

(19)



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS
ESPAÑA



(11) Número de publicación: **2 941 888**

(51) Int. Cl.:

H04N 19/597 (2014.01)
H04N 19/70 (2014.01)
H04N 19/46 (2014.01)
H04N 19/61 (2014.01)
H04N 13/00 (2006.01)

(12)

TRADUCCIÓN DE PATENTE EUROPEA

T3

(96) Fecha de presentación y número de la solicitud europea: **11.04.2008 E 21180823 (3)**

(97) Fecha y número de publicación de la concesión europea: **22.02.2023 EP 3920538**

(54) Título: **Organización en mosaico en codificación y decodificación de vídeo**

(30) Prioridad:

12.04.2007 US 923014 P
20.04.2007 US 925400 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
26.05.2023

(73) Titular/es:

DOLBY INTERNATIONAL AB (100.0%)
77 Sir John Rogerson's Quay Block C Grand
Canal Docklands
Dublin, D02 VK60, IE

(72) Inventor/es:

PANDIT, PURVIN BIBHAS;
YIN, PENG y
TIAN, DONG

(74) Agente/Representante:

LINAGE GONZÁLEZ, Rafael

ES 2 941 888 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Organización en mosaico en codificación y decodificación de vídeo

Referencia cruzada a solicitud relacionada

Esta solicitud es una solicitud divisional europea de la solicitud de patente europea EP 21150513.6 (referencia: A16060EP08), para la que se presentó el formulario 1001 de la OEP el 7 de enero de 2021.

Campo técnico

Los presentes principios están relacionados generalmente con codificación y/o decodificación de vídeo.

Antecedentes

Los fabricantes de pantallas de vídeo pueden usar una estructura para disponer u organizar en mosaico diferentes vistas en un único fotograma. Las vistas se pueden extraer luego de sus respectivas ubicaciones y ser representadas.

El documento EP 1 581 003 A1 describe un sistema para monitorizar una pluralidad de señales de vídeo en una red de vídeo interna, tal como un entorno de grabación de difusión, o red de cámaras de seguridad. El sistema comprende un monitor de imagen que tiene una pantalla en la que se expone una ilustración de vídeo de mosaico que contiene las señales de vídeo desde una pluralidad de cámaras u otros dispositivos de vídeo. La selección de una señal de vídeo de la ilustración de mosaico se puede realizar para exponer una versión a pantalla completa de esa señal.

Compendio

Según un aspecto general, se accede a una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen. Se accede a información que indica cómo se combinan las múltiples imágenes en la imagen de vídeo accedida. La imagen de vídeo se decodifica para proporcionar una representación decodificada de las múltiples imágenes combinadas. La información accedida y la imagen de vídeo decodificada se proporcionan como salida.

Según otro aspecto general, se genera información que indica cómo se combinan en una única imagen múltiples imágenes incluidas en una imagen de vídeo. La imagen de vídeo se codifica para proporcionar una representación codificada de las múltiples imágenes combinadas. La información generada y la imagen de vídeo codificada se proporcionan como salida.

Según otro aspecto general, una señal o estructura de señal incluye información que indica cómo se combinan en la única imagen de vídeo múltiples imágenes incluidas en una única imagen de vídeo. La señal o estructura de señal también incluye una representación codificada de las múltiples imágenes combinadas.

Según otro aspecto general, se accede a una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen. Se accede a información que indica cómo se combinan las múltiples imágenes en la imagen de vídeo accedida. La imagen de vídeo se decodifica para proporcionar una representación decodificada de al menos una de las múltiples imágenes. La información accedida y la representación decodificada se proporcionan como salida.

Según otro aspecto general, se accede a una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen. Se accede a información que indica cómo se combinan las múltiples imágenes en la imagen de vídeo accedida. La imagen de vídeo se decodifica para proporcionar una representación decodificada de las múltiples imágenes combinadas. Se recibe entrada de usuario que selecciona al menos una de las múltiples imágenes para exponer. Se proporciona una salida decodificada de la al menos una imagen seleccionada, la salida decodificada se proporciona sobre la base de la información accedida, la representación decodificada, y la entrada de usuario.

