho khối chức năng 1740. Khối chức năng 1740 này đặt biến `j` bằng 0, và chuyển điều khiển cho khối phân xử 1742. Khối phân xử 1742 này xác định xem giá trị hiện thời của biến `j` có nhỏ hơn giá trị hiện thời của phần tử cú pháp `num_parts[sub_view_id[i]]` hay không. Nếu có, thì điều khiển được chuyển cho khối chức năng 1744. Nếu không, điều khiển được chuyển cho

khối chức năng 1736.

Khối chức năng 1744 phân tích phần tử cú pháp `num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]]`, và chuyển điều khiển cho khối chức năng 1746. Khối chức năng 1776 phân tích các hệ số cho tất cả các bộ lọc xếp lát điểm ảnh, và chuyển điều khiển cho khối chức năng 1736.

Fig.18 là sơ đồ thể hiện phương pháp xử lý các hình cho nhiều cảnh và chiều sâu của cảnh để chuẩn bị mã hoá các hình bằng cách sử dụng phần mở rộng MVC của chuẩn MPEG-4 AVC được biểu thị chung bằng số chỉ dẫn 1800.

Phương pháp 1800 bao gồm khối bắt đầu 1805 mà bắt đầu chuyển quyền điều khiển cho khối chức năng 1810. Khối chức năng 1810 này sắp xếp N cảnh và bản đồ chiều sâu, trong số toàn bộ M cảnh và bản đồ chiều sâu, tại một nấc thời gian cụ thể là siêu hình có dạng xếp lát, và chuyển quyền điều khiển cho khối chức năng 1815. Khối chức năng 1815 này thiết lập phần tử cú pháp `num_coded_views_minus1`, và chuyển điều khiển cho khối chức năng 1820. Khối chức năng 1820 này thiết lập phần tử cú pháp `view_id[i]` cho tất cả chiều sâu (`num_coded_views_minus1 + 1`) tương ứng với `view_id[i]`, và chuyển điều khiển cho khối chức năng 1825. Khối chức năng 1825 này thiết lập thông tin phụ thuộc tham chiếu giữa các cảnh cho hình chiều sâu neo, và chuyển điều khiển cho khối chức năng 1830. Khối chức năng 1830 này thiết lập thông tin phụ thuộc tham chiếu giữa các cảnh cho các hình chiều sâu không neo, và chuyển điều khiển cho khối chức năng 1835. Khối chức năng 1835 thiết lập phần tử cú pháp `pseudo_view_present_flag`, và chuyển điều khiển cho khối phân xử 1840. Khối phân xử 1840 này xác định xem giá trị hiện thời của phần tử cú pháp `pseudo_view_present_flag` có đúng hay không. Nếu có, thì điều khiển được chuyển cho khối chức năng 1845. Nếu không, điều khiển được chuyển cho khối kết thúc 1899.

Khối chức năng 1845 thiết lập các phần tử cú pháp sau đây, và chuyển quyền điều khiển cho khối chức năng 1850: `tiling_mode`; `org_pic_width_in_mbs_minus1`; và `org_pic_height_in_mbs_minus1`. Khối chức năng 1850 gọi phần tử cú pháp

pseudo_view_info(view_id) cho mỗi cảnh mã hóa, và chuyển điều khiển cho khối kết thúc 1899.

Fig.19 là sơ đồ thể hiện phương pháp mã hoá các hình cho nhiều cảnh và chiều sâu của cảnh bằng cách sử dụng phần mở rộng mã hoá video đa cảnh (MVC) của chuẩn MPEG-4 AVC được biểu thị chung bằng số chỉ dẫn 1900.

Phương pháp 1900 bao gồm khối bắt đầu 1902 mà chuyển quyền điều khiển cho khối chức năng 1904. Khối chức năng 1904 thiết lập phần tử cú pháp num_sub_views_minus1, và chuyển điều khiển cho khối chức năng 1906. Khối chức năng 1906 này đặt biến i bằng 0, và chuyển điều khiển cho khối phân xử 1908. Khối phân xử 1908 này xác định xem biến i có nhỏ hơn số lượng các sub_views hay không. Nếu có, thì điều khiển được chuyển cho khối chức năng 1910. Nếu không, điều khiển được chuyển cho khối chức năng 1920.

Khối chức năng 1910 thiết lập phần tử cú pháp sub_view_id[i], và chuyển điều khiển cho khối chức năng 1912. Khối chức năng 1912 này thiết lập phần tử cú pháp num_parts_minus1[sub_view_id[i]], và chuyển điều khiển cho khối chức năng 1914. Khối chức năng 1914 này đặt biến j bằng 0, và chuyển điều khiển cho khối phân xử 1916. Khối phân xử 1916 xác định biến j có nhỏ hơn phần tử cú pháp num_parts_minus1[sub_view_id[i]] hay không. Nếu có, thì điều khiển được chuyển cho khối chức năng 1918. Nếu không, điều khiển được chuyển cho khối phân xử 1922.

Khối chức năng 1918 thiết lập các phần tử cú pháp sau, tăng biến j, và trả lại quyền điều khiển cho khối phân xử 1916:

```
loc_left_offset[sub_view_id[i]][j];
loc_top_offset[sub_view_id[i]][j];
frame_crop_left_offset[sub_view_id[i]][j];
frame_crop_right_offset[sub_view_id[i]][j];
```

`frame_crop_top_offset[sub_view_id[i]][j]`; và

`frame_crop_bottom_offset[sub_view_id[i]][j]`.

Khối chức năng 1920 mã hoá chiều sâu hiện thời cho cảnh hiện thời bằng cách sử dụng kỹ thuật MVC, và chuyển quyền điều khiển cho khối kết thúc 1999. Tín hiệu chiều sâu có thể được mã hóa tương tự như cách tín hiệu video tương ứng của nó được mã hóa. Ví dụ, tín hiệu chiều sâu của một cảnh có thể được vào trong một lát mà chỉ chứa các tín hiệu chiều sâu khác, hoặc chỉ có các tín hiệu video, hoặc cả tín hiệu chiều sâu lẫn tín hiệu video. Lớp (cảnh giả) sau đó được xử lý như một cảnh của MVC, và cũng có thể có những lát khác xử lý như các cảnh khác của MVC.

Khối phân xử 1922 xác định xem phần tử cú pháp `tiling_mode` có bằng 0 hay không. Nếu có, thì điều khiển được chuyển cho khối chức năng 1924. Nếu không, điều khiển được chuyển cho khối chức năng 1938.

Khối chức năng 1924 thiết lập phần tử cú pháp `flip_dir[sub_view_id[i]]` và phần tử cú pháp `upsample_view_flag[sub_view_id[i]]`, và chuyển điều khiển cho khối phân xử 1926. Khối phân xử 1926 này xác định xem giá trị hiện thời của phần tử cú pháp `upsample_view_flag[sub_view_id[i]]` có bằng 1 hay không. Nếu có, thì điều khiển được chuyển cho khối chức năng 1928. Nếu không, điều khiển được chuyển cho khối phân xử 1930.

Khối chức năng 1928 thiết lập phần tử cú pháp `upsample_filter[sub_view_id[i]]`, và chuyển quyền điều khiển cho khối phân xử 1930. Khối phân xử 1930 xác định xem giá trị của phần tử cú pháp `upsample_filter[sub_view_id[i]]` có bằng 3 hay không. Nếu có, điều khiển được chuyển cho khối chức năng 1932. Nếu không, điều khiển được chuyển cho khối chức năng 1936.

Khối chức năng 1932 thiết lập các phần tử cú pháp sau, và chuyển quyền điều khiển cho khối chức năng 1934: `vert_dim[sub_view_id[i]]`; `hor_dim[sub_view_id[i]]`; và `quantizer[sub_view_id[i]]`. Khối chức năng 1934 thiết lập các hệ số bộ lọc cho mỗi thành phần YUV, và chuyển điều khiển cho khối chức năng 1936.

Khối chức năng 1936 tăng biến i , và trả lại quyền điều khiển cho khối phân xử 1908.

Khối chức năng 1938 thiết lập phần tử cú pháp $\text{pixel_dist_x}[\text{sub_view_id}[i]]$ và phần tử cú pháp $\text{flip_dist_y}[\text{sub_view_id}[i]]$, và chuyển quyền điều khiển cho khối chức năng 1940. Khối chức năng 1940 này đặt biến j bằng 0, và chuyển điều khiển cho khối phân xử 1942. Khối phân xử 1942 này xác định xem giá trị hiện thời của biến j có nhỏ hơn giá trị hiện thời của phần tử cú pháp $\text{num_parts}[\text{sub_view_id}[i]]$ hay không. Nếu có, thì điều khiển được chuyển cho khối chức năng 1944. Nếu không, điều khiển được chuyển cho khối chức năng 1936.

Khối chức năng 1944 thiết lập phần tử cú pháp $\text{num_pixel_tiling_filter_coeffs_minus1}[\text{sub_view_id}[i]]$, và chuyển điều khiển cho khối chức năng 1946. Khối chức năng 1946 này thiết lập các hệ số cho tất cả các bộ lọc xếp lát điểm ảnh, và chuyển điều khiển cho khối chức năng 1936.

Fig.20 là sơ đồ thể hiện phương pháp xử lý các hình cho nhiều cảnh và chiều sâu của cảnh để chuẩn bị giải mã các hình bằng cách sử dụng phần mở rộng mã hoá video đa cảnh (MVC) của chuẩn MPEG-4 AVC được biểu thị chung bằng số chỉ dẫn 2000.

Phương pháp 2000 bao gồm khối bắt đầu năm 2005 mà chuyển quyền điều khiển cho khối chức năng 2015. Khối chức năng 2015 này phân tích phần tử cú pháp $\text{num_coded_views_minus1}$, và chuyển điều khiển cho khối chức năng 2020. Khối chức năng 2020 này phân tích phần tử cú pháp $\text{view_id}[i]$ cho tất cả chiều sâu $(\text{num_coded_views_minus1} + 1)$ tương ứng với $\text{view_id}[i]$, và chuyển điều khiển cho khối chức năng 2025. Khối chức năng 2025 này phân tích thông tin phụ thuộc tham chiếu giữa các cảnh của các hình chiều sâu neo, và chuyển điều khiển cho khối chức năng 2030. Khối chức năng 2030 phân tích thông tin phụ thuộc tham chiếu giữa các cảnh của các hình chiều sâu không neo, và chuyển điều khiển cho khối chức năng 2035. Khối chức năng 2035 này phân tích phần tử cú pháp $\text{pseudo_view_present_flag}$, và chuyển điều khiển cho khối phân xử 2040. Khối phân xử 2040 xác định xem giá trị hiện

thời của phần tử cú pháp `pseudo_view_present_flag` có đúng hay không. Nếu đúng, thì điều khiển được chuyển cho khối chức năng 2045. Nếu không, điều khiển được chuyển cho khối kết thúc 2099.

Khối chức năng 2045 phân tích các phần tử cú pháp sau đây, và chuyển điều khiển cho khối chức năng 2050: `tiling_mode`; `org_pic_width_in_mbs_minus1`; và `org_pic_height_in_mbs_minus1`. Khối chức năng 2050 gọi phần tử cú pháp `pseudo_view_info(view_id)` cho mỗi cảnh mã hóa, và chuyển điều khiển cho khối kết thúc 2099.

Fig.21 là sơ đồ thể hiện phương pháp giải mã các hình cho nhiều cảnh và chiều sâu của cảnh bằng cách sử dụng phần mở rộng mã hoá video đa cảnh (MVC) của chuẩn MPEG-4 AVC được biểu thị chung bằng số chỉ dẫn 2100.

Phương pháp 2100 bao gồm khối bắt đầu 2102 mà bắt đầu với tham số đầu vào `pseudo_view_id`, và chuyển quyền điều khiển cho khối chức năng 2104. Khối chức năng 2104 này phân tích phần tử cú pháp `num_sub_views_minus1`, và chuyển điều khiển cho khối chức năng 2106. Khối chức năng 2106 này đặt biến `i` bằng 0, và chuyển điều khiển cho khối phân xử 2108. Khối phân xử 2108 này xác định xem biến `i` có nhỏ hơn số lượng các `sub_views` hay không. Nếu có, thì điều khiển được chuyển cho khối chức năng 2110. Nếu không, điều khiển được chuyển cho khối chức năng 2120.

Khối chức năng 2110 phân tích phần tử cú pháp `sub_view_id[i]`, và chuyển điều khiển cho khối chức năng 2112. Khối chức năng 2112 này phân tích phần tử cú pháp `num_parts_minus1[sub_view_id[i]]`, và chuyển điều khiển cho khối chức năng 2114. Khối chức năng 2114 này đặt biến `j` bằng 0, và chuyển điều khiển cho khối phân xử 2116. Khối phân xử 2116 xác định xem biến `j` có nhỏ hơn phần tử cú pháp `num_parts_minus1[sub_view_id[i]]` hay không. Nếu có, thì điều khiển được chuyển cho khối chức năng 2118. Nếu không, điều khiển được chuyển cho khối phân xử 2122.

Khối chức năng 2118 thiết lập các phần tử cú pháp sau, tăng biến `j`, và trả quyền điều khiển cho khối phân xử 2116:


```

loc_left_offset[sub_view_id[i]][j]; loc_top_offset[sub_view_id[i]][j];

frame_crop_left_offset[sub_view_id[i]][j]; frame_crop_right_offset[sub_view_id[
i]][j];

frame_crop_top_offset[sub_view_id[i]][j]; và

frame_crop_bottom_offset[sub_view_id[i]][j].

```

Khối chức năng 2120 giải mã hình hiện thời bằng cách sử dụng kỹ thuật mã hoá video đa cảnh, và chuyển điều khiển cho khối chức năng 2121. Khối chức năng 2121 này tách mỗi cảnh từ hình bằng cách sử dụng cú pháp mức cao, và chuyển điều khiển cho khối kết thúc 2199. Việc tách mỗi cảnh bằng cách sử dụng cú pháp mức cao được mô tả như trước đây.

Khối phân xử 2122 xác định xem phần tử cú pháp `tiling_mode` có bằng 0 hay không. Nếu có, thì quyền điều khiển được chuyển cho khối chức năng 2124. Nếu không, điều khiển được chuyển cho khối chức năng 2138.

Khối chức năng 2124 phân tích phần tử cú pháp `flip_dir[sub_view_id [i]]` và phần tử cú pháp `upsample_view_flag[sub_view_id[i]]`, và chuyển điều khiển cho khối phân xử 2126. Khối phân xử 2126 này xác định xem giá trị hiện thời của phần tử cú pháp `upsample_view_flag[sub_view_id[i]]` có bằng 1 hay không. Nếu bằng, thì điều khiển được chuyển cho khối chức năng 2128. Nếu không, điều khiển được chuyển cho khối phân xử 2130.

Khối chức năng 2128 phân tích phần tử cú pháp `upsample_filter[sub_view_id[i]]`, và chuyển quyền điều khiển cho khối phân xử 2130. Khối phân xử 2130 này xác định xem giá trị của phần tử cú pháp `upsample_filter[sub_view_id[i]]` có bằng 3 hay không. Nếu có, điều khiển được chuyển cho khối chức năng 2132. Nếu không, điều khiển được chuyển cho khối chức năng 2136.

Khối chức năng 2132 phân tích các phần tử cú pháp sau đây, và chuyển quyền điều khiển cho khối chức năng 2134: `vert_dim[sub_view_id[i]]`; `hor_dim[sub_view_id[i]]`; và `quantizer[sub_view_id[i]]`. Khối chức năng 2134 phân tích các hệ số bộ lọc cho mỗi

thành phần YUV, và chuyển điều khiển cho khối chức năng 2136.

Khối chức năng 2136 tăng biến i , và trả quyền điều khiển cho khối phân xử 2108.

Khối chức năng 2138 phân tích phần tử cú pháp $\text{pixel_dist_x}[\text{sub_view_id}[i]]$ và phần tử cú pháp $\text{flip_dist_y}[\text{sub_view_id}[i]]$, và chuyển điều khiển cho khối chức năng 2140. Khối chức năng 2140 đặt biến j bằng 0, và chuyển điều khiển cho khối phân xử 2142. Khối phân xử 2142 này xác định xem giá trị hiện thời của biến j có nhỏ hơn giá trị hiện thời của phần tử cú pháp $\text{num_parts}[\text{sub_view_id}[i]]$ hay không. Nếu có, thì điều khiển được chuyển cho khối chức năng 2144. Nếu không, điều khiển được chuyển cho khối chức năng 2136.

Khối chức năng 2144 phân tích phần tử cú pháp $\text{num_pixel_tiling_filter_coeffs_minus1}[\text{sub_view_id}[i]]$, và chuyển quyền điều khiển cho khối chức năng 2146. Khối chức năng 2146 này phân tích các hệ số cho tất cả các bộ lọc xếp lát điểm ảnh, và chuyển điều khiển cho khối chức năng 2136.

Fig.22 là sơ đồ thể hiện các ví dụ xếp lát tại mức điểm ảnh, được biểu thị chung bằng số chỉ dẫn 220. Fig.22 này được miêu tả thêm dưới đây.

Xếp lát cảnh bằng cách sử dụng MPEG-4 AVC hoặc MVC

Một ứng dụng mã hóa video đa cảnh là truyền hình điểm cảnh miễn phí (hoặc FTV). Ứng dụng này yêu cầu người sử dụng có thể di chuyển tự do giữa hai hoặc nhiều cảnh. Để thực hiện việc này, các cảnh "ảo" giữa hai cảnh cần phải được nội suy hoặc tổng hợp. Có một vài phương pháp thực hiện nội suy cảnh. Một trong số các phương pháp này sử dụng chiều sâu của nội suy/tổng hợp cảnh.

Mỗi cảnh có thể có tín hiệu chiều sâu gắn kèm. Do đó, chiều sâu có thể được coi là một dạng tín hiệu video khác. Fig.9 thể hiện một ví dụ về tín hiệu chiều sâu 900. Để cho phép sử dụng các ứng dụng như FTV, tín hiệu chiều sâu được truyền cùng tín hiệu video. Trong khung xếp lát được đề xuất, tín hiệu chiều sâu cũng có thể được thêm vào như một trong số các lát. Fig.10 thể hiện ví dụ về tín hiệu chiều sâu thêm vào như các lát. Tín hiệu/lát chiều sâu được thể hiện bên phải Fig.10.

Ngay khi chiều sâu được mã hóa như một lát của toàn bộ khung, cú pháp mức cao có thể biểu thị lát nào là tín hiệu chiều sâu để người trình bày có thể sử dụng tín hiệu chiều sâu một cách thích hợp.

Trong trường hợp khi chuỗi đầu vào (như thể hiện Fig.1) được mã hóa bằng cách sử dụng bộ mã hoá chuẩn MPEG-4 AVC (hoặc bộ mã hóa tương ứng với chuẩn và/hoặc khuyến nghị mã hoá video khác), cú pháp mức cao được đề xuất có thể có mặt trong, ví dụ, SPS, PPS, phần đầu phiên, và/hoặc thông báo SEI. Một phương án của phương pháp đề xuất được thể hiện trong bảng 1 nơi mà cú pháp có mặt trong thông báo SEI.

Trong trường hợp khi các chuỗi đầu vào của các cảnh giả (như thể hiện trong Fig.1) được mã hóa bằng cách sử dụng phần mở rộng MVC của bộ mã hoá chuẩn MPEG-4 AVC (hoặc bộ mã hóa tương ứng với chuẩn video đa cảnh đối với chuẩn và/hoặc khuyến nghị mã hoá video khác), cú pháp mức cao được đề xuất có thể có mặt trong SPS, PPS, phần đầu phiên, thông báo SEI hoặc hiện trạng định rõ. Một phương án của phương pháp đề xuất được thể hiện trong bảng 1. Bảng 1 này thể hiện các phần tử cú pháp có mặt trong cấu trúc SPS, bao gồm các phần tử cú pháp trong các phương án của sáng chế.

Bảng 1

seq_parameter_set_mvc_extension() {	C	diễn giải
num_views_minus_1		ue(v)
cho(i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
cho(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_10[i]		ue(v)
cho(j = 0; j < num_anchor_refs_10[i]; j++)		
anchor_ref_10[i][j]		ue(v)
num_anchor_refs_11[i]		ue(v)
cho(j = 0; j < num_anchor_refs_11[i]; j++)		
anchor_ref_11[i][j]		ue(v)
}		
cho(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_10[i]		ue(v)
cho(j = 0; j < num_non_anchor_refs_10[i]; j++)		

non_anchor_ref 10[i][j]		ue(v)
num_non_anchor_refs_11[l]		ue(v)
cho(j = 0; j < num_non_anchor_refs_11[i]; j++)		
non_anchor_ref 11[i][j]		ue(v)
}		
pseudo_view_present_flag		u(1)
nếu (pseudo_view_present_flag) {		
tiling_mode		
org_pic_width_in_mbs_minus1		
org_pic_height_in_mbs_minus1		
cho(i =0; i < num_views_minus_1; i++)		
pseudo_view_info(i);		
}		
}		

Bảng 2 thể hiện các phần tử cú pháp cho phần tử cú pháp pseudo_view_info của bảng 1, theo một phương án của sáng chế.

Bảng 2

pseudo_view_info (pseudo_view_id) {	C	diễn giải
num_sub_views_minus_1 [pseudo_view_id]	5	ue(v)
nếu (num_sub_views_minus_1 != 0) {		
cho (i =0; i < num_sub_views_minus_1 [pseudo_view_id]; i++) {		
sub_view_id[i]	5	ue(v)
num_parts_minus1[sub_view_id[i]]	5	ue(v)
cho(j = 0; j <= num_parts_minus1[sub_view_id[i]]; j++) {		
loc_left_offset[sub_view_id[i]][j]	5	ue(v)
loc_top_offset[sub_view_id[i]][j]	5	ue(v)
frame_crop_left_offset[sub_view_id[i]][j]	5	ue(v)
frame_crop_right_offset[sub_view_id[i]][j]	5	ue(v)
frame_crop_top_offset[sub_view_id[i]][j]	5	ue(v)
frame_crop_bottom_offset[sub_view_id[i]][j]	5	ue(v)
}		
nếu (tiling_mode == 0) {		
flip_dir[sub_view_id[i]][j]	5	u(2)
upsample_view_flag[sub_view_id[i]]	5	u(1)
nếu (upsample_view_flag[sub_view_id[i]])		
upsample_filter[sub_view_id[i]]	5	u(2)
if(upsample_fiter[sub_view_id[i]] == 3) {		
vert_dim[sub_view_id[i]]	5	ue(v)

hor_dim[sub_view_id[i]]	5	ue(v)
quantizer[sub_view_id[i]]	5	ue(v)
cho (yuv= 0; yuv< 3; yuv++) {		
cho (y = 0 y < vert_dim[sub_view_id[i]] - 1; y ++) {		
cho (x = 0; x < hor_dim[sub_view_id[i]] - 1; x ++)		
filter_coefs[sub_view_id[i]] [yuv][y][x]	5	se(v)
}		
}		
}		
} // nếu (tiling_mode == 0)		
ngược lại nếu (tiling_mode == 1) {		
pixel_dist_x[sub_view_id[i]]		
pixel_dist_y[sub_view_id[i]]		
cho(j = 0; j <= num_parts[sub_view_id[i]]; j++) {		
num_pixel_tiling_filter_coefs_minus1 [sub_view_id[i]][j]		
cho (coeff_idx = 0; coeff_idx <=		
num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]][j]; j++)		
pixel_tiling_filter_coefs[sub_view_id[i]][j]		
} // cho (j = 0; j <= num_parts[sub_view_id[i]]; j++)		
} // ngược lại nếu (tiling_mode == 1)		
} // cho (i = 0; i < num_sub_views_minus_1; i++)		
} // nếu(num_sub_views_minus_1 != 0)		
}		

Ý nghĩa của các phần tử cú pháp thể hiện trên bảng 1 và bảng 2

pseudo_view_present_flag là đúng biểu thị rằng một số cảnh là siêu cảnh của nhiều cảnh con.

tiling_mode bằng 0 biểu thị rằng các cảnh con được xếp lát tại mức hình. Giá trị 1 biểu thị rằng việc xếp lát được thực hiện ở mức điểm ảnh.

Thông báo SEI mới có thể sử dụng một giá trị cho loại trọng tải SEI mà không được dùng trong chuẩn MPEG-4 AVC hoặc phần mở rộng của chuẩn MPEG-4 AVC. Thông báo SEI mới này bao gồm một số phần tử cú pháp có ý nghĩa như sau.

num_coded_views_minus1 + 1 biểu thị số lượng các cảnh mã hoá được hỗ trợ bởi các luồng bit. Giá trị của num_coded_views_minus1 trong phạm vi từ 0 đến 1023, bao gồm.

$\text{org_pic_width_in_mbs_minus1} + 1$ xác định chiều rộng của hình trong mỗi cảnh theo các đơn vị khối macro.

Biến cho chiều rộng hình theo các đơn vị khối macro được suy ra như sau:

$$\text{PicWidthInMbs} = \text{org_pic_width_in_mbs_minus1} + 1$$

Biến của chiều rộng hình cho thành phần sắc độ được suy ra như sau:

$$\text{PicWidthInSamplesL} = \text{PicWidthInMbs} * 16$$

Biến của chiều rộng hình cho các thành phần sắc thái được suy ra như sau:

$$\text{PicWidthInSamplesC} = \text{PicWidthInMbs} * \text{MbWidthC}$$

$\text{org_pic_height_in_mbs_minus1} + 1$ xác định chiều cao của hình trong mỗi cảnh theo các đơn vị khối macro.

Biến cho chiều cao hình theo các đơn vị khối macro được suy ra như sau:

$$\text{PicHeightInMbs} = \text{org_pic_height_in_mbs_minus1} + 1$$

Biến của chiều cao hình cho thành phần sắc độ được suy ra như sau:

$$\text{PicHeightInSamplesL} = \text{PicHeightInMbs} * 16$$

Biến của chiều cao hình cho các thành phần sắc thái được suy ra như sau:

$$\text{PicHeightInSamplesC} = \text{PicHeightInMbs} * \text{MbHeightC}$$

$\text{num_sub_views_minus1} + 1$ biểu thị số lượng các cảnh phụ được mã hoá trong cảnh hiện thời. Giá trị $\text{num_coded_views_minus1}$ trong phạm vi từ 0 đến 1023, bao gồm.

$\text{sub_view_id}[i]$ xác định sub_view_id của cảnh con có thứ tự giải mã được biểu thị bằng i .

$\text{num_parts}[\text{sub_view_id}[i]]$ xác định số lượng các phần mà hình của $\text{sub_view_id}[i]$ được tách thành.

$\text{loc_left_offset}[\text{sub_view_id}[i]][j]$ và $\text{loc_top_offset}[\text{sub_view_id}[i]][j]$ lần lượt

định rõ các vị trí theo độ dịch điểm ảnh bên trái và bên trên, tại đó phần hiện thời j nằm trong hình cấu trúc lại cuối cùng của cảnh có sub_view_id bằng $\text{sub_view_id}[i]$.

$\text{view_id}[i]$ xác định view_id của cảnh có thứ tự mã hoá biểu thị bởi i .

$\text{frame_crop_left_offset}[\text{view_id}[i]][j]$, $\text{frame_crop_right_offset}[\text{view_id}[i]][j]$, $\text{frame_crop_top_offset}[\text{view_id}[i]][j]$, và $\text{frame_crop_bottom_offset}[\text{view_id}[i]][j]$ định rõ các mẫu của các hình trong chuỗi video được mã hoá mà là một phần của num_part và view_id i , dưới dạng vùng hình chữ nhật được xác định theo các tọa độ khung đầu ra.

Các biến CropUnitX và CropUnitY được suy ra như sau:

- Nếu chroma_format_idc bằng 0, CropUnitX và CropUnitY được suy ra như sau:

$$\text{CropUnitX} = 1$$

$$\text{CropUnitY} = 2 - \text{frame_mbs_only_flag}$$

- ngược lại (chroma_format_idc bằng 1, 2, hoặc 3), CropUnitX và CropUnitY được suy ra như sau:

$$\text{CropUnitX} = \text{SubWidthC}$$

$$\text{CropUnitY} = \text{SubHeightC} * (2 - \text{frame_mbs_only_flag})$$

Khung cắt hình chữ nhật bao gồm các mẫu sắc độ có các tọa độ khung ngang sau:

$\text{CropUnitX} * \text{frame_crop_left_offset}$ đến $\text{PicWidthInSamplesL} - (\text{CropUnitX} * \text{frame_crop_right_offset} + 1)$ và tọa độ khung dọc từ $\text{CropUnitY} * \text{frame_crop_top_offset}$ đến $(16 * \text{FrameHeightInMbs}) - (\text{CropUnitY} * 20 * \text{frame_crop_bottom_offset} + 1)$. Giá trị $\text{frame_crop_left_offset}$ sẽ nằm trong khoảng từ 0 đến $(\text{PicWidthInSamplesL}/\text{CropUnitX}) - (\text{frame_crop_right_offset} + 1)$; và giá trị $\text{frame_crop_top_offset}$ nằm trong khoảng từ 0 đến $(16 * \text{FrameHeightInMbs}/\text{CropUnitY}) - (\text{frame_crop_bottom_offset} + 1)$.

Khi chroma_format_idc không bằng 0, các mẫu quy định tương ứng của hai mảng sắc thái là các mẫu có các tọa độ khung ($x/\text{SubWidthC}$, $y/\text{SubHeightC}$), trong đó (x , y) là

các tọa độ khung của các mẫu sắc độ quy định.

Đối với các trường được giải mã, các mẫu quy định của trường được giải mã là các mẫu giảm trong hình chữ nhật xác định trong các tọa độ khung.

`num_parts[view_id[i]]` xác định số lượng các phần mà hình `view_id[i]` được tách thành.

`depth_flag[view_id[i]]` xác định xem phần hiện thời có phải là tín hiệu chiều sâu hay không. Nếu `depth_flag` bằng 0, thì phần hiện thời không phải là tín hiệu chiều sâu. Nếu `depth_flag` bằng 1, thì phần hiện thời là tín hiệu chiều sâu kết hợp với cảnh được nhận ra bởi `view_id[i]`.

`flip_dir[sub_view_id[i]][j]` xác định hướng lật cho phần hiện thời. `Flip_dir` bằng 0 biểu thị không có hướng lật nào, `flip_dir` bằng 1 biểu thị lật theo chiều ngang, `flip_dir` bằng 2 biểu thị lật theo hướng thẳng dọc, và `flip_dir` bằng 3 biểu thị lật theo các chiều ngang và dọc.

`flip_dir[view_id[i]][j]` xác định hướng lật cho phần hiện hành. `Flip_dir` bằng 0 biểu thị không có hướng lật nào, `flip_dir` bằng 1 biểu thị lật theo chiều ngang, `flip_dir` bằng 2 biểu thị lật theo hướng thẳng dọc, và `flip_dir` bằng 3 biểu thị lật theo các chiều ngang và dọc.

`loc_left_offset[view_id[i]][j]`, `loc_top_offset[view_id[i]][j]` xác định vị trí theo độ dịch vòng điểm ảnh, tại đó phần hiện thời `j` nằm trong hình cấu trúc lại cuối cùng của cảnh có `view_id` bằng `view_id[i]`

`upsample_view_flag[view_id[i]]` biểu thị hình thuộc về một cảnh được xác định bởi `view_id[i]` có được lấy mẫu lên hay không. `upsample_view_flag[view_id[i]]` bằng 0 xác định rằng hình có `view_id` bằng `view_id[i]` sẽ không được lấy mẫu lên. `upsample_view_flag[view_id[i]]` bằng 1 xác định rằng hình có `view_id` bằng `view_id[i]` sẽ được lấy mẫu lên.

`upsample_filter[view_id[i]]` biểu thị loại bộ lọc mà được sử dụng để lấy mẫu lên.

upsample_filter[view_id[i]] bằng 0 biểu thị rằng bộ lọc 6-tap AVC sẽ được dùng, upsample_filter[view_id[i]] bằng 1 biểu thị rằng bộ lọc 4-tap SVC sẽ được dùng, upsample_filter[view_id[i]] bằng 2 biểu thị rằng bộ lọc song tuyến tính sẽ được dùng, upsample_filter[view_id[i]] bằng đến 3 biểu thị rằng các hệ số bộ lọc tùy ý được truyền. Khi upsample_fiter[view_id[i]] không có mặt nó được đặt bằng 0. Trong phương án này, chúng ta sử dụng bộ lọc tùy ý 2D. Bộ lọc này có thể dễ dàng mở rộng thành bộ lọc 1D, và một số bộ lọc phi tuyến khác.

vert_dim[view_id[i]] xác định chiều dọc của bộ lọc 2D tùy ý.

hor_dim[view_id[i]] xác định chiều dọc của bộ lọc 2D tùy ý.

quantizer[view_id[i]] xác định hệ số lượng tử hoá mỗi hệ số bộ lọc.

filter_coeffs[view_id[i]][yuv][y][x] xác định các hệ số bộ lọc được lượng tử hoá. yuv biểu thị thành phần mà các hệ số bộ lọc áp dụng vào. yuv bằng 0 xác định thành phần Y, yuv bằng 1 xác định thành phần U, và yuv bằng 2 xác định thành phần V.

pixel_dist_x[sub_view_id[i]] và pixel_dist_y[sub_view_id[i]] lần lượt xác định khoảng cách theo chiều ngang và chiều dọc trong cảnh giả cấu trúc lại cuối cùng giữa các điểm ảnh lân cận trong cảnh có sub_view_id bằng sub_view_id[i].

num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]][j] + 1 biểu thị số lượng các hệ số bộ lọc khi chế độ xếp lát được đặt bằng 1.

pixel_tiling_filter_coeffs[sub_view_id[i]][j] biểu thị các hệ số bộ lọc được cần đến để đại diện cho bộ lọc mà được dùng để lọc hình được xếp lát.

Các ví dụ xếp lát tại mức điểm ảnh

Fig.22 là hai ví dụ thể hiện việc xếp cảnh giả bằng cách xếp lát các điểm ảnh từ 4 cảnh được biểu thị chung bằng các số chỉ dẫn 2210 và 2220. Bốn cảnh được biểu thị chung bằng số chỉ dẫn 2250. Các giá trị cú pháp cho ví dụ thứ nhất trong Fig.22 được thể hiện trên bảng 3 dưới đây.

Bảng 3

pseudo_view_info (pseudo_view_id) {	Giá trị
num_sub_views_minus_1 [pseudo_view_id]	3
sub_view_id[0]	0
num_parts_minus1[0]	0
loc_left_offset[0][0]	0
loc_top_offset[0][0]	0
pixel_dist_x[0][0]	0
pixel_dist_y[0][0]	0
sub_view_id[1]	0
num_parts_minus1 [1]	0
loc_left_offset[1][0]	1
loc_top_offset[1][0]	0
pixel_dist_x[1][0]	0
pixel_dist_y[1][0]	0
sub_view_id[2]	0
num_parts_minus1[2]	0
loc_left_offset[2][0]	0
loc_top_offset[2][0]	1
pixel_dist_x[2][0]	0
pixel_dist_y[2][0]	0
sub_view_id[3]	0
num_parts_minus1[3]	0
loc_left_offset[3][0]	1
loc_top_offset[3][0]	1
pixel_dist_x[3][0]	0
pixel_dist_y[3][0]	0

Các giá trị cú pháp cho ví dụ thứ hai trong Fig.22 đều giống nhau trừ hai phần tử cú pháp sau đây: loc_left_offset[3][0] bằng 5 và loc_top_offset[3][0] bằng 3.

Độ dịch vòng biểu thị các điểm ảnh tương ứng với cảnh sẽ bắt đầu tại một vị trí dịch nhất định. Điều này được thể hiện trong Fig.22 (2220). Việc này có thể được thực hiện, ví dụ, khi hai cảnh tạo ra các ảnh trong đó các đối tượng chung xuất hiện dịch từ cảnh này sang cảnh khác. Ví dụ, nếu máy quay (camera) thứ nhất và thứ hai (trình diễn các cảnh thứ nhất và thứ hai) chụp hình đối tượng, thì đối tượng xuất hiện sẽ được dịch 5 điểm ảnh sang bên phải trong cảnh thứ hai so với cảnh thứ nhất. Điều này có nghĩa là điểm ảnh (i-5, j) trong cảnh thứ nhất tương ứng với điểm ảnh (i, j) trong cảnh thứ hai.

Nếu các điểm ảnh của hai cảnh chỉ đơn giản là điểm ảnh trên điểm ảnh được xếp lát, thì các điểm ảnh này có thể không có nhiều mối tương quan giữa các điểm ảnh lân cận trong lát, và độ lợi mã hóa không gian có thể rất nhỏ. Ngược lại, khi dịch một lát sao cho điểm ảnh $(i-5, j)$ từ một cảnh được đặt đến điểm ảnh (i, j) tiếp theo từ hai cảnh, mối tương quan không gian có thể được tăng thêm và độ lợi mã hóa không gian cũng có thể rất được tăng thêm. Bởi vì, ví dụ, các điểm ảnh tương ứng cho đối tượng trong các cảnh thứ nhất và thứ hai đang được xếp lát gần với nhau.

Do đó, sự có mặt của `loc_left_offset` và `loc_top_offset` có thể có lợi cho việc mã hóa. Thông tin độ dịch có thể thu được bằng phương tiện ngoài. Ví dụ, thông tin vị trí của các camera hoặc các vectơ chênh lệch toàn cầu giữa các cảnh có thể được dùng để xác định thông tin độ dịch này.

Để xê dịch, một số điểm ảnh trong cảnh giả không được gán các giá trị điểm ảnh từ một cảnh bất kỳ. Tiếp tục ví dụ ở trên, khi xếp điểm ảnh $(i-5, j)$ từ một cảnh sát cạnh điểm ảnh (i, j) từ hai cảnh, các giá trị $i = 0 \dots 4$ không có điểm ảnh $(i-5, j)$ từ một cảnh để xếp lát, do đó, những điểm ảnh này rỗng trong lát. Đối với những điểm ảnh trong cảnh giả (lát) mà không được gán các giá trị điểm ảnh từ một cảnh bất kỳ, ít nhất một cài đặt sử dụng một thủ tục nội suy tương tự như thủ tục nội suy điểm ảnh phụ theo phần bù chuyển động trong AVC. Tức là, các điểm ảnh lát rỗng có thể được nội suy từ các điểm ảnh lân cận. Việc nội suy này có thể dẫn đến mối tương quan không gian lớn hơn trong lát và độ lợi mã hóa lớn hơn cho lát.

Trong mã hóa video, chúng ta có thể chọn loại mã hóa khác cho mỗi hình, như các hình I, P, và B. Ngoài ra, đối với mã hoá video đa cảnh, chúng ta định nghĩa các hình neo và không neo. Theo một phương án, sáng chế đề xuất rằng việc quyết định nhóm có thể được thực hiện dựa trên kiểu hình. Thông tin nhóm này được thể hiện trong cú pháp mức cao.

Như được thể hiện trên Fig.11, ví dụ về 5 cảnh được xếp lát trên một khung được biểu thị chung bằng số chỉ dẫn 1100. Cụ thể, chuỗi khán phòng được thể hiện với 5 cảnh

xếp trên một khung. Ngoài ra, có thể thấy rằng cảnh thứ năm được chia thành hai phần để nó có thể được xếp lát trên một khung hình chữ nhật. Ở đây, mỗi cảnh có kích thước QVGA nên toàn bộ chiều của khung là 640x600. Vì 600 không phải là bội 16 nên nó sẽ được mở rộng đến 608.

Đối với ví dụ này, thông báo SEI có thể được thể hiện trong bảng 4.