Los detalles de una o más implementaciones se presentan en los dibujos adjuntos y la siguiente descripción. Incluso si se describe de manera particular, debe quedar claro que se pueden configurar o plasmar implementaciones de diversas maneras.

La invención se establece en las reivindicaciones adjuntas.

Breve descripción de los dibujos

La FIG. 1 es un diagrama que muestra un ejemplo de cuatro vistas organizadas en mosaico en un único fotograma; la FIG. 2 es un diagrama que muestra un ejemplo de cuatro vistas volteadas y organizadas en mosaico en un único fotograma;

la FIG. 3 muestra un diagrama de bloques para un codificador de vídeo al que se pueden aplicar los presentes principios, según una realización de los presentes principios;

- la FIG. 4 muestra un diagrama de bloques para un decodificador de vídeo al que se pueden aplicar los presentes principios, según una realización de los presentes principios;
- la FIG. 5 es un diagrama de flujo para un método para codificar imágenes para una pluralidad de vistas usando el Estándar MPEG-4 AVC, según una realización de los presentes principios;
- 5 la FIG. 6 es un diagrama de flujo para un método para decodificar imágenes para una pluralidad de vistas usando el Estándar MPEG-4 AVC, según una realización de los presentes principios;
- la FIG. 7 es un diagrama de flujo para un método para codificar imágenes para una pluralidad de vistas y profundidades usando el Estándar MPEG-4 AVC, según una realización de los presentes principios;
- 10 la FIG. 8 es un diagrama de flujo para un método para decodificar imágenes para una pluralidad de vistas y profundidades usando el Estándar MPEG-4 AVC, según una realización de los presentes principios;
- la FIG. 9 es un diagrama que muestra un ejemplo de una señal de profundidad, según una realización de los presentes principios;
- la FIG. 10 es un diagrama que muestra un ejemplo de una señal de profundidad añadida como pieza de mosaico, según una realización de los presentes principios;
- 15 la FIG. 11 es un diagrama que muestra un ejemplo de 5 vistas organizadas en mosaico en un único fotograma, según una realización de los presentes principios.
- la FIG. 12 es un diagrama de bloques para un codificador ejemplar de Codificación de vídeo multivista (MVC) al que se pueden aplicar los presentes principios, según una realización de los presentes principios;
- 20 la FIG. 13 es un diagrama de bloques para un decodificador ejemplar de Codificación de vídeo multivista (MVC) al que se pueden aplicar los presentes principios, según una realización de los presentes principios;
- la FIG. 14 es un diagrama de flujo para un método para procesar imágenes para una pluralidad de vistas en preparación para codificar las imágenes usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC, según una realización de los presentes principios;
- 25 la FIG. 15 es un diagrama de flujo para un método para codificar imágenes para una pluralidad de vistas usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC, según una realización de los presentes principios;
- la FIG. 16 es un diagrama de flujo para un método para procesar imágenes para una pluralidad de vistas en preparación para decodificar las imágenes usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC, según una realización de los presentes principios;
- 30 la FIG. 17 es un diagrama de flujo para un método para decodificar imágenes para una pluralidad de vistas usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC, según una realización de los presentes principios;
- la FIG. 18 es un diagrama de flujo para un método para procesar imágenes para una pluralidad de vistas y profundidades en preparación para codificar las imágenes usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC, según una realización de los presentes principios;
- 35 la FIG. 19 es un diagrama de flujo para un método para codificar imágenes para una pluralidad de vistas y profundidades usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC, según una realización de los presentes principios;
- la FIG. 20 es un diagrama de flujo para un método para procesar imágenes para una pluralidad de vistas y profundidades en preparación para decodificar las imágenes usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC, según una realización de los presentes principios;
- 40 la FIG. 21 es un diagrama de flujo para un método para decodificar imágenes para una pluralidad de vistas y profundidades usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC, según una realización de los presentes principios;
- 45 la FIG. 22 es un diagrama que muestra ejemplos de organización en mosaico a nivel de píxel, según una realización de los presentes principios; y
- la FIG. 23 muestra un diagrama de bloques para un dispositivo de procesamiento de vídeo al que se pueden aplicar los presentes principios, según una realización de los presentes principios.