Bảng 4

<code>multiview_display_info(payloadSize) {</code>	Giá trị
<code>num_coded_views_minus1</code>	5
<code>org_pic_width_in_mbs_minus1</code>	40
<code>org_pic_height_in_mbs_minus1</code>	30
<code>view_id[0]</code>	0
<code>num_parts[view_id[0]]</code>	1
<code>depth_flag[view_id[0]][0]</code>	0
<code>flip_dir[view_id[0]][0]</code>	0
<code>loc_left_offset[view_id[0]][0]</code>	0
<code>loc_top_offset[view_id[0]][0]</code>	0
<code>frame_crop_left_offset[view_id[0]][0]</code>	0
<code>frame_crop_right_offset[view_id[0]][0]</code>	320
<code>frame_crop_top_offset[view_id[0]][0]</code>	0
<code>frame_crop_bottom_offset[view_id[0]][0]</code>	240
<code>upsample_view_flag[view_id[0]]</code>	1
<code>nếu(upsample_view_flag[view_id[0]]) {</code>	
<code>vert_dim[view_id[0]]</code>	6
<code>hor_dim[view_id[0]]</code>	6
<code>quantizer[view_id[0]]</code>	32
<code>cho (yuv= 0; yuv< 3; yuv++) {</code>	
<code>cho (y = 0; y < vert_dim[view_id[i]] - 1; y ++) {</code>	
<code>cho (x = 0; x < hor_dim[view_id[i]] - 1; x ++)</code>	
<code>filter_coefs[view_id[i]] [yuv][y][x]</code>	XX
<code>view_id[1]</code>	1
<code>num_parts[view_id[1]]</code>	1
<code>depth_flag[view_id[0]][0]</code>	0

flip_dir[view_id[1]][0]	0
loc_left_offset[view_id[1]][0]	0
loc_top_offset[view_id[1]][0]	0
frame_crop_left_offset[view_id[1]][0]	320
frame_crop_right_offset[view_id[1]][0]	640
frame_crop_top_offset[view_id[1]][0]	0
frame_crop_bottom_offset[view_id[1]][0]	320
upsample_view_flag[view_id[1]]	1
if(upsample_view_flag[view_id[1]]) {	
vert_dim[view_id[1]]	6
hor_dim[view_id[1]]	6
quantizer[view_id[1]]	32
cho (yuv= 0; yuv< 3; yuv++) {	
cho (y = 0; y < vert_dim[view_id[i]] - 1; y ++)	
cho (x = 0; x < hor_dim[view_id[i]] - 1; x ++)	
filter_coefs[view_id[i]] [yuv][y][x]	XX
..... (tương tự như cảnh 2,3)	
view_id[4]	4
num_parts[view_id[4]]	2
depth_flag[view_id[0]][0]	0
flip_dir[view_id[4]][0]	0
loc_left_offset[view_id[4]][0]	0
loc_top_offset[view_id[4]][0]	0
frame_crop_left_offset[view_id[4]][0]	0
frame_crop_right_offset[view_id[4]][0]	320
frame_crop_top_offset[view_id[4]][0]	480
frame_crop_bottom_offset[view_id[4]][0]	600
flip_dir[view_id[4]][1]	0
loc_left_offset[view_id[4]][1]	0
loc_top_offset[view_id[4]][1]	120
frame_crop_left_offset[view_id[4]][1]	320
frame_crop_right_offset[view_id[4]][1]	640
frame_crop_top_offset[view_id[4]][1]	480
frame_crop_bottom_offset[view_id[4]][1]	600

upsample_view_flag[view_id[4]]	1
nếu(upsample_view_flag[view_id[4]]) {	
vert_dim[view_id[4]]	6
hor_dim[view_id[4]]	6
quantizer[view_id[4]]	32
cho (yuv= 0; yuv< 3; yuv++) {	
cho (y = 0; y < vert_dim[view_id[i]] - 1; y ++) {	
cho (x = 0; x < hor_dim[view_id[i]] - 1; x ++)	
filter_coefs[view_id[i]] [yuv][y][x]	XX

Bảng 5 thể hiện cấu trúc cú pháp chung để truyền thông tin đa cảnh cho ví dụ thể hiện trong bảng 4.

Bảng 5

multiview_display_info(payloadSize) {	C	diễn giải
num_coded_views_minus1	5	ue(v)
org_pic_width_in_mbs_minus1	5	ue(v)
org_pic_height_in_mbs_minus1	5	ue(v)
cho(i = 0; i <= num_coded_views_minus1; i++) {		
view_id[i]	5	ue(v)
num_parts[view_id[i]]	5	ue(v)
cho(j = 0; j <= num_parts[i]; j++) {		
depth_flag[view_id[i]][j]		
flip_dir[view_id[i]][j]	5	u(2)
loc_left_offset[view_id[i]][j]	5	ue(v)
loc_top_offset[view_id[i]][j]	5	ue(v)
frame_crop_left_offset[view_id[i]][j]	5	ue(v)
frame_crop_right_offset[view_id[i]][j]	5	ue(v)
frame_crop_top_offset[view_id[i]][j]	5	ue(v)
frame_crop_bottom_offset[view_id[i]][j]	5	ue(v)
}		
upsample_view_flag[view_id[i]]	5	u(1)
nếu(upsample_view_flag[view_id[i]])		
upsample_filter[view_id[i]]	5	ue(2)
nếu(upsample_fiter[view_id[i]] == 3) {		
vert_dim[view_id[i]]	5	ue(v)
hor_dim[view_id[i]]	5	ue(v)
quantizer[view_id[i]]	5	ue(v)

cho (yuv= 0; yuv< 3; yuv++) {		
cho (y = 0; y < vert_dim[view_id[i]] - 1; y ++) {		
cho (x = 0; x < hor_dim[view_id[i]] - 1; x ++)		
filter_coeffs[view_id[i]] [yuv][y][x]	5	se(v)
}		
}		
}		
}		

Fig.23 là sơ đồ thể hiện thiết bị xử lý video 2300. Thiết bị xử lý video 2300 này có thể là, ví dụ, hộp đặt trên nóc hoặc thiết bị khác nhận tín hiệu video được mã hoá và tạo ra, ví dụ, tín hiệu video được giải mã để hiển thị cho người dùng hoặc để lưu trữ. Do đó, thiết bị 2300 này có thể cung cấp đầu ra của nó cho TV, màn hình máy tính, hoặc máy tính hay thiết bị xử lý khác.

Thiết bị 2300 bao gồm bộ giải mã 2310 nhận tín hiệu dữ liệu 2320. Tín hiệu dữ liệu 2320 này có thể bao gồm, ví dụ, AVC hay luồng MVC tương thích. Bộ giải mã 2310 giải mã tất cả hoặc một phần tín hiệu nhận được 2320 và cung cấp đầu ra là tín hiệu video được giải mã 2330 và thông tin xếp lát 2340. Video được giải mã 2330 và thông tin xếp lát 2340 này được cung cấp cho bộ lựa chọn 2350. Thiết bị 2300 còn bao gồm giao tiếp người dùng 2360 mà nhận đầu vào của người dùng 2370. Giao tiếp người dùng 2360 này cung cấp tín hiệu chọn hình 2380, dựa vào đầu vào của người dùng 2370, cho bộ chọn 2350. Tín hiệu chọn hình 2380 và đầu vào của người dùng 2370 biểu thị hình nào trong số các hình của người dùng mong muốn được hiển thị. Bộ chọn 2350 tạo ra (các) hình được chọn như đầu ra 2390. Bộ chọn 2350 này sử dụng thông tin chọn hình 2380 để chọn các hình trong video được giải mã 2330 để tạo ra như đầu ra 2390. Bộ chọn 2350 sử dụng thông tin xếp lát 2340 để xác định (các) vị trí hình được chọn trong video được giải mã 2330.

Theo các phương án khác, bộ chọn 2350 bao gồm giao tiếp người dùng 2360, và trong các cài đặt khác, không có giao tiếp người dùng 2360 nào được yêu cầu bởi bộ

chọn 2350 nhận trực tiếp đầu vào của người dùng 2370 mà không cần chức năng giao tiếp riêng nào được thực hiện. Bộ chọn 2350 có thể được thực hiện bằng phần mềm hoặc mạch tích hợp, ví dụ. Bộ chọn 2350 cũng có thể tích hợp bộ giải mã 2310.

Tổng quát hơn, các bộ giải mã theo các phương án được mô tả trong bản mô tả này có thể tạo ra đầu vào được giải mã chứa toàn bộ lát. Thêm vào hoặc cách khác, các bộ giải mã có thể tạo ra các đầu vào được giải mã chỉ chứa một hoặc các hình được chọn (ví dụ, các ảnh hoặc tín hiệu chiều sâu) từ một lát.

Như đề cập ở trên, cú pháp mức cao có thể được dùng để thực hiện việc báo hiệu theo một hoặc nhiều phương án của sáng chế. Cú pháp mức cao có thể được dùng, ví dụ, nhưng không hạn chế ở, dấu hiệu bất kỳ sau đây: số lượng các cảnh mã hoá có mặt trong khung lớn hơn; chiều rộng và chiều cao ban đầu của tất cả các cảnh; đối với mỗi cảnh được mã hoá, bộ định danh cảnh tương ứng với một cảnh; đối với mỗi cảnh được mã hoá, số phần khung của cảnh được tách ra; đối với mỗi phần của cảnh, chiều lật (mà có thể là, ví dụ, không lật, chỉ lật theo chiều ngang, chỉ lật theo chiều dọc hoặc lật theo chiều ngang và dọc); đối với mỗi phần của cảnh, vị trí bên trái trong các điểm ảnh hoặc các khối macro tại đó phần hiện thời thuộc về khung cuối cùng cho một cảnh; đối với mỗi phần của cảnh, vị trí bên trên của phần này trong các điểm ảnh hoặc các khối macro tại đó phần hiện thời thuộc về khung cuối cùng cho một cảnh; đối với mỗi phần của cảnh, vị trí bên trái, trong khung giải mã/mã hoá lớn hiện thời, của cửa sổ cắt trong các điểm ảnh hoặc các khối macro; đối với mỗi phần của cảnh, vị trí bên phải, trong khung giải mã/mã hoá lớn hiện thời, của cửa sổ cắt trong các điểm ảnh hoặc các khối macro; và, đối với mỗi phần của cảnh, vị trí bên dưới, trong khung giải mã/mã hoá lớn hiện thời, của cửa sổ cắt trong các điểm ảnh hoặc các khối macro; đối với mỗi cảnh được mã hoá, cảnh cần phải được lấy mẫu lên trước khi xuất ra (trong đó nếu việc lấy mẫu lên cần phải được thực hiện, thì cú pháp mức cao có thể được dùng để biểu thị phương pháp lấy mẫu lên (bao gồm, nhưng không hạn chế ở, bộ lọc AVC 6-tap, bộ lọc SVC 4-tap, bộ lọc song tuyến tính hoặc bộ lọc tuyến tính 1D, 2D hoặc bộ lọc phi tuyến tính).

Cần lưu ý rằng các thuật ngữ “bộ mã hoá” và “bộ giải mã” bao gồm các cấu trúc

chung và không bị hạn chế bởi bất kỳ một chức năng hoặc dấu hiệu nào. Ví dụ, bộ giải mã có thể nhận sóng mang điều chế mà chứa luồng bit được mã hoá, và giải điều chế luồng bit được mã hoá này, cũng như giải mã luồng bit.

Các phương pháp đã được mô tả ở đây. Tuy nhiên, cần lưu ý rằng các biến đổi được tính đến có thể thay đổi một hoặc nhiều dấu hiệu cụ thể được mô tả cho các phương pháp này. Hơn nữa, rất nhiều dấu hiệu mà được mô tả ở đây đều đã biết trong lĩnh vực và, do đó, không cần được mô tả chi tiết thêm nữa.

Hơn nữa, tham chiếu đã được thực hiện với việc dùng các cú pháp mức cao để gửi thông tin nhất định trong một số cài đặt. Tuy nhiên, cần phải hiểu rằng, các tiến trình thực thi khác sử dụng cú pháp mức thấp hơn, hoặc các cơ chế khác hoàn toàn (chẳng hạn, thông tin gửi là một phần dữ liệu được mã hóa) để cung cấp cùng một loại thông tin (hoặc các biến đổi của thông tin đó).

Các tiến trình thực thi khác tạo ra việc xếp lát và báo hiệu thích hợp cho phép nhiều cảnh (các hình, tổng quát hơn) sẽ được xếp lát thành một hình, mã hóa như một hình, và gửi như một hình. Thông tin báo hiệu có thể cho phép bộ xử lý trước tách rời các cảnh/hình. Hơn nữa, các hình được xếp lát có thể là một cảnh, nhưng ít nhất một trong số các cảnh có thể là thông tin chiều sâu. Những tiến trình này có thể tạo ra một hoặc nhiều ưu điểm. Ví dụ, người dùng có thể muốn xem nhiều cảnh theo cách xếp lát, và những tiến trình này tạo ra cách có lợi để mã hóa và truyền hoặc lưu trữ các cảnh này bằng cách xếp lát chúng trước khi mã hoá và truyền/lưu trữ chúng theo cách xếp lát.

Những tiến trình thực thi mà xếp lát nhiều cảnh trong ngữ cảnh AVC và/hoặc MVC cũng tạo ra các ưu điểm bổ sung. AVC có vẻ như chỉ được dùng cho một cảnh, nên không có một cảnh thêm vào nào được kỳ vọng. Tuy nhiên, tiến trình thực thi dựa vào AVC này có thể tạo ra nhiều cảnh trong môi trường AVC vì các cảnh được xếp lát có thể được sắp xếp để, ví dụ, bộ giải mã biết rằng các hình được xếp lát thuộc về các cảnh khác nhau (ví dụ, hình trên bên trái trong cảnh giả là cảnh 1, hình trên bên phải là cảnh 2, vv...).

Hơn nữa, MVC đã bao gồm nhiều cảnh, nên nhiều cảnh không được kỳ vọng sẽ được đưa vào một cảnh giả. Hơn nữa, MVC bị hạn chế trên số lượng các cảnh mà có thể được hỗ trợ, và các tiến trình thực thi dựa vào MVC này tăng thêm các cảnh mà có thể được hỗ trợ để cho phép (như trong tiến trình thực thi dựa vào AVC) các cảnh thêm vào sẽ được xếp lát. Ví dụ, mỗi cảnh giả có thể tương ứng với một trong số các cảnh MVC được hỗ trợ, và bộ giải mã có thể biết rằng mỗi “cảnh được hỗ trợ” thường chứa bốn cảnh theo thứ tự xếp lát được sắp xếp trước. Do đó, trong tiến trình thực thi này, số lượng các cảnh có thể gấp 4 lần lượt số lượng “các cảnh được hỗ trợ”.

Tiến trình thực thi mô tả ở đây có thể được thực hiện trong, ví dụ, phương pháp hay quy trình, thiết bị, hoặc chương trình phần mềm. Dù chỉ được mô tả dưới dạng một tiến trình thực hiện (ví dụ, chỉ được mô tả như phương pháp), việc thực hiện các dấu hiệu mô tả ở đây cũng có thể được thực hiện trong các dạng khác (ví dụ, thiết bị hoặc chương trình). Thiết bị có thể được thực hiện trong, ví dụ, phần cứng, phần mềm, và phần sụn thích hợp. Các phương pháp có thể được thực hiện trong, ví dụ, thiết bị như, ví dụ, bộ xử lý, mà được dùng để chỉ các thiết bị xử lý nói chung, bao gồm, ví dụ, máy tính, bộ vi xử lý, mạch tích hợp, hoặc thiết bị logic lập trình được. Các thiết bị xử lý còn bao gồm các thiết bị truyền thông như, ví dụ, máy tính, điện thoại di động, thiết bị trợ giúp số cá nhân (“PDA”)/thiết bị di động, và các thiết bị khác hỗ trợ truyền thông tin giữa những người dùng cuối.

Việc thực hiện các quy trình và dấu hiệu mô tả ở đây có thể được thể hiện trong các thiết bị hoặc các ứng dụng khác nhau, cụ thể là, ví dụ, thiết bị hoặc ứng dụng gắn với kỹ thuật mã hóa và giải mã dữ liệu. Các ví dụ về thiết bị bao gồm bộ mã hoá video, bộ giải mã video, bộ mã hoá- giải mã video, máy chủ web, hộp đặt trên nóc, máy tính xách tay, máy tính cá nhân, điện thoại di động, PDA và các thiết bị truyền thông khác. Rõ ràng là, thiết bị có thể di động và thậm chí được lắp đặt trong xe di động.

Hơn nữa, phương pháp có thể được thực hiện bằng các lệnh mà được thực thi bởi bộ xử lý, và các lệnh này có thể được lưu trữ trên phương tiện đọc được bằng bộ xử lý như, ví dụ, mạch tích hợp, vật mang phần mềm hoặc thiết bị lưu trữ khác như, ví dụ, đĩa

cứng, đĩa compact, bộ nhớ truy nhập ngẫu nhiên (RAM - random access memory), hoặc bộ nhớ chỉ đọc (ROM - read-only memory). Các lệnh có thể tạo ra chương trình máy tính hoạt động trên phương tiện đọc được bằng bộ xử lý. Rõ ràng là, bộ xử lý có thể bao gồm phương tiện đọc được bằng bộ xử lý có, ví dụ, các lệnh để thực hiện tiến trình xử lý. Các chương trình ứng dụng có thể được tải lên, và được thực thi bởi, máy bao gồm cấu trúc thích hợp bất kỳ. Tốt hơn là, máy được thực hiện trên nền máy tính có phần cứng như một hoặc nhiều đơn vị xử lý trung tâm (CPU - central processing unit), RAM, và các giao tiếp vào ra (I/O - input/output). Nền máy tính cũng có thể bao gồm hệ điều hành và mã vi lệnh. Các tiến trình và các chức năng khác được mô tả ở đây có thể là một phần của mã vi lệnh hoặc một phần của chương trình ứng dụng, hoặc kết hợp bất kỳ của chúng, mà có thể được thực thi bằng CPU. Hơn nữa, các thiết bị ngoại vi khác có thể được nối với nền máy tính như thiết bị lưu trữ dữ liệu khác và thiết bị in.

Hiển nhiên là, việc thực hiện cũng có thể tạo ra tín hiệu được định dạng để mang thông tin mà có thể được, ví dụ, lưu trữ hoặc truyền. Thông tin có thể bao gồm, các lệnh để thực hiện phương pháp, hoặc dữ liệu tạo ra bởi một trong số các quy trình thực hiện mô tả ở đây. Tín hiệu này có thể được định dạng, ví dụ, như sóng điện từ (chẳng hạn, sử dụng phần phổ tần số vô tuyến) hoặc như tín hiệu băng gốc. Việc định dạng có thể bao gồm, ví dụ, mã hoá luồng dữ liệu, tạo ra cú pháp, và điều chế sóng mang với luồng dữ liệu được mã hoá và cú pháp. Thông tin mà tín hiệu mang có thể là, ví dụ, thông tin số hoặc tương tự. Tín hiệu có thể được truyền trên các đường liên kết có dây hoặc không dây khác nhau, như đã biết.

Cần phải hiểu rằng, do một số thành phần và phương pháp hệ thống cấu thành được mô tả cùng các hình vẽ kèm theo được thực thi tốt nhất trong phần mềm, nên các kết nối thật giữa các thành phần hệ thống với các khối chức năng xử lý có thể khác nhau tùy thuộc vào cách mà sáng chế được lập chương trình. Các kiến thức được đưa ra ở đây, một chuyên gia trong lĩnh vực sẽ có khả năng tính toán và thực hiện hoặc định hình tương tự theo sáng chế.

Phần mô tả trên bộc lộ các phương án của sáng chế để chuyên gia trong lĩnh vực có thể hiểu và sử dụng được sử dụng. Các thay đổi đối với các phương án này sẽ trở nên hiển nhiên đối với chuyên gia trong lĩnh vực, và các nguyên lý căn bản được mô tả ở đây có thể được áp dụng cho các phương án khác mà không tách khỏi phạm vi bảo hộ của sáng chế. Như vậy sáng chế không bị hạn chế ở các phương án được mô tả ở đây mà theo phạm vi rộng nhất tuân theo các nguyên lý của sáng chế.

YÊU CẦU BẢO HỘ

1. Phương pháp để giải mã các hình video được mã hóa, phương pháp bao gồm các bước:
 - nhận, trong bộ xử lý, luồng bit video được mã hóa bao gồm các hình video được mã hóa;
 - xác định từ luồng bit video được mã hóa, hình đơn trong khung hình chữ nhật, trong đó hình đơn chứa hình thứ nhất và hình thứ hai được xếp lát lại với nhau;
 - truy cập thông tin để xác định cách mà hình thứ nhất và hình thứ hai là được xếp lát lại với nhau sử dụng hai hoặc hơn hai lát, trong đó thông tin được truy cập chỉ thị rằng hình thứ hai được tách thành hai hoặc hơn hai phần sao cho nó có thể được sắp xếp trong khung hình chữ nhật cùng với hình thứ nhất; và
 - giải mã hình đơn thành hình thứ nhất và hình thứ hai.
2. Phương pháp theo điểm 1 trong đó ít nhất một thành phần trong số hình thứ nhất hoặc một phần của hình thứ hai được lật một cách riêng biệt theo hướng nằm ngang hoặc thẳng đứng.
3. Phương pháp theo điểm 1 trong đó hình đơn còn bao gồm ảnh độ sâu được thêm vào như là lát, trong đó ảnh độ sâu được kết hợp với hình thứ nhất hoặc hình thứ hai.
4. Phương pháp theo điểm 1 trong đó hình thứ nhất và hình thứ hai tương ứng với cảnh bên trái và cảnh bên phải của ảnh lập thể.
5. Phương pháp theo điểm 1 còn bao gồm bước: truy cập thông tin lấy mẫu lên liên quan tới hình thứ nhất hoặc hình thứ hai.
6. Phương pháp theo điểm 1 còn bao gồm bước: truy cập thông tin độ dịch của vị trí của các lát trong hình đơn so với gốc của nó.
7. Phương pháp theo điểm 6 trong đó thông tin độ dịch bao gồm một hoặc nhiều thành phần trong số độ dịch bên trái, độ dịch bên phải, độ dịch bên trên, hoặc độ dịch phía dưới.

8. Phương pháp để ghi mã các hình video thành luồng bit được mã hóa, phương pháp bao gồm:

nhận các hình video;

xếp lát lại với nhau nhiều hình thành hình video đơn trong khung hình chữ nhật, nhiều hình chứa hình thứ nhất và hình thứ hai;

sinh ra thông tin trong phân đoạn báo hiệu chỉ thị cách mà nhiều hình được kết hợp thành các lát, trong đó thông tin được sinh ra chỉ thị rằng hình thứ hai được tách thành hai hoặc hơn hai phần sao cho nó có thể được sắp xếp trong khung hình chữ nhật cùng với hình thứ nhất;

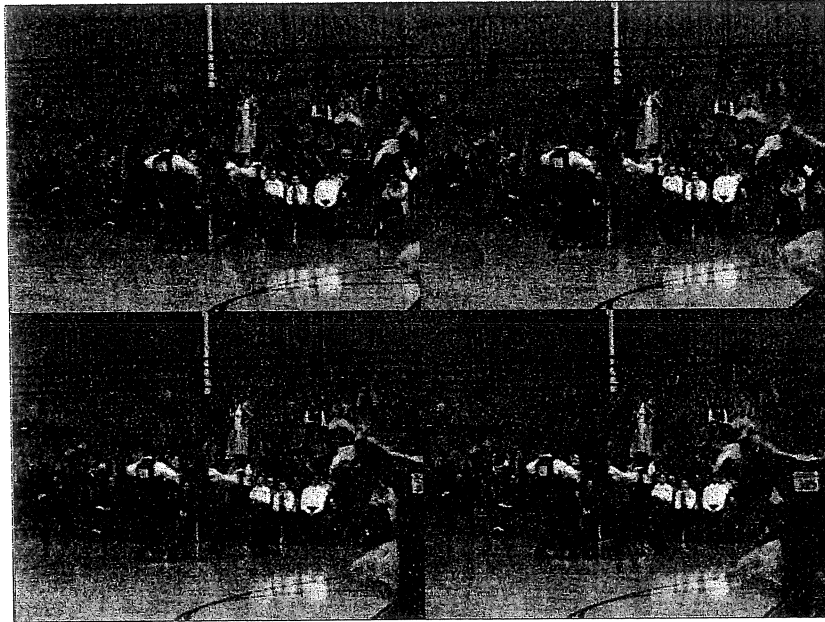
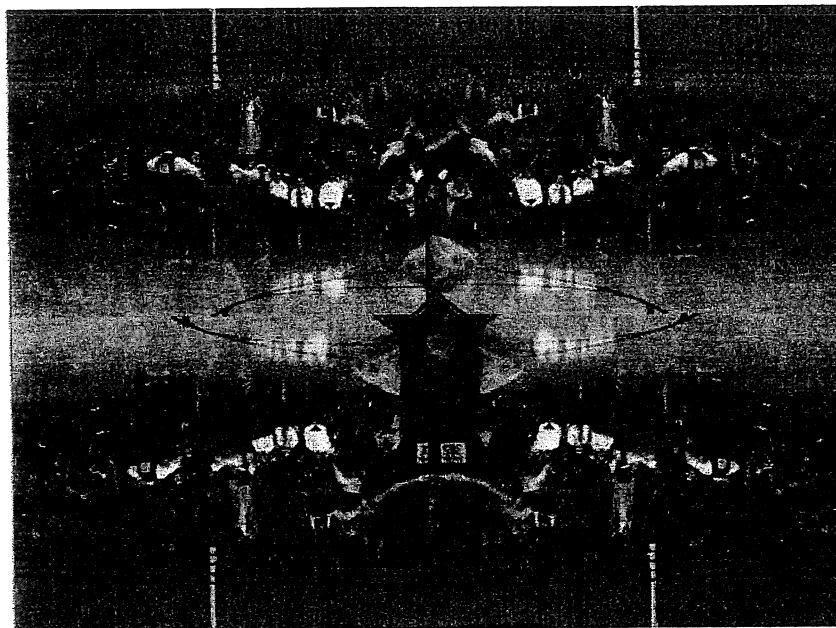
ghi mã hình video đơn để tạo ra hình được mã hóa; và

kết hợp phân đoạn báo hiệu và hình được mã hóa làm đầu ra.

9. Phương pháp theo điểm 8 trong đó thông tin trong phân đoạn báo hiệu chỉ thị rằng ít nhất một thành phần trong số hình thứ nhất hoặc một phần của hình thứ hai được lật một cách riêng biệt theo hướng nằm ngang hoặc thẳng đứng.

10. Phương pháp theo điểm 8 trong đó hình video đơn còn bao gồm ảnh độ sâu được thêm vào như là lát, trong đó ảnh độ sâu được kết hợp với hình thứ nhất hoặc hình thứ hai.

1/28

100**FIG. 1**200**FIG. 2**

2/28

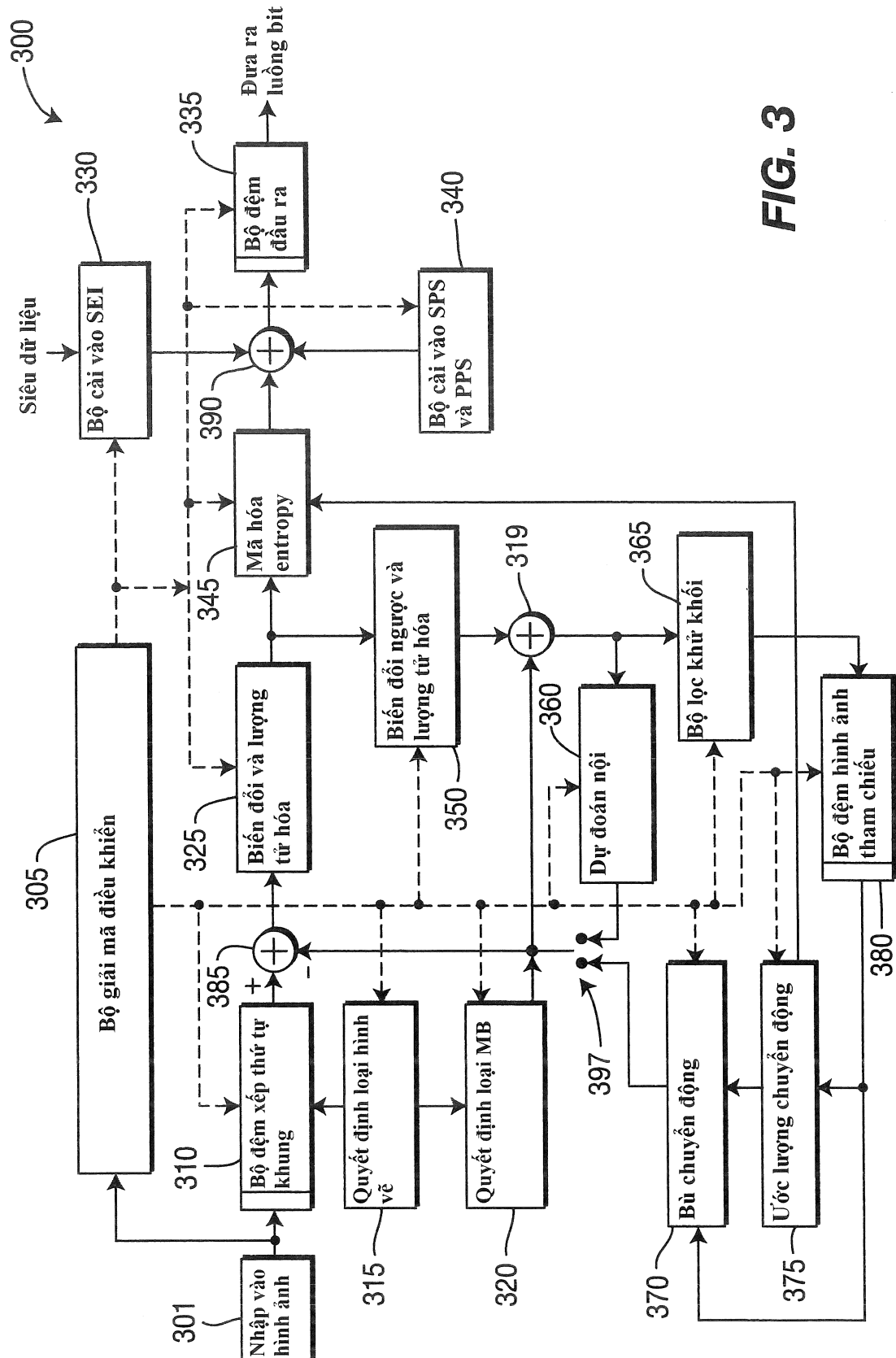


FIG. 3

3/28

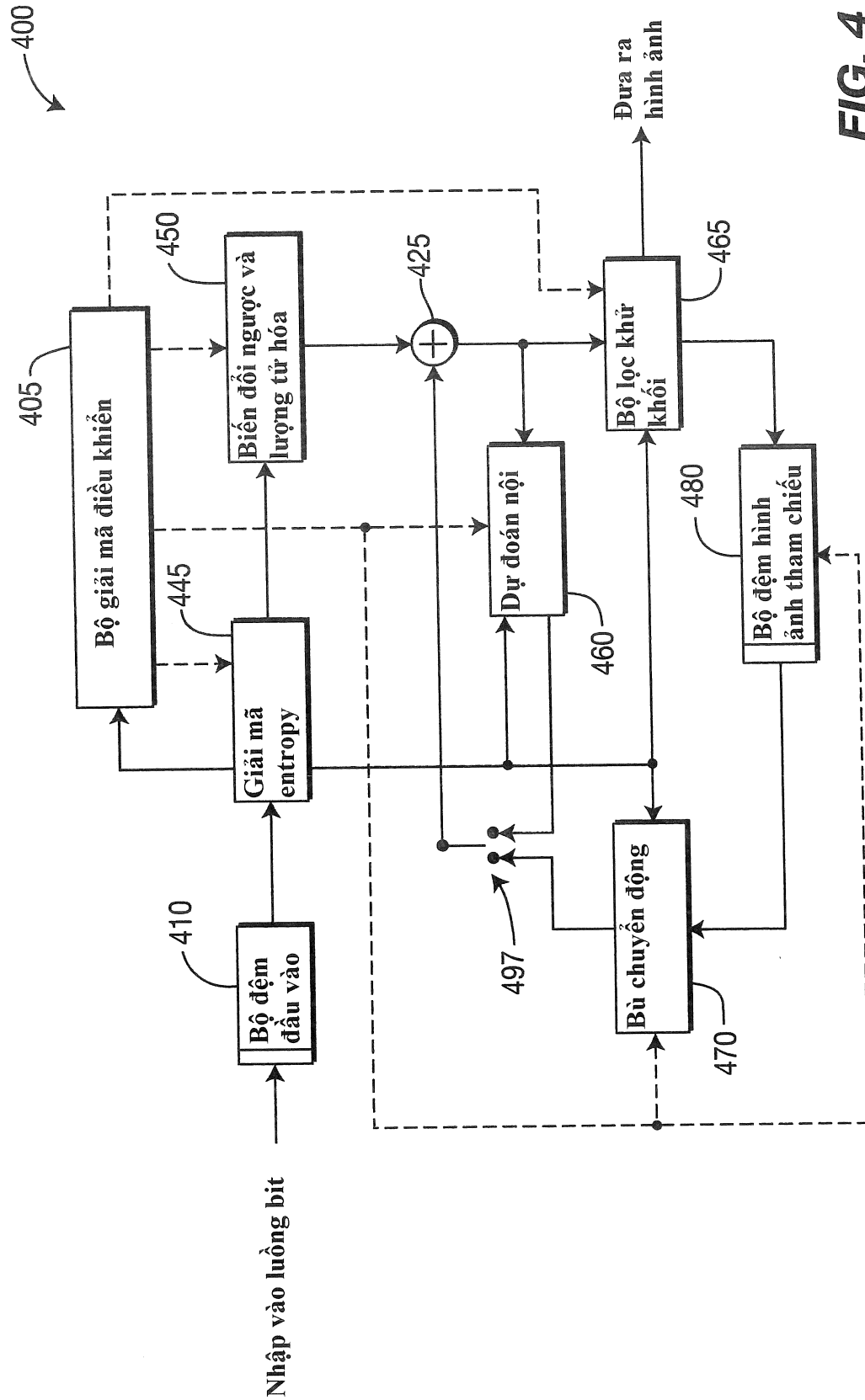
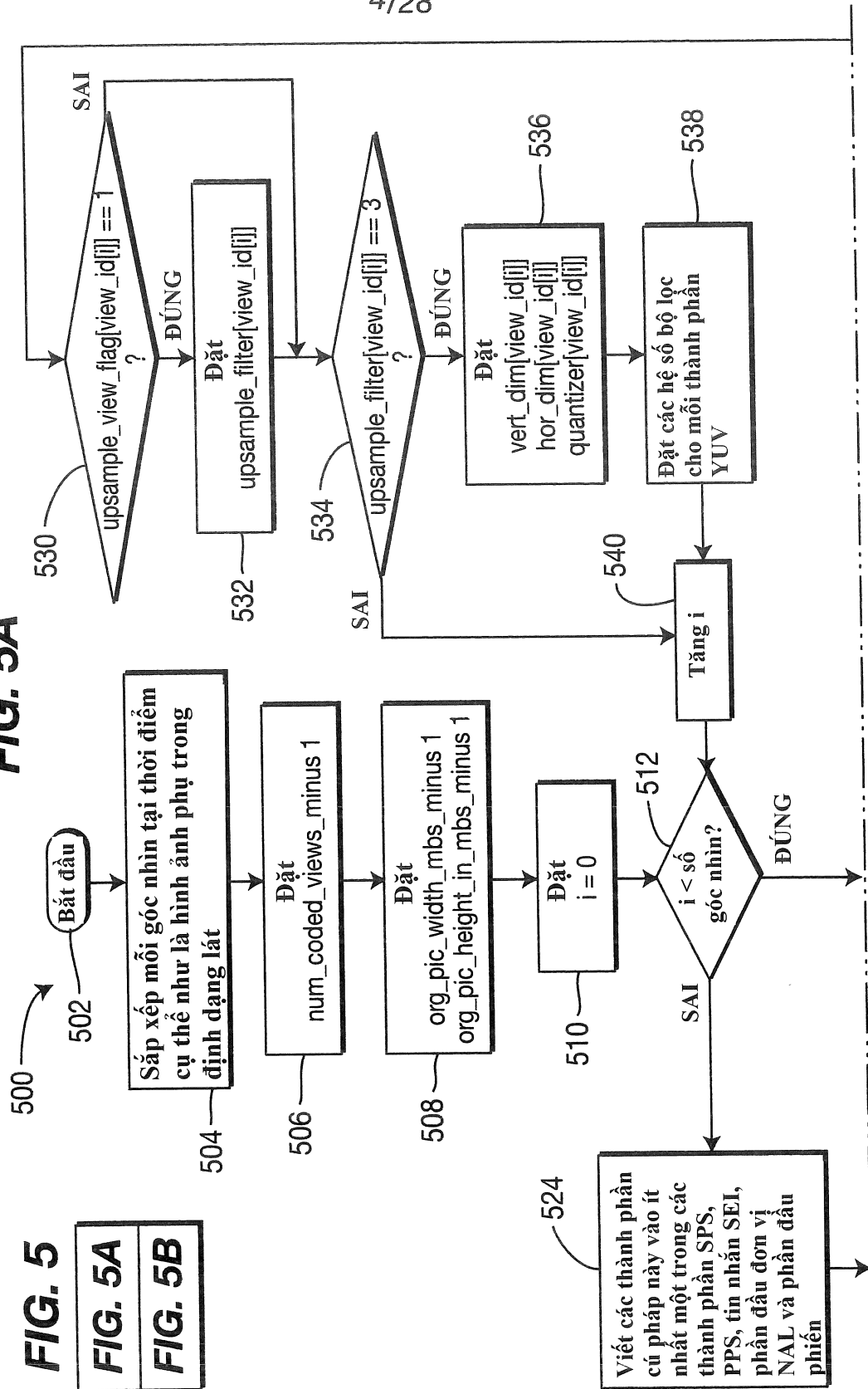