Descripción detallada

Diversas implementaciones se dirigen a métodos y aparatos para organizar en mosaico vistas en codificación y decodificación de vídeo. Se apreciará que los expertos en la técnica podrán idear diversas disposiciones que, aunque no explícitamente descritas o mostradas en esta memoria, incorporan los presentes.

5 Todos los ejemplos y el lenguaje condicional mencionados en esta memoria están pensados para finalidades pedagógicas para ayudar al lector a entender los presentes principios y los conceptos a los que contribuyen el/los inventor(es) para impulsar la técnica, y se han de interpretar sin limitación a tales ejemplos y condiciones mencionados específicamente.

10 Además, todas las declaraciones en esta memoria que nombran principios, aspectos y realizaciones de los presentes principios, así como ejemplos específicos de los mismos, pretenden englobar ambos equivalentes estructurales y funcionales de los mismos. Adicionalmente, se pretende que tales equivalentes incluyan tanto equivalentes actualmente conocidos así como equivalentes desarrollados en el futuro, es decir, elementos desarrollados que realicen la misma función, independientemente de la estructura.

15 Así, por ejemplo, los expertos en la técnica apreciarán que los diagramas de bloques presentados en esta memoria representan vistas conceptuales de circuitería ilustrativa que incorpora los presentes principios. De manera similar, se apreciará que diagramas de flujo, diagramas de transición de estado, pseudocódigo, y similares representan diversos procesos que se pueden representar sustancialmente en medios legibles por ordenador y por tanto ser ejecutados por un ordenador o procesador, tanto si dicho ordenador o procesador se muestra explícitamente o no.

20 Las funciones de los diversos elementos mostrados en las figuras se pueden proporcionar mediante el uso de hardware dedicado así como hardware capaz de ejecutar software en asociación con software apropiado. Cuando son proporcionadas por un procesador, las funciones pueden ser proporcionadas por un único procesador dedicado, por un único procesador compartido, o por una pluralidad de procesadores individuales, algunos de los cuales pueden ser compartidos. Además, el uso explícito del término "procesador" o "controlador" no se debe interpretar para referirse exclusivamente a hardware capaz de ejecutar software, y puede incluir implícitamente, sin limitación, hardware procesador de señales digitales ("DSP"), memoria de solo lectura ("ROM") para almacenar software, memoria de acceso aleatorio ("RAM"), y almacenamiento no volátil.

25 También se puede incluir otro hardware, convencional y/o personalizado. De manera similar, commutadores mostrados en las figuras son conceptuales únicamente. Su función puede ser llevada a cabo a través de la operación de lógica de programa, a través de lógica dedicada, a través de la interacción de control de programa y lógica dedicada, o incluso manualmente, la técnica particular es seleccionable por el implementador como se entenderá más específicamente a partir del contexto.

30 En las reivindicaciones de este documento, cualquier elemento expresado como medios para realizar una función especificada se pretende que englobe cualquier manera de realizar esa función, incluyendo, por ejemplo, a) una combinación de elementos de circuito que realizan esa función o b) software en cualquier forma, incluyendo, por lo tanto, firmware, microcódigo o algo semejante, combinado con circuitería apropiada para ejecutar ese software para realizar la función. Los presentes principios como son definidos por tales reivindicaciones residen en el hecho de que las funcionalidades proporcionadas por los diversos medios mencionados se combinan y juntan de la manera que reclaman las reivindicaciones. Así se considera que cualesquiera medios que puedan esas funcionalidades son equivalentes a los mostrados en esta memoria.

35 Referencia en la memoria descriptiva a "una realización" (o "una implementación") o "una realización" (o "una implementación") de los presentes principios significa que un rasgo, estructura, característica, etc. particulares descritos en conexión con la realización se incluyen en al menos una realización de los presentes principios. Así, las apariciones de la frase "en una realización" o "en una realización" que aparecen en diversos lugares por toda la memoria descriptiva no necesariamente hacen referencia todas a la misma realización.

40 Se tiene que apreciar que el uso de los términos "y/o" y "al menos uno de", por ejemplo, en los casos de "A y/o B" y "al menos uno de A y B", pretende para englobar la selección de la primera opción enumerada (A) únicamente, o la selección de la segunda opción enumerada (B) únicamente, o la selección de ambas opciones (A y B). Como ejemplo adicional, en los casos de "A, B, y/o C" y "al menos uno de A, B, y C", tal frase pretende englobar la selección de la primera opción enumerada (A) únicamente, o la selección de la segunda opción enumerada (B) únicamente, o la selección de la tercera opción enumerada (C) únicamente, o la selección de la primera y el segundo opciones enumeradas (A y B) únicamente, o la selección de la primera y tercera opciones enumeradas (A y C) únicamente, o la selección de la segunda y tercera opciones enumeradas (B y C) únicamente, o la selección de las tres opciones (A y B y C). Esto se puede extender, como es fácilmente evidente para un experto en esta técnica y relacionadas, para tantos elementos enumerados.

45 Además, se tiene que apreciar que si bien una o más realizaciones de los presentes principios se describen en esta memoria con respecto al Estándar MPEG-4 AVC, los presentes principios no se limitan a solamente este estándar y, así, se puede utilizar con respecto a otros estándares, recomendaciones y extensiones del mismo, particularmente estándares de codificación de vídeo, recomendaciones, y extensiones de los mismos, incluidas extensiones del Estándar MPEG-4 AVC.

Además, se tiene que apreciar que si bien una o más de otras realizaciones de los presentes principios se describen en esta memoria con respecto a la extensión de codificación de vídeo multivista del Estándar MPEG-4 AVC, los presentes principios no se limitan solamente a esta extensión y/o este estándar y, así, se puede utilizar con respecto a otros estándares de codificación de vídeo, recomendaciones, y extensiones de los mismos relativos a codificación de vídeo multivista.

5 Codificación de vídeo multivista (MVC) es la estructura de compresión para la codificación de secuencias multivista. Una secuencia de Codificación de vídeo multivista (MVC) es un conjunto de dos o más secuencias de vídeo que capturan la misma escena desde un punto de vista diferente.

10 También, se tiene que apreciar que si bien una o más de otras realizaciones de los presentes principios se describen en esta memoria que usa información de profundidad con respecto a contenido de vídeo, los presentes principios no se limitan a tales realizaciones y, así, se pueden implementar otras realizaciones que no usan información de profundidad.

15 Adicionalmente, como se emplea en esta memoria, "sintaxis de alto nivel" se refiere a sintaxis presente en el flujo de bits que reside jerárquicamente por encima de la capa macrobloque. Por ejemplo, sintaxis de alto nivel, como se emplea en esta memoria, puede referirse a, aunque sin limitación a esto, sintaxis a nivel de encabezado de segmento, nivel de Información de Mejora Suplementaria (SEI, del inglés *Supplemental Enhancement Information*), nivel de conjunto de parámetros de imagen (PPS, del inglés *Picture Parameter Set*), nivel de conjunto de parámetros de secuencia (SPS, del inglés *Sequence Parameter Set*), el conjunto de Parámetros de Vista (VPS, del inglés *View Parameter Set*), y nivel de encabezado de unidad de Capa de Abstracción de Red (NAL, del inglés *Network Abstraction Layer*).

20 En la implementación actual de codificación multívdeo (MVC) basada en la Organización Internacional para Estandarización / Comisión Electrotécnica Internacional (ISO/IEC), estándar Moving Picture Experts Group-4 (MPEG-4) Codificación de Vídeo Avanzada (AVC) Parte 10 / Unión Internacional de Telecomunicaciones, Recomendación H.264 del Sector de Telecomunicaciones (ITU-T) (más adelante en esta memoria la "Estándar 25 MPEG-4 AVC"), el software de referencia logra predicción multivista al codificar cada vista con un único codificador y teniendo en consideración las referencias de vista en cruz. Cada vista es codificada por el codificador como flujo de bits separado en su resolución original y más tarde todos los flujos de bits se combinan para formar un único flujo de bits que luego se decodifica. Cada vista produce una salida decodificada YUV separada.

30 Otro planteamiento para predicción multivista implica agrupar un conjunto de vistas en pseudovistas. En un ejemplo de este planteamiento, se pueden organizar en mosaico las imágenes de N vistas del total de M vistas (muestreadas al mismo tiempo) en un fotograma más grande o un superfotograma con posible submuestreo u otras operaciones. Cambiando a la FIG. 1, un ejemplo de cuatro vistas organizadas en mosaico en un único fotograma se indica generalmente con el numeral de referencia 100. Las cuatro vistas están en su orientación normal.