FIG. 4

4/28

FIG. 5A



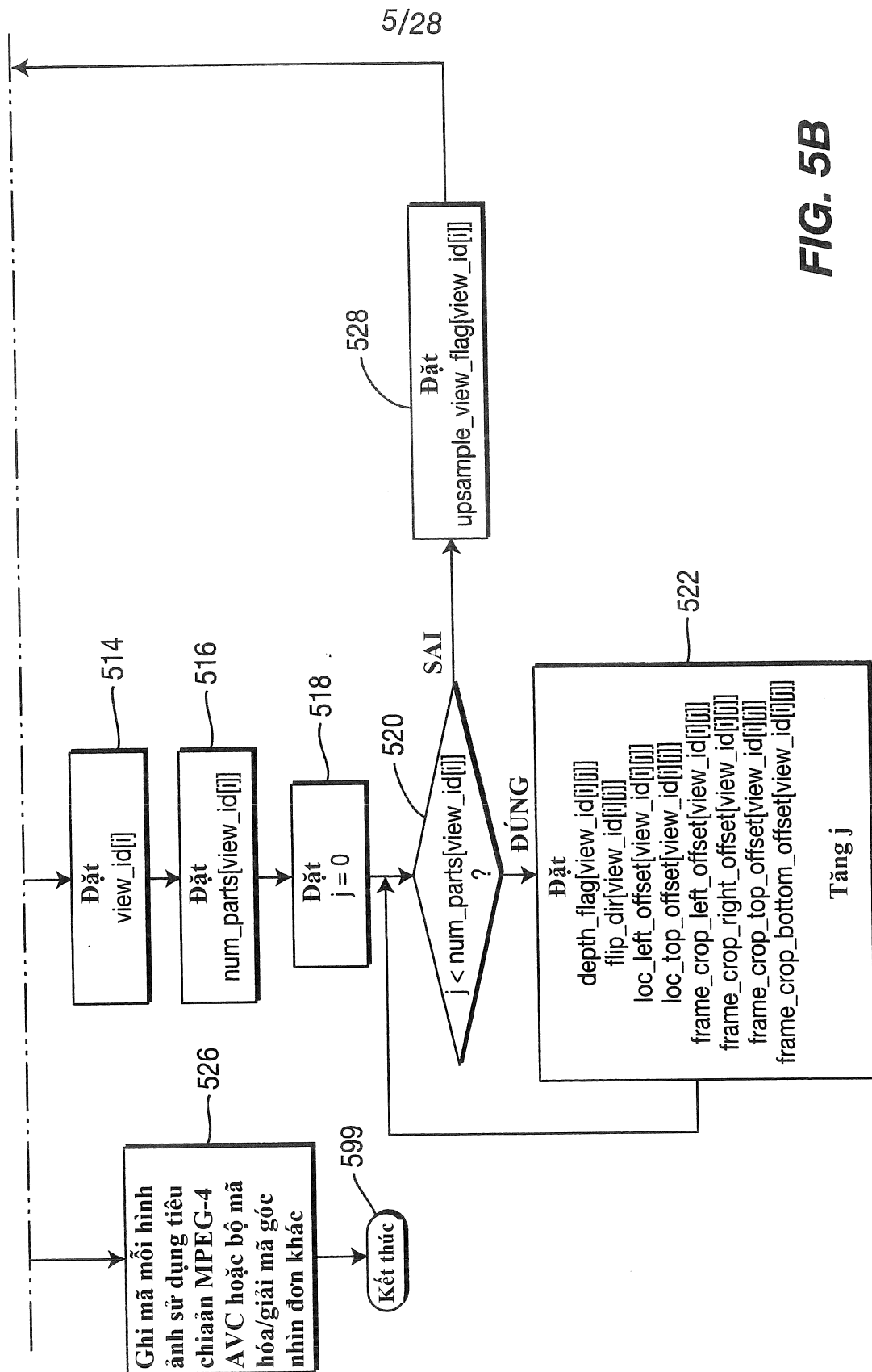


FIG. 5B

FIG. 6A

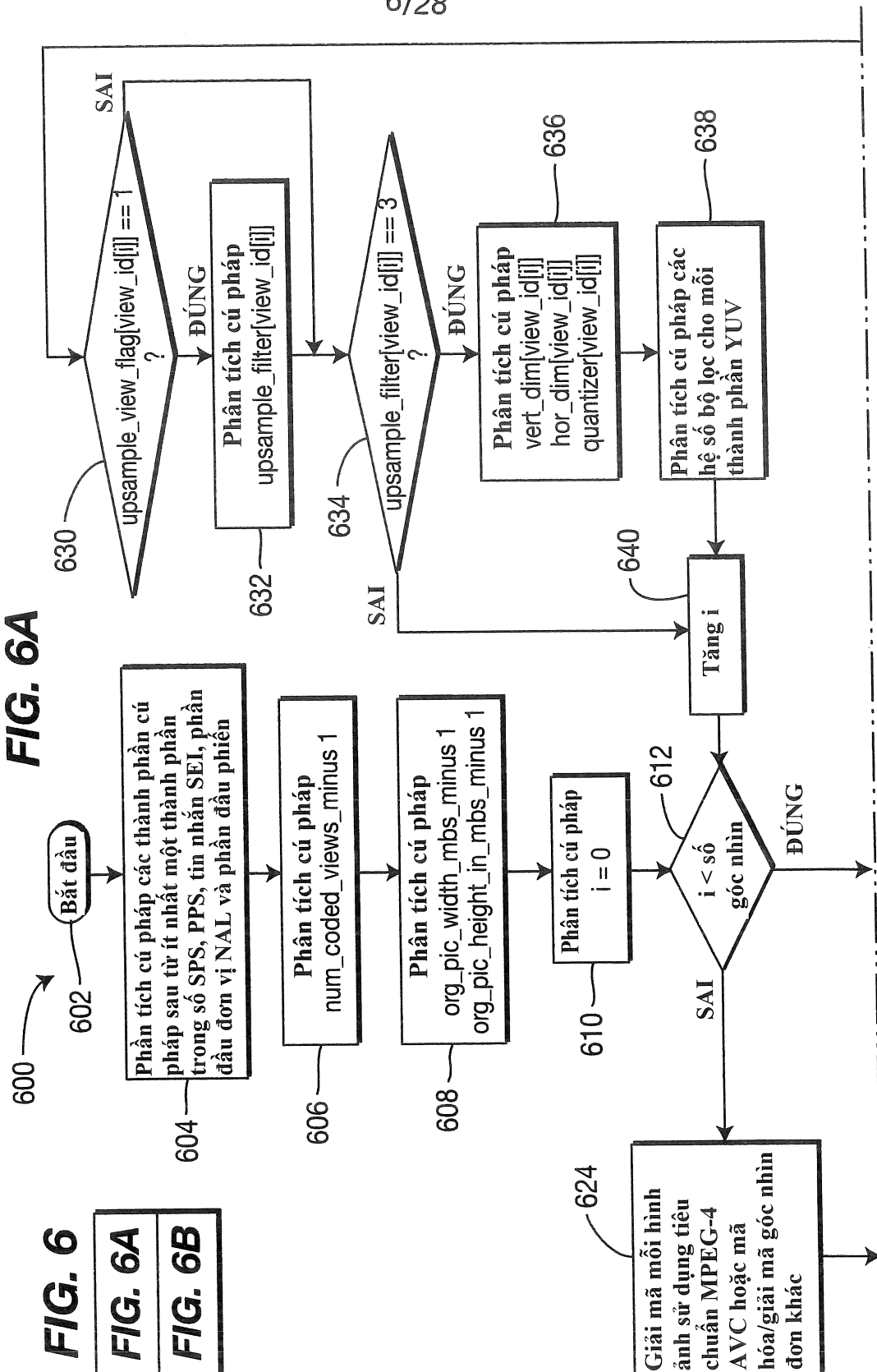


FIG. 6

FIG. 6A

FIG. 6B

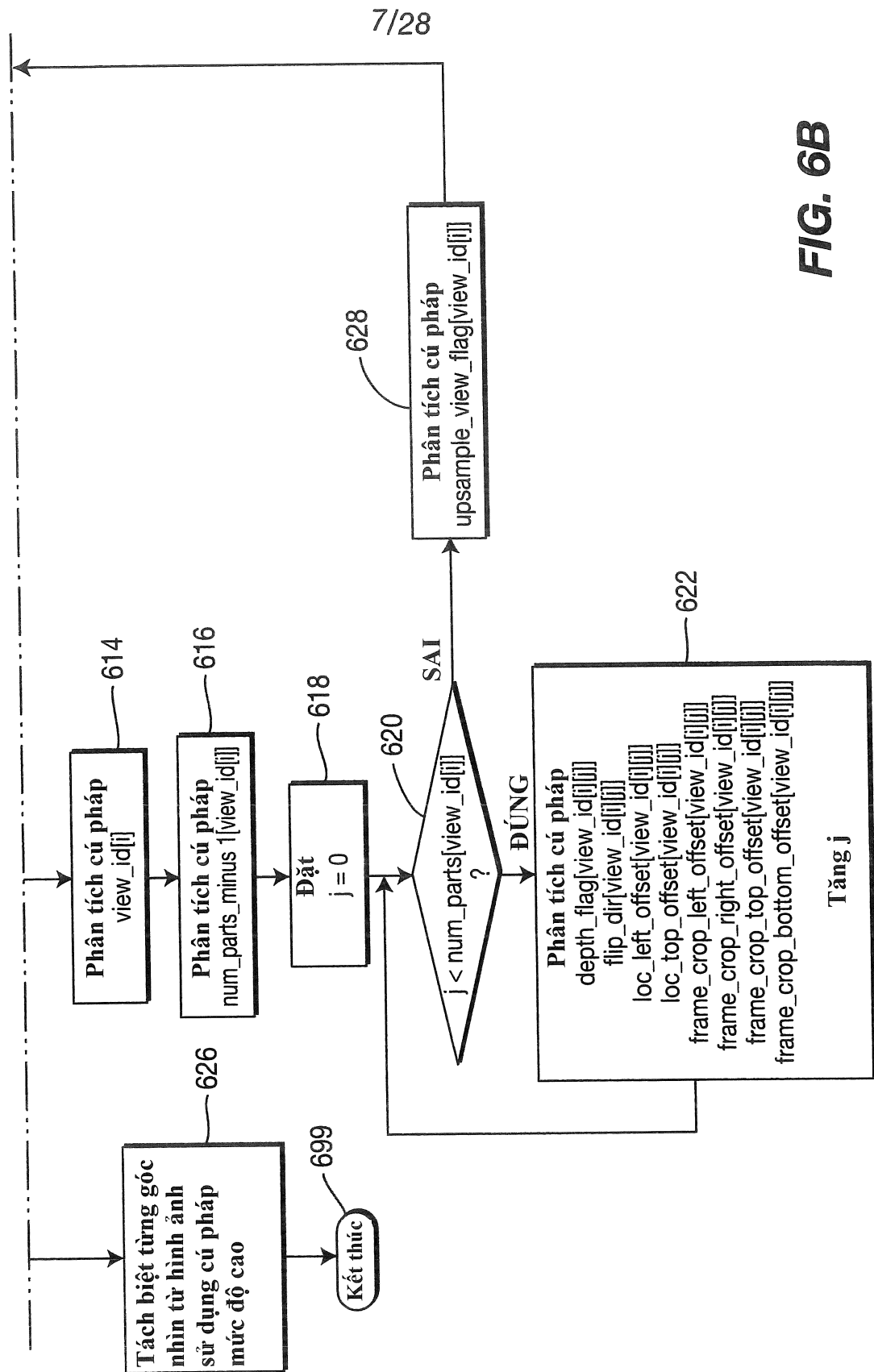


FIG. 6B

FIG. 7A

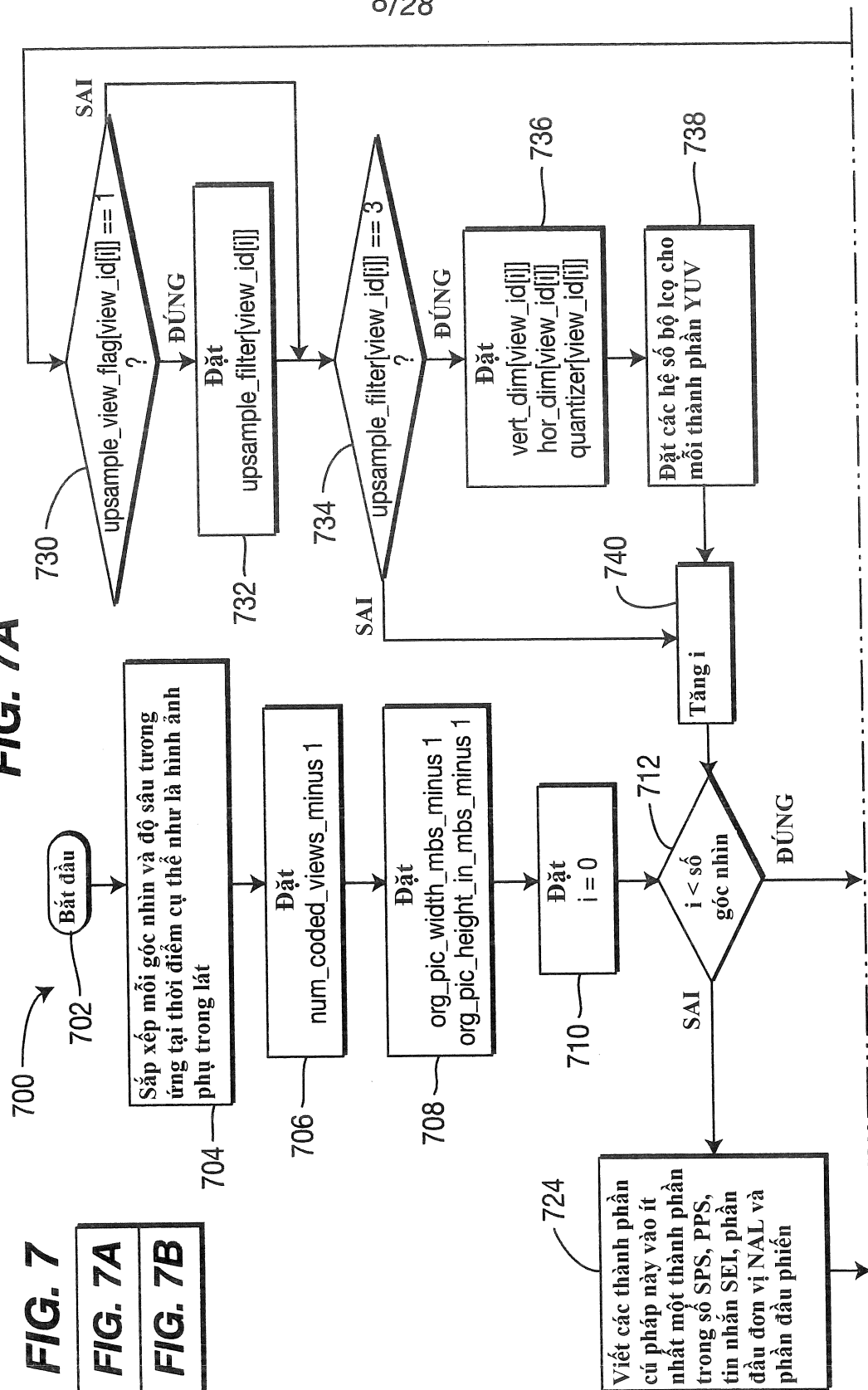


FIG. 7

FIG. 7A

FIG. 7B

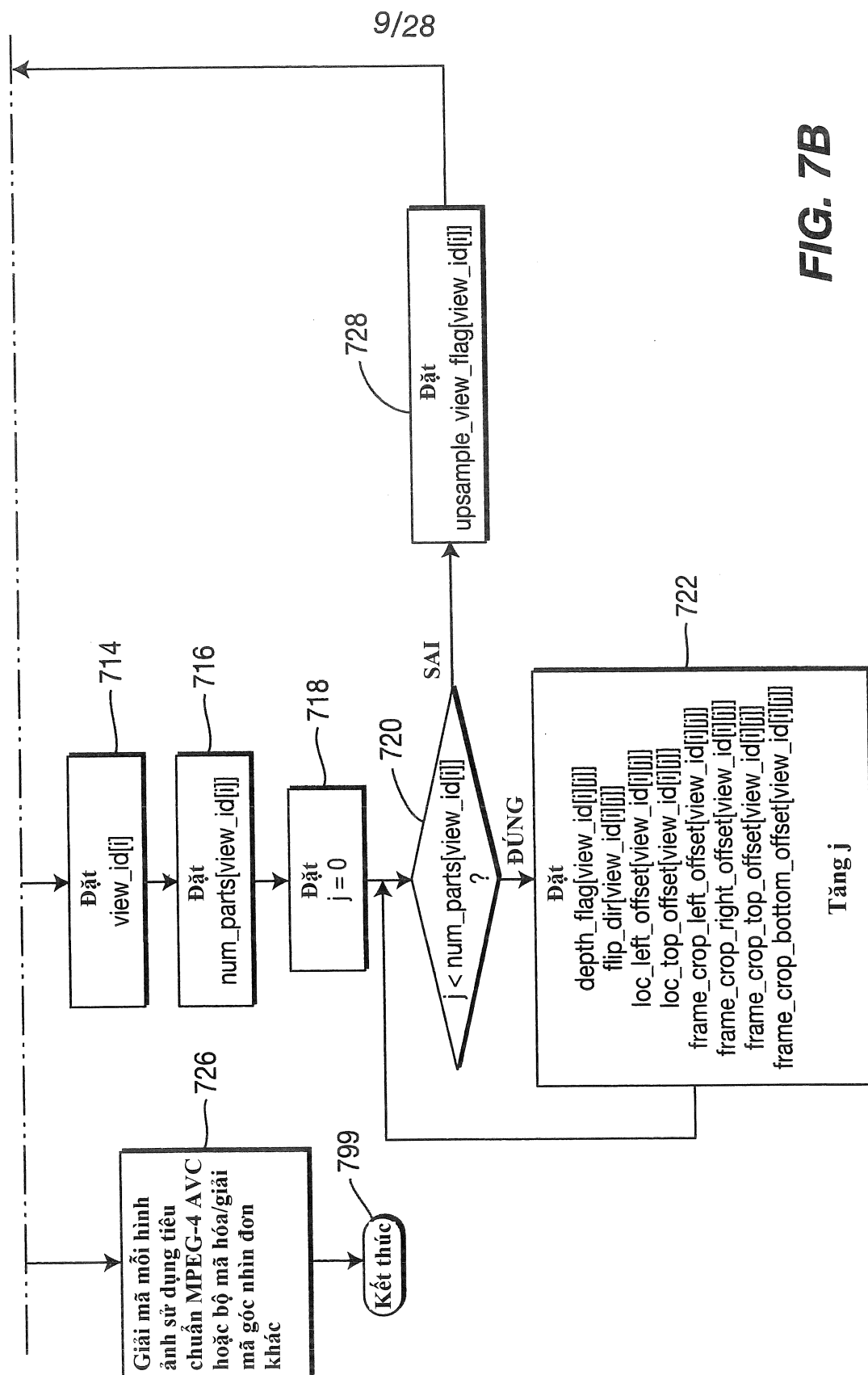
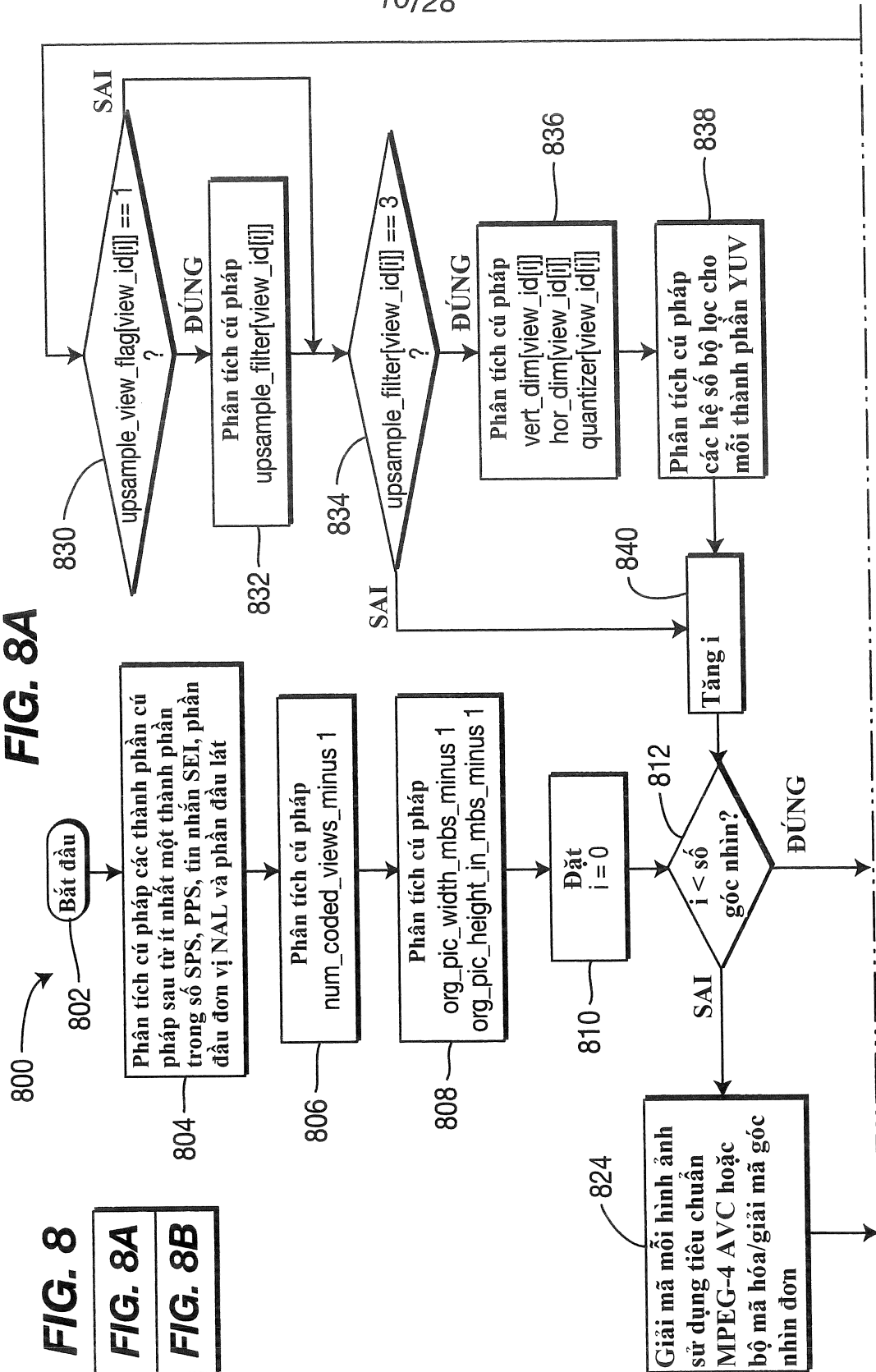


FIG. 7B

10/28

FIG. 8A

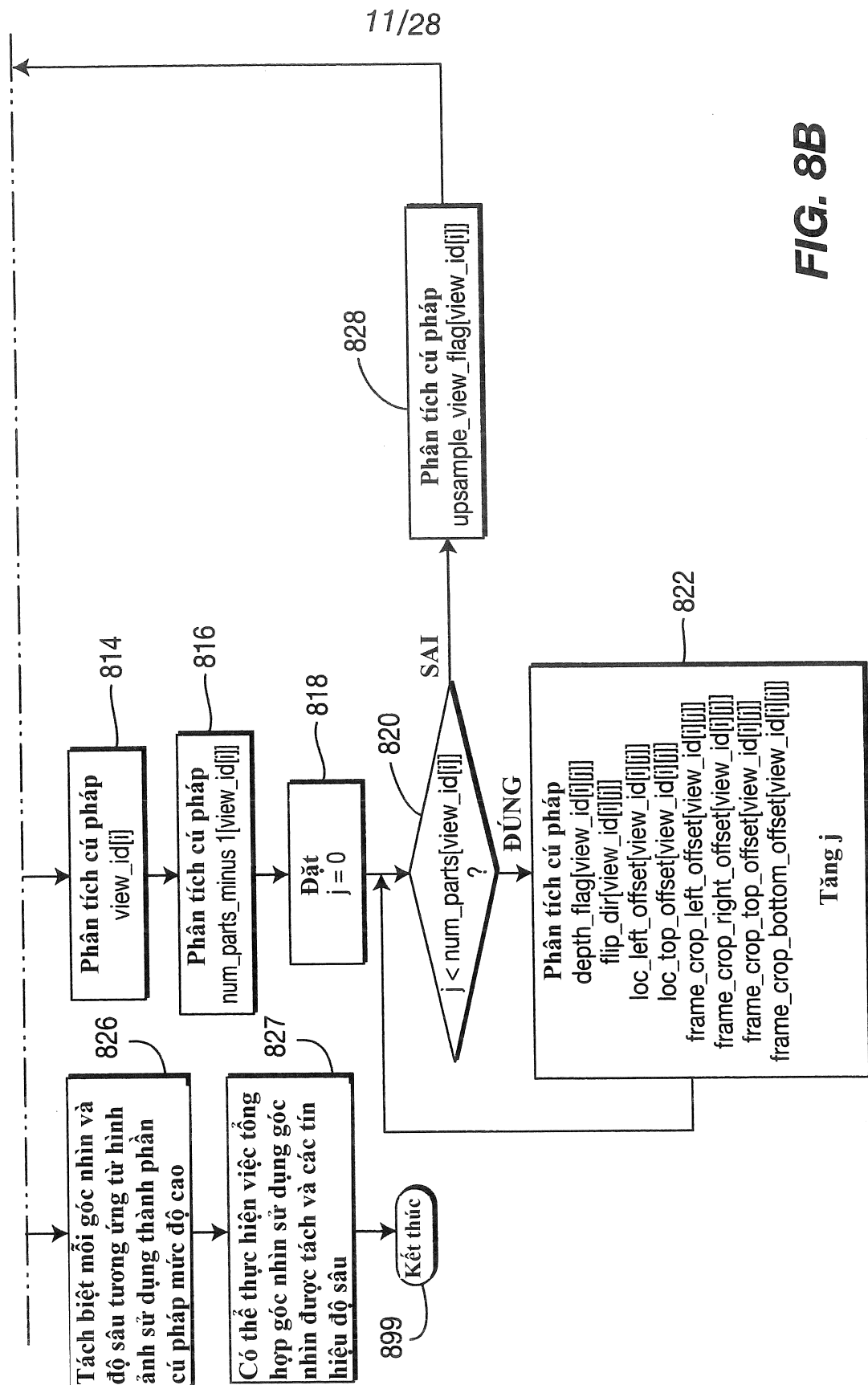
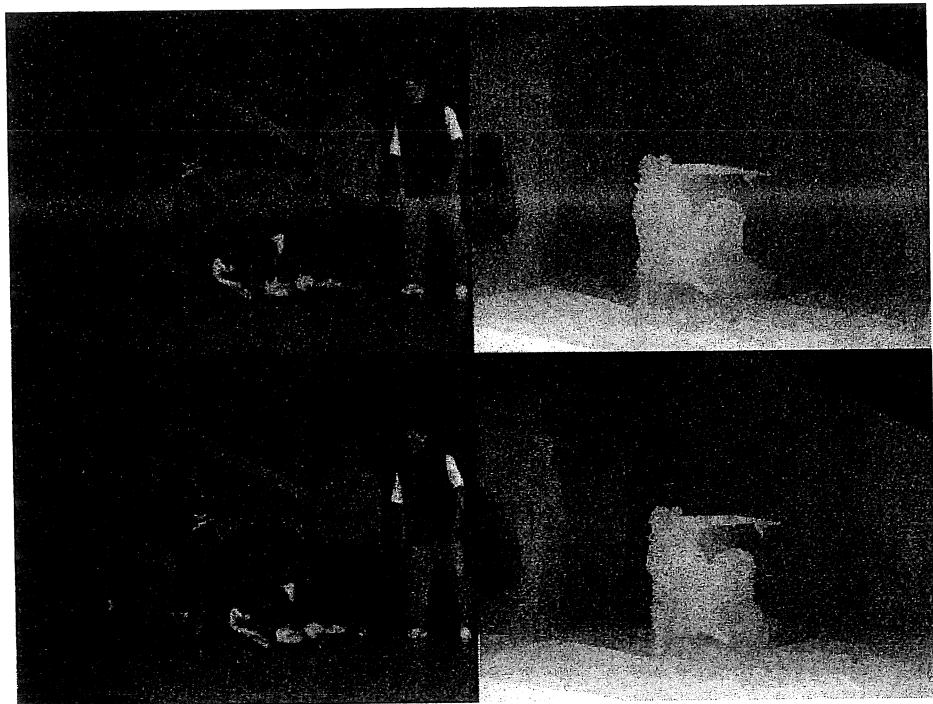
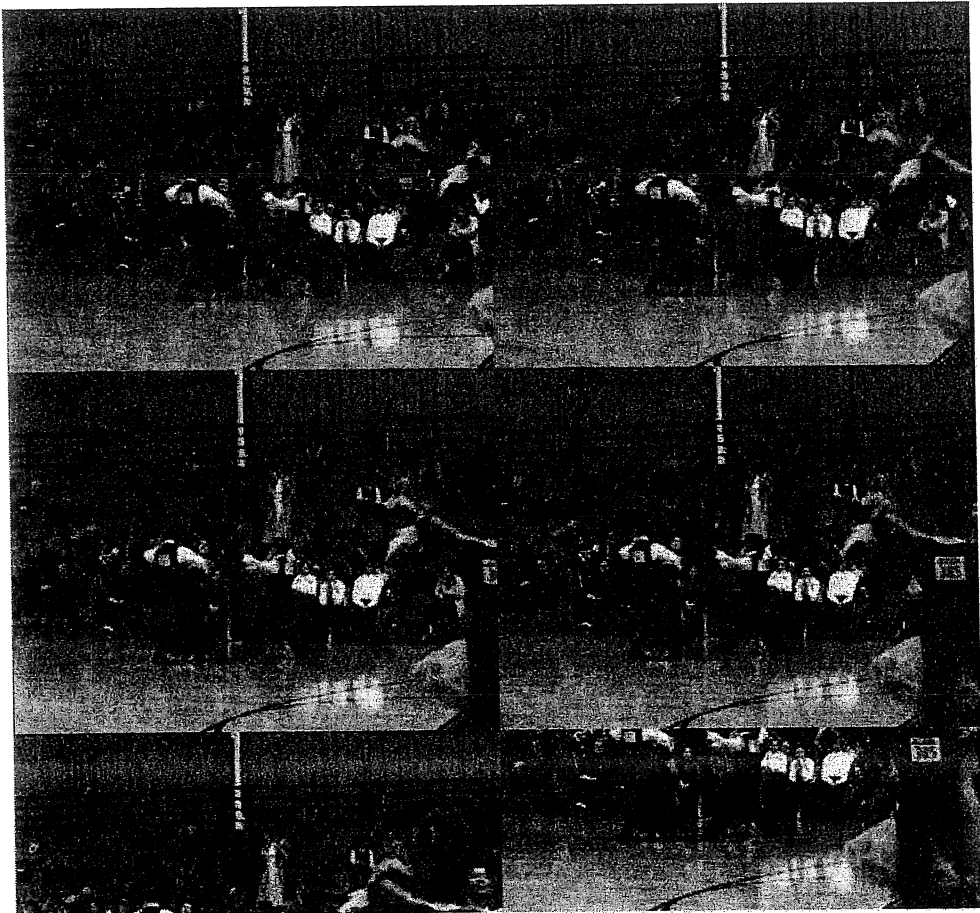


FIG. 8B

12/28

900**FIG. 9**1000**FIG. 10**

13/28

1100**FIG. 11**

14/28

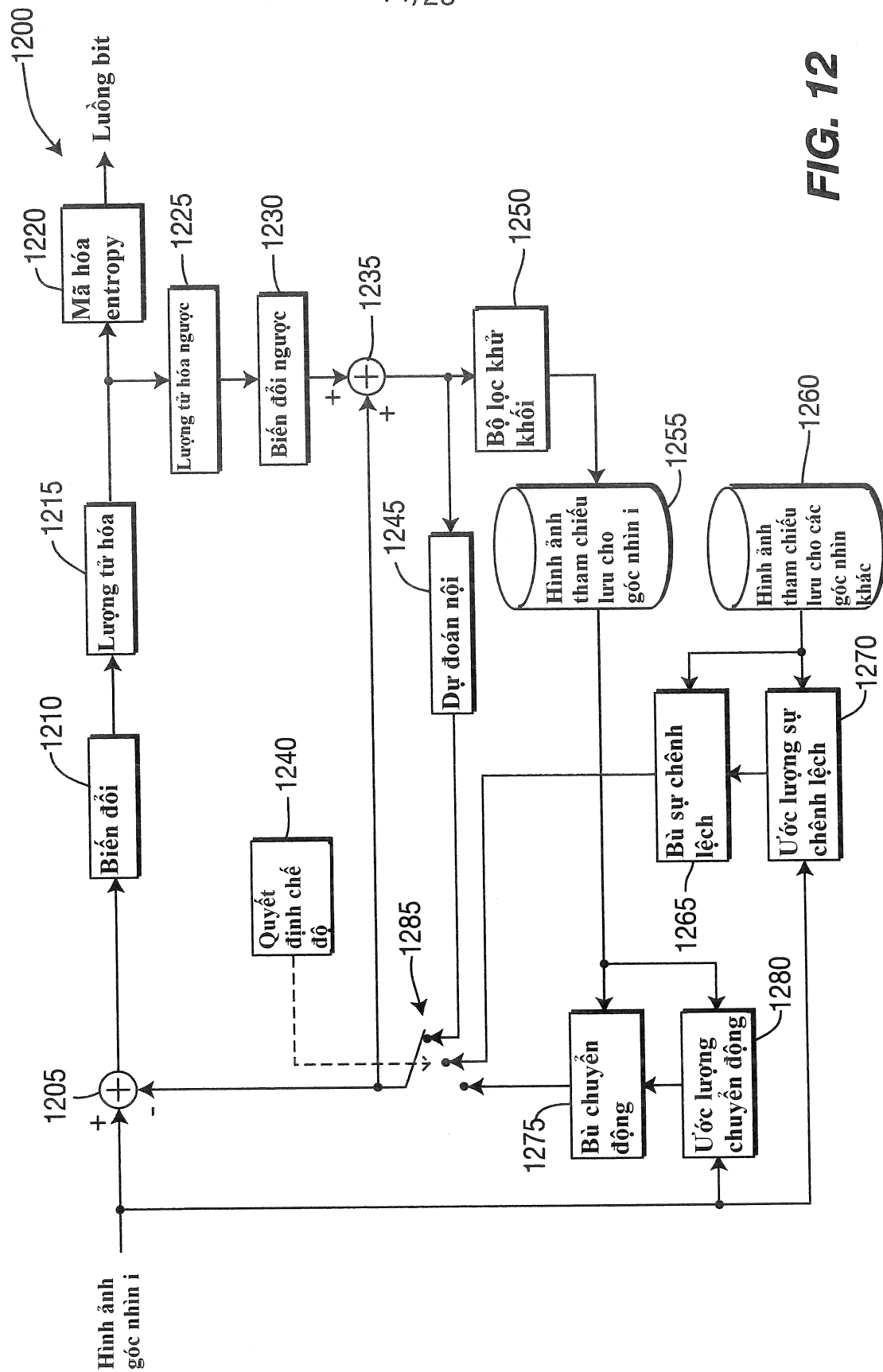


FIG. 12

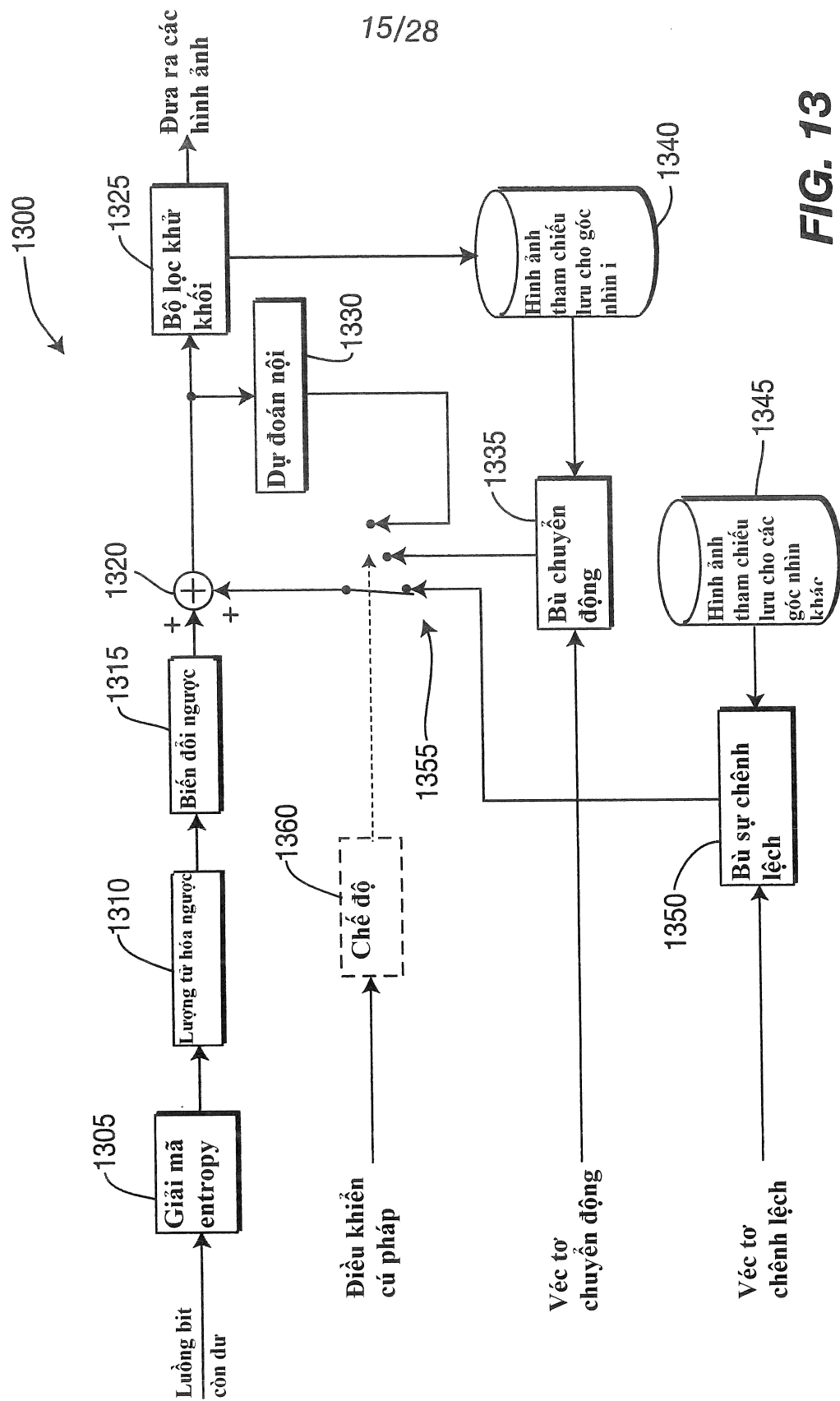


FIG. 13

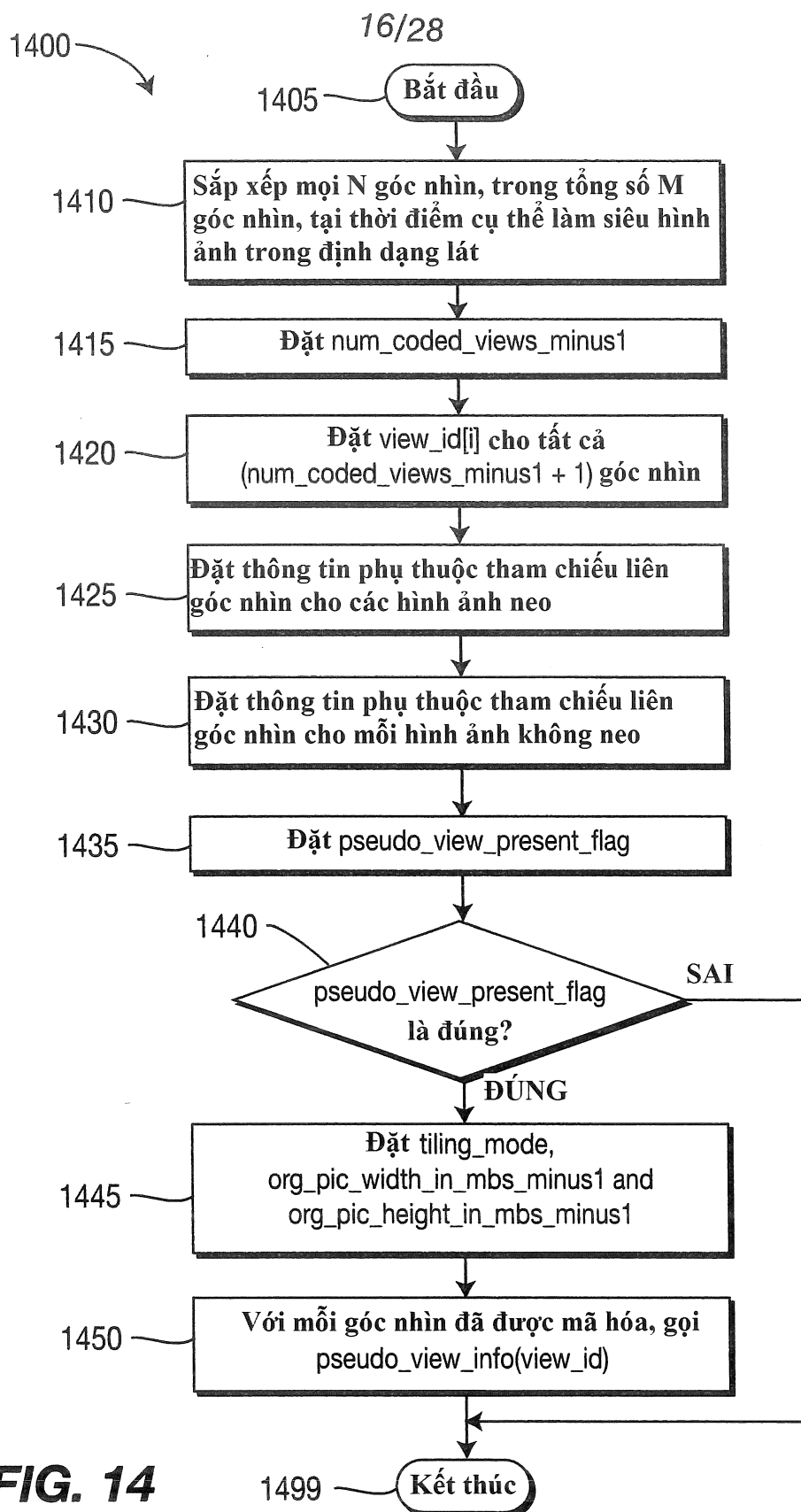
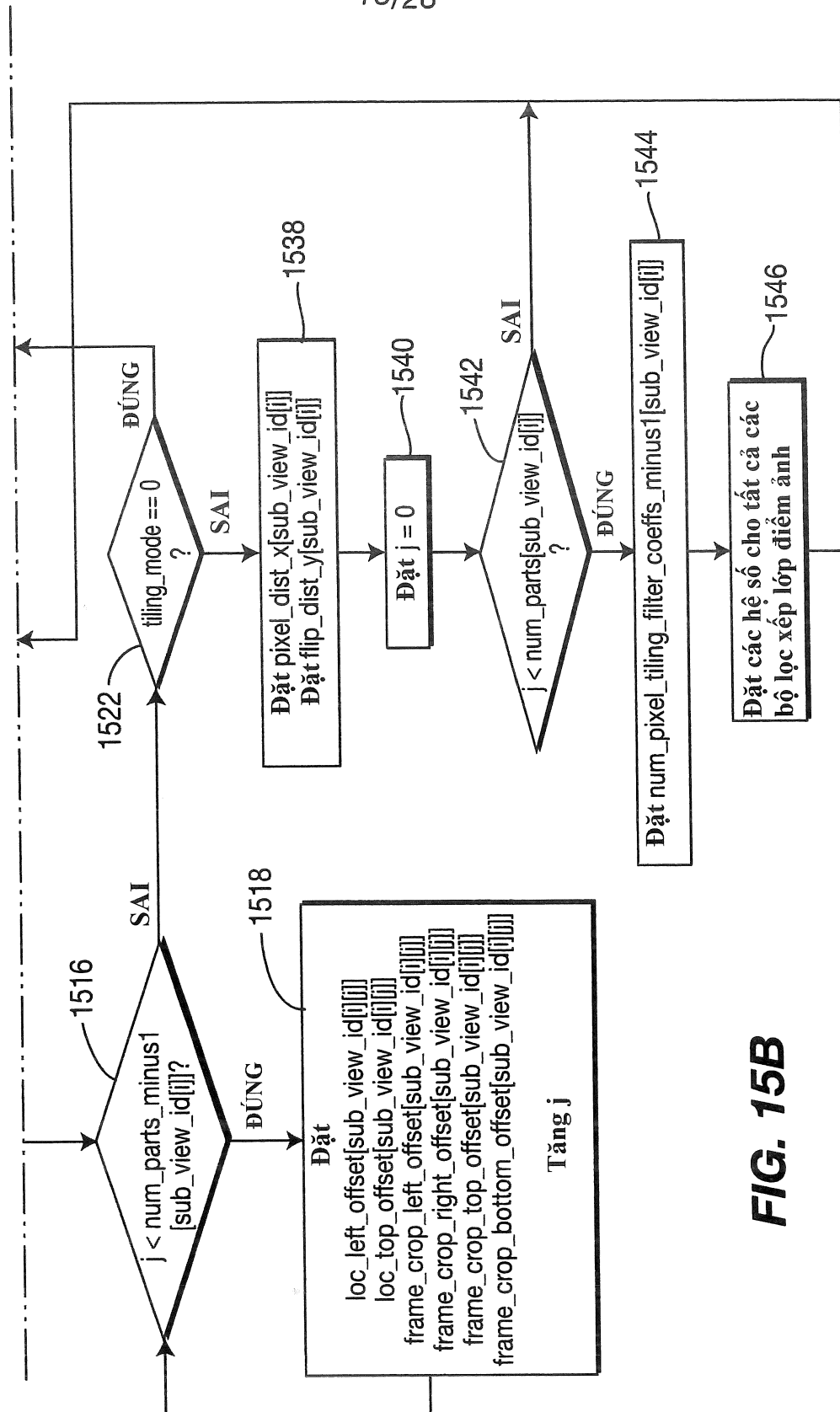


FIG. 14

18/28

**FIG. 15B**

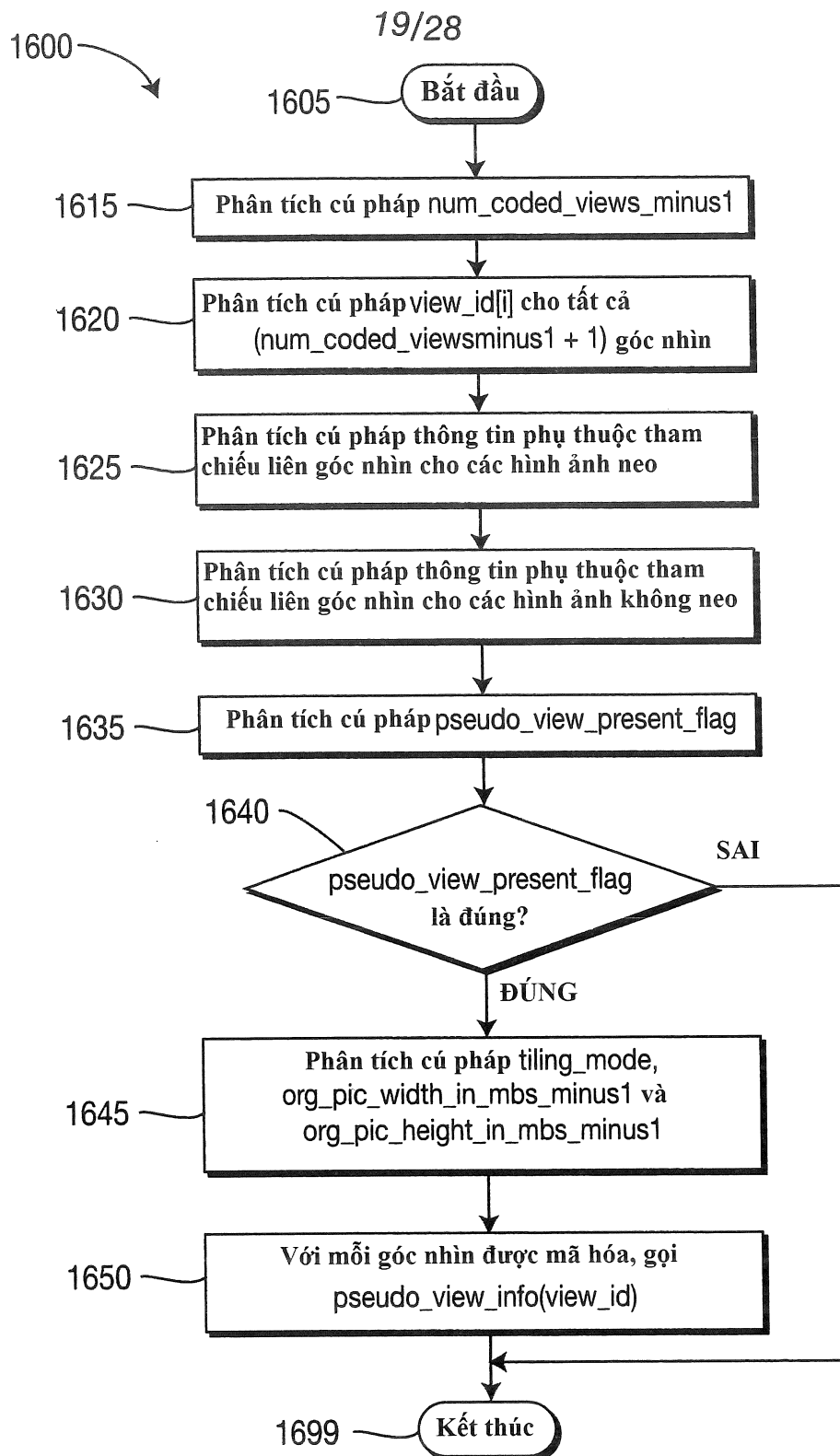
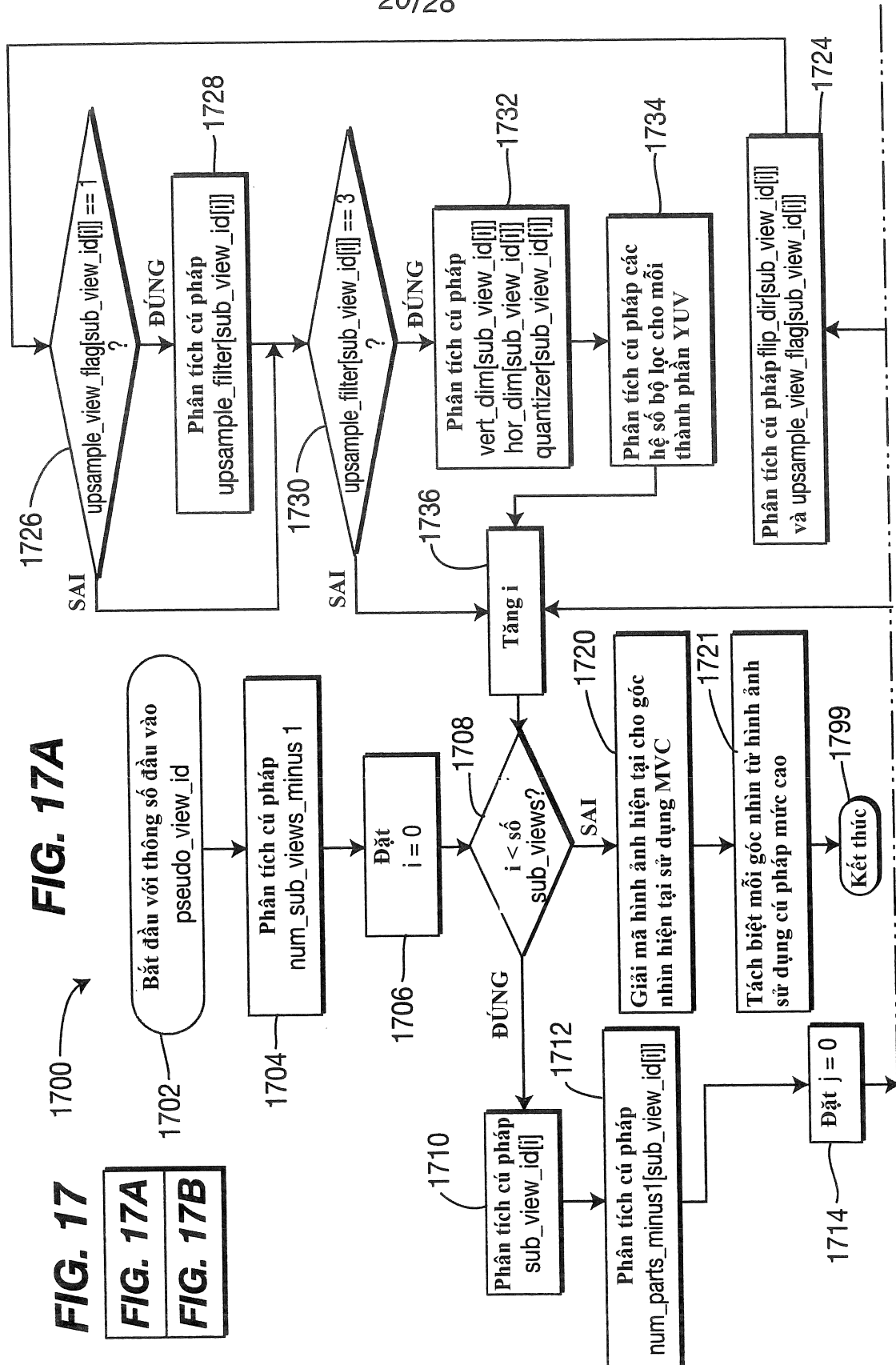
**FIG. 16**

FIG. 17A

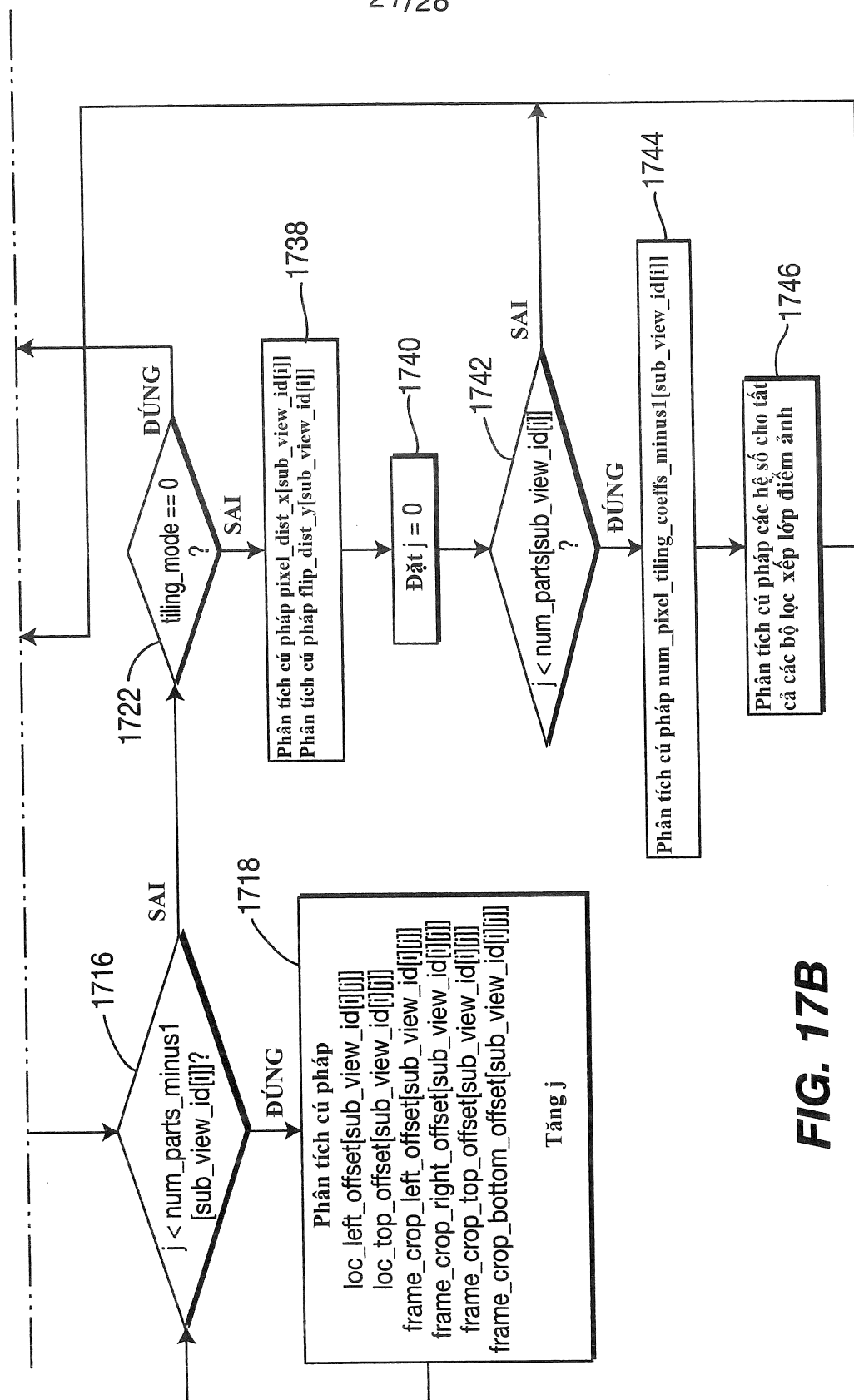
FIG. 17

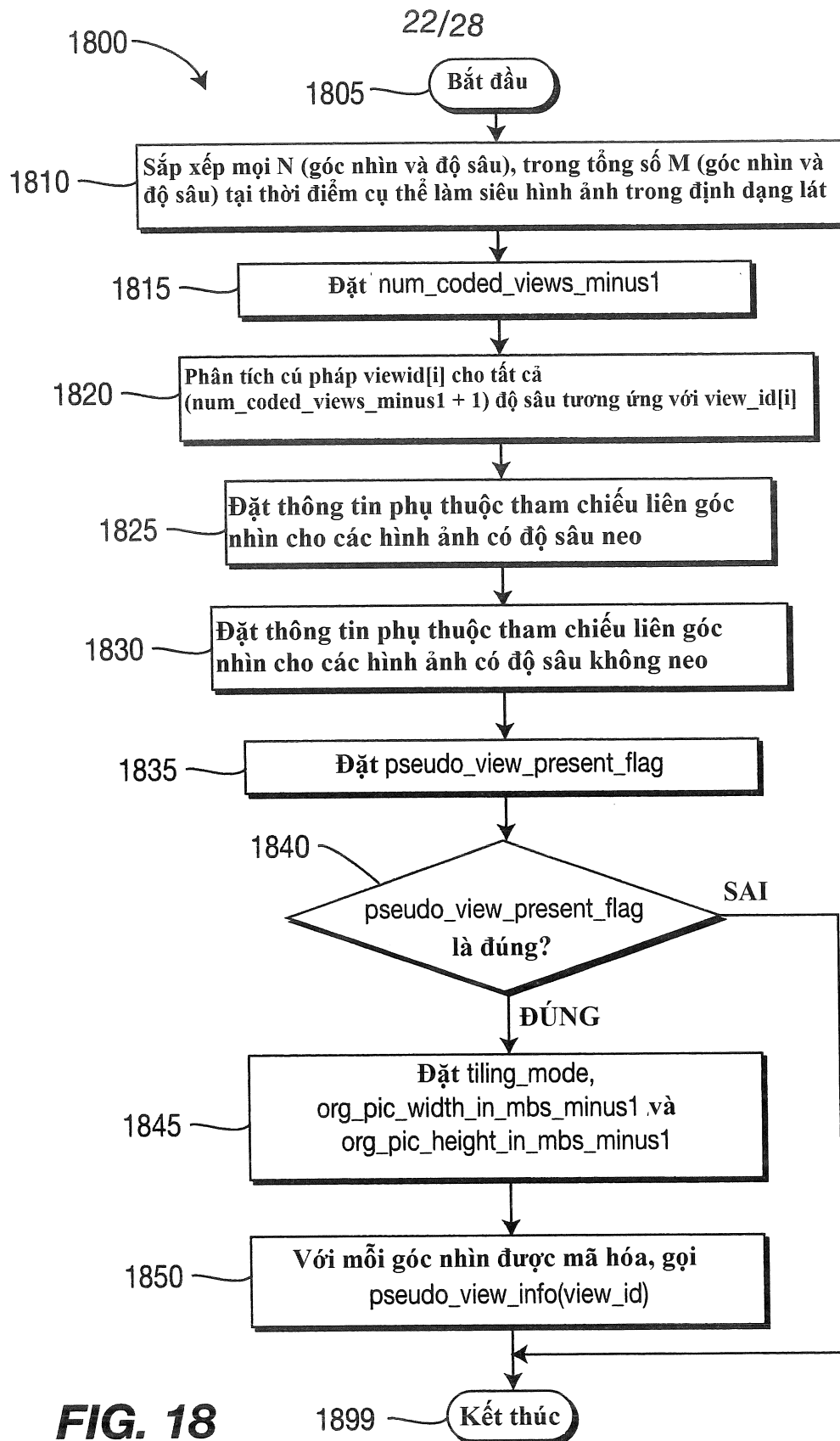
FIG. 17A

FIG. 17B

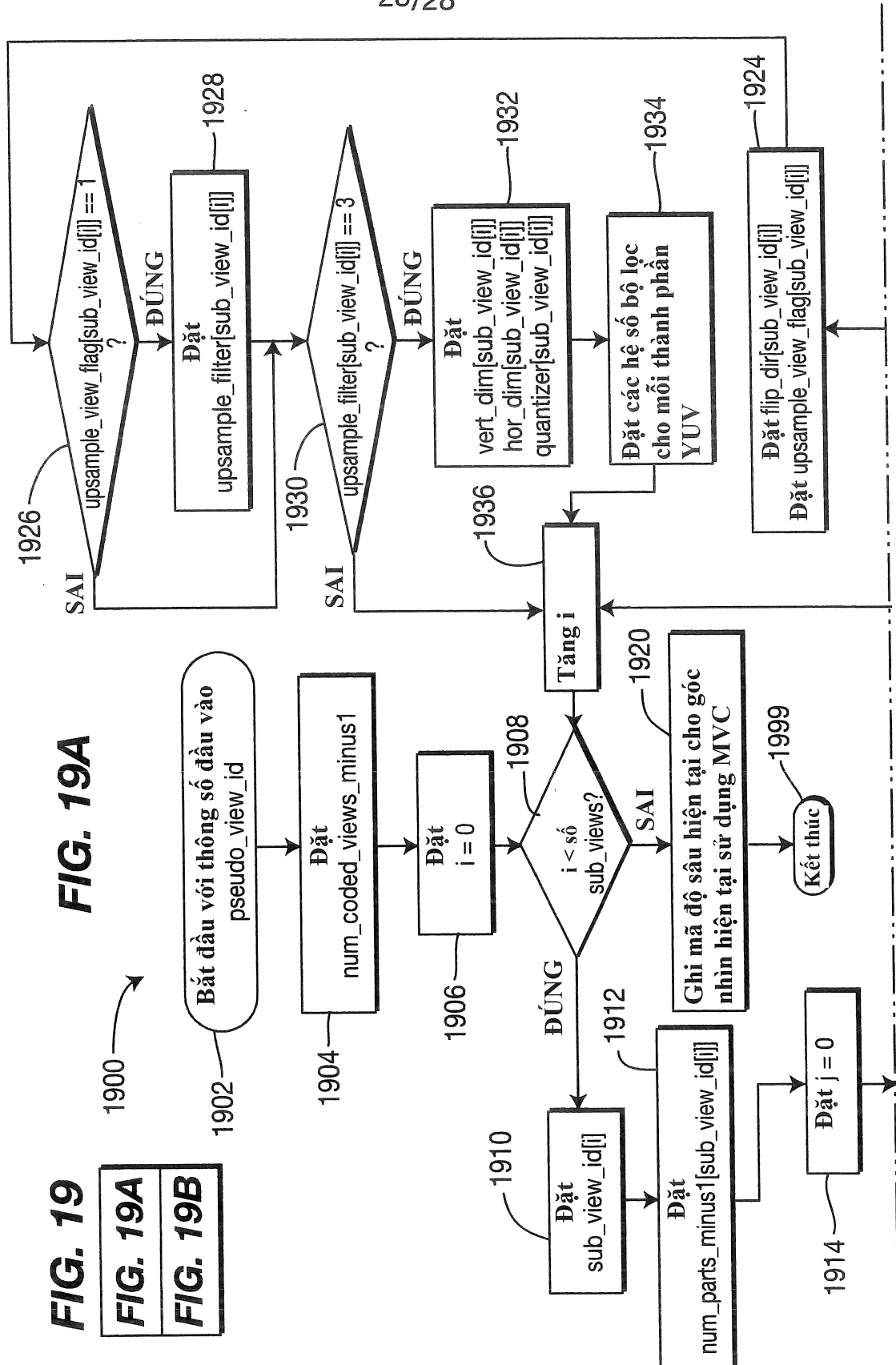


21/28

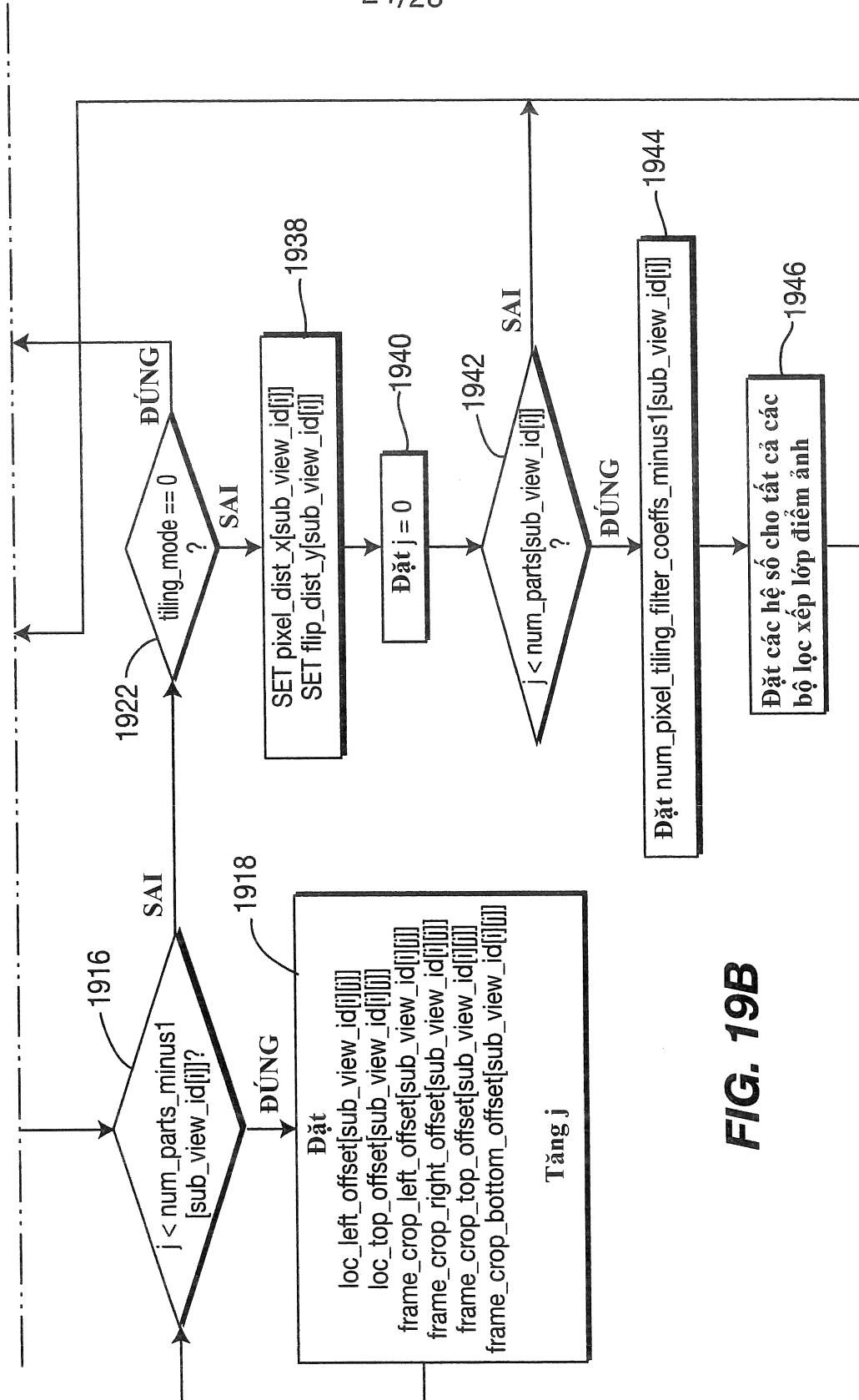
**FIG. 17B**

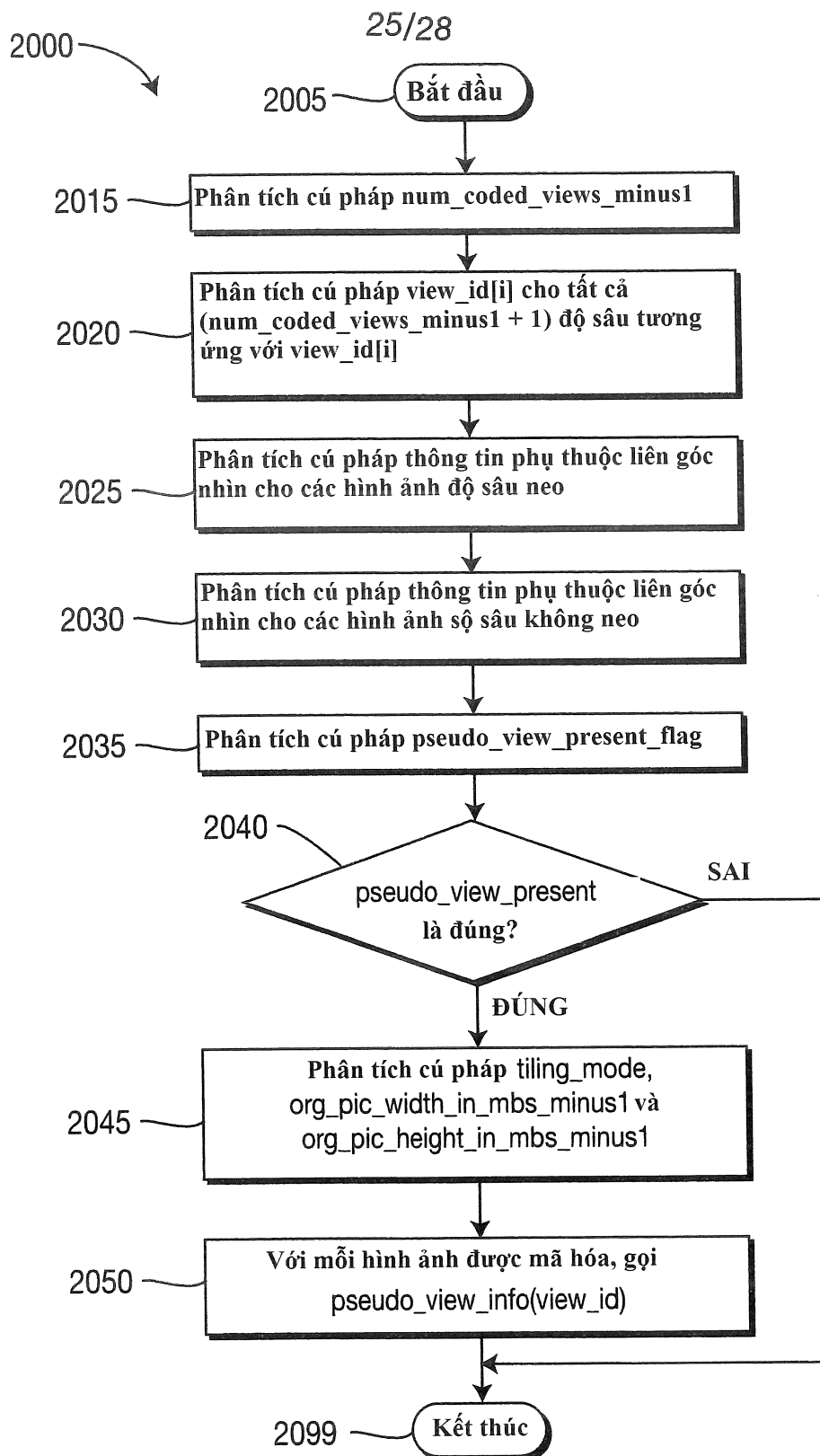


23/28

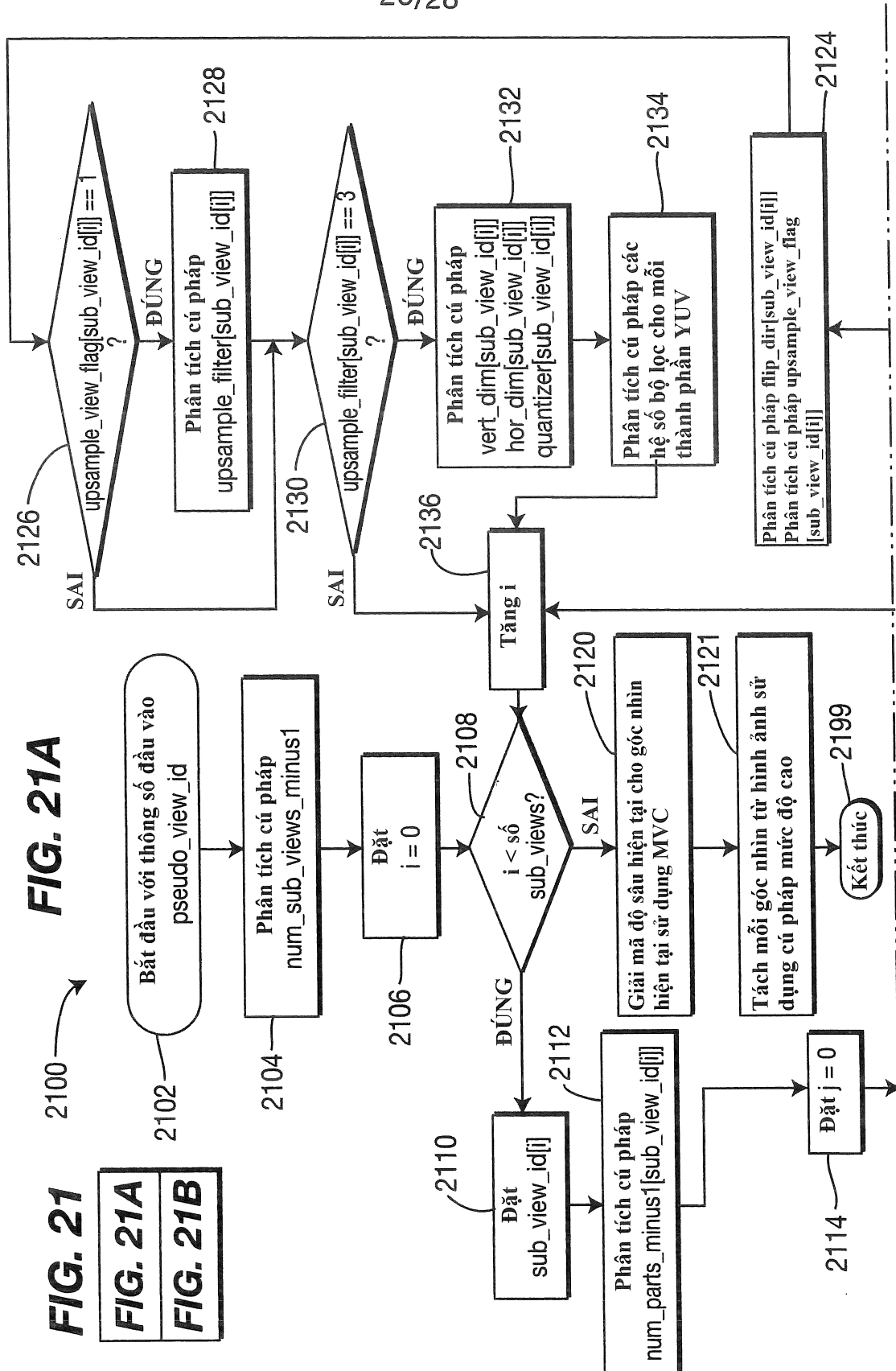
FIG. 19A**FIG. 19****FIG. 19A**
FIG. 19B

24/28

**FIG. 19B**

**FIG. 20**

26/28



27/28

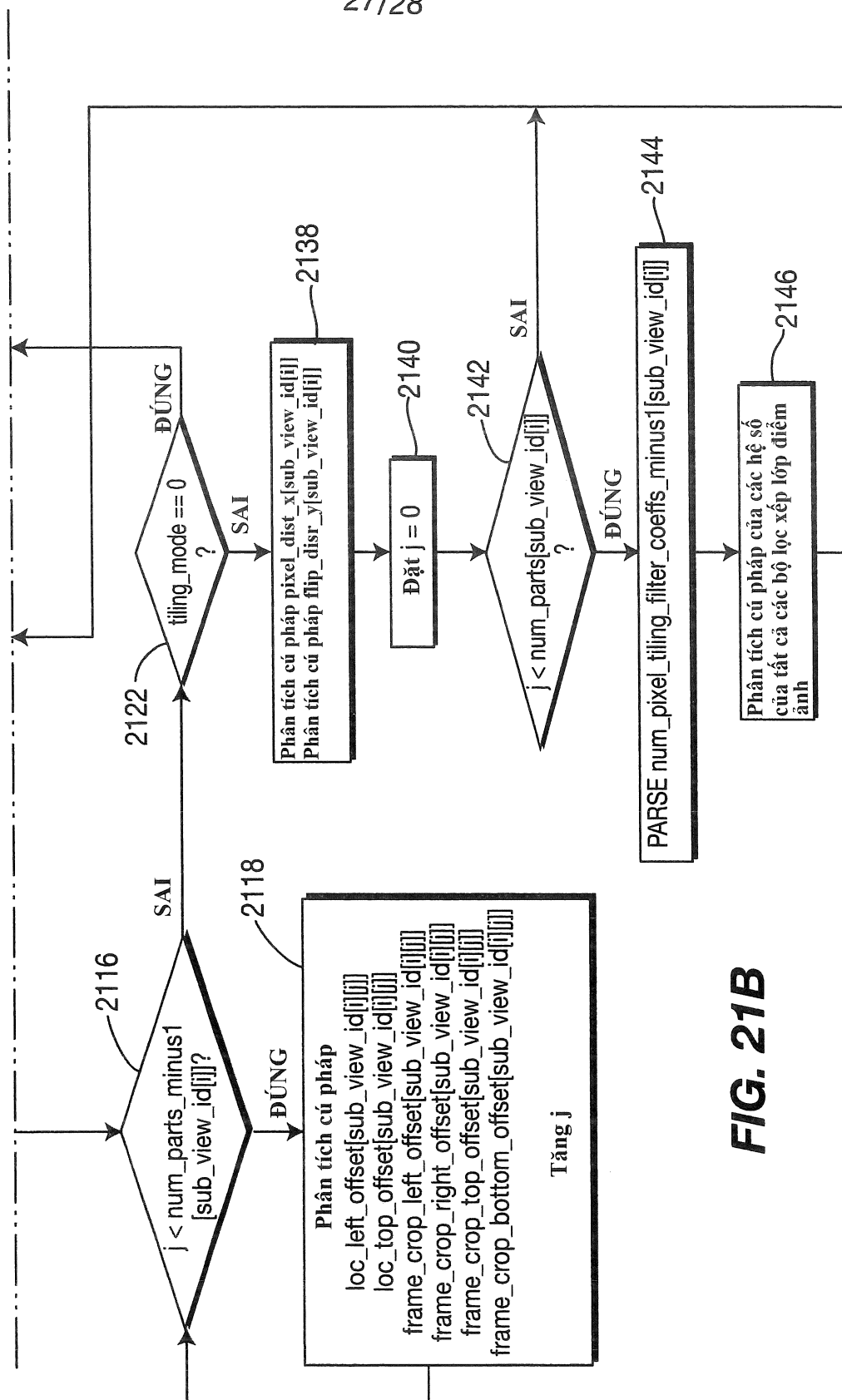
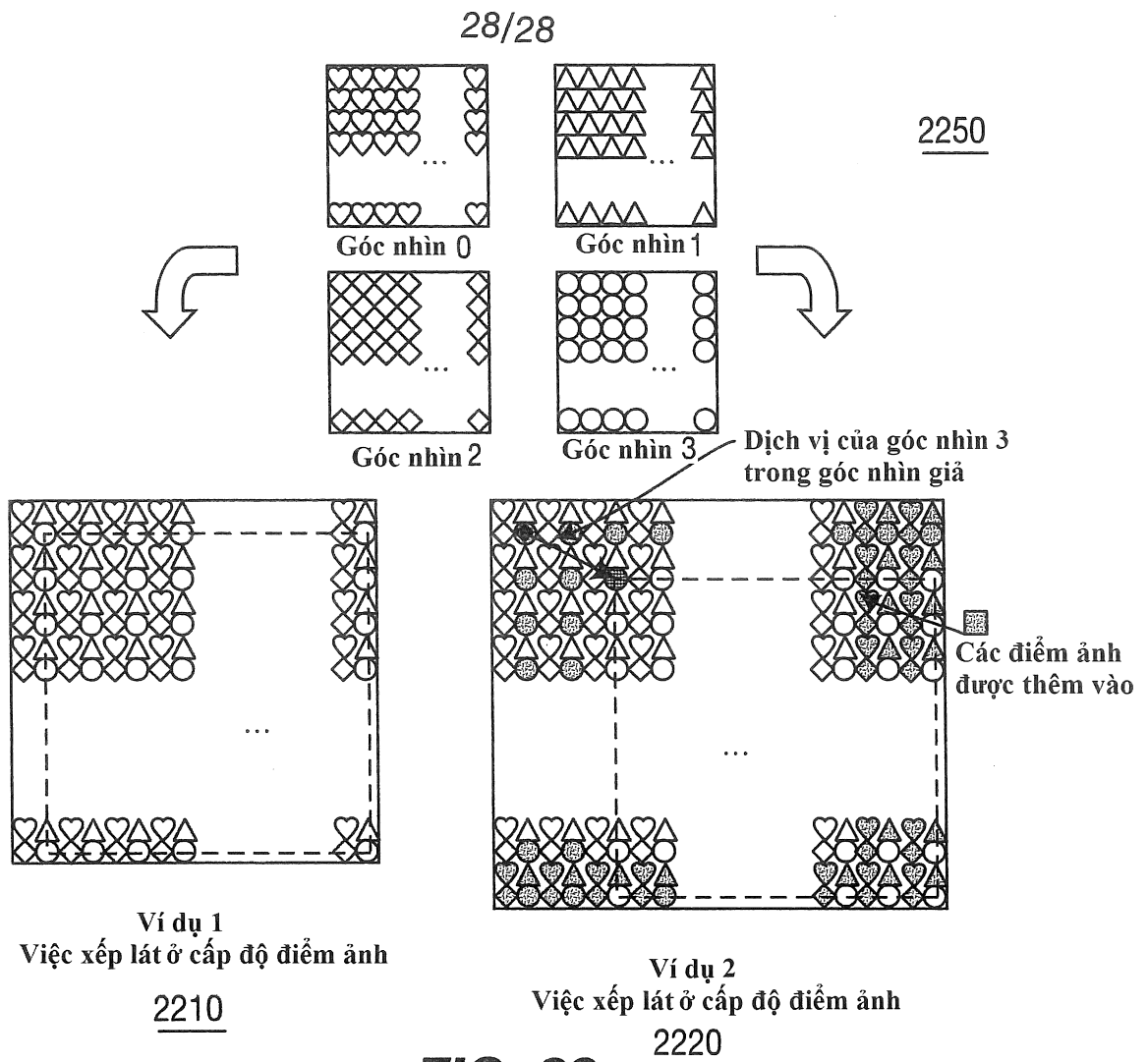


FIG. 21B

**FIG. 22**