35 Cambiando a la FIG. 2, un ejemplo de cuatro vistas volteadas y organizadas en mosaico en un único fotograma se indica generalmente con el numeral de referencia 200. La vista superior-izquierda está en su orientación normal. La vista superior-derecha está volteada horizontalmente. La vista inferior-izquierda está volteada verticalmente. La vista inferior-derecha está volteada tanto horizontal como verticalmente. Así, si hay cuatro vistas, entonces una imagen de cada vista se dispone en un superfotograma como una pieza de mosaico. Esto resulta en una única secuencia de entrada no codificada con gran resolución.

40 Como alternativa, se puede submuestrear la ilustración para producir una menor resolución. Así, se crean múltiples secuencias que incluyen cada una diferentes vistas que se organizan en mosaico juntas. Cada una de tal secuencia forma entonces una pseudovista, donde cada pseudovista incluye N vistas diferentes organizadas en mosaico. La FIG. 1 muestra una pseudovista, y la FIG. 2 muestra otra pseudovista. Estas pseudovistas se pueden codificar entonces usando estándares existentes de codificación de vídeo tales como el Estándar ISO/IEC MPEG-2 y el 45 Estándar MPEG-4 AVC.

Incluso otro planteamiento para predicción multivista simplemente implica codificar las diferentes vistas independientemente usando un nuevo estándar y, tras decodificar, organizar en mosaico las vistas según requiera el reproductor.

50 Además, en otro planteamiento, las vistas también se pueden organizar en mosaico por píxeles. Por ejemplo, en una supervista que se compone de cuatro vistas, píxel (x, y) puede ser de la vista 0, mientras que el píxel (x+1, y) puede ser de la vista 1, el píxel (x, y+1) puede ser de la vista 2, y el píxel (x+1, y+1) puede ser de la vista 3.

55 Muchos fabricantes de pantallas usan este tipo de trabajo de fotogramas para disponer u organizar en mosaico diferentes vistas en un único fotograma y luego extraer las vistas de sus respectivas ubicaciones y representarlas. En tales casos, no hay manera estándar para determinar si el flujo de bits tiene este tipo de propiedad. Así, si un sistema usa el método de organizar en mosaico imágenes de diferentes vistas en un fotograma grande, entonces el método de extraer las diferentes vistas es en propiedad.

Sin embargo, no hay manera estándar para determinar si el flujo de bits tiene este tipo de propiedad. Se propone sintaxis de alto nivel a fin de facilitar al solicitante o al reproductor extraer tal información a fin de ayudar a exponer u otro procesamiento posterior. También es posible que la subimágenes tengan diferentes resoluciones y puede ser necesario algo de supermuestreo para finalmente representar la vista. El usuario también puede querer tener el 5 método de supermuestreo indicado en la sintaxis de alto nivel. Adicionalmente, también se pueden trasmitir parámetros para cambiar el enfoque de profundidad.

En una realización se propone un nuevo mensaje de Información de Mejora Suplementaria (SEI, del inglés *Supplemental Enhancement Information*) para señalización de información multivista en un flujo de bits compatible con Estándar MPEG-4 AVC donde cada imagen incluye subimágenes que pertenecen a una vista diferente. La 10 realización está pensada, por ejemplo, para la exposición fácil y conveniente de flujos de vídeo multivista en monitores tridimensionales (3D) que pueden usar este tipo de estructura. El concepto se puede extender a otros estándares de codificación de vídeo y recomendaciones que señalan tal información usando sintaxis de alto nivel.

Además, en una realización se propone un método de señalización de cómo disponer vistas antes de que sean 15 enviadas al codificador y/o decodificador de vídeo multivista. Ventajosamente, la realización puede llevar a una implementación simplificada de la codificación multivista, y puede beneficiar la eficiencia de codificación. Ciertas vistas se pueden poner juntas y formar una pseudovista o supervista y luego la supervista organizada en mosaico es tratada como vista normal por un codificador y/o decodificador común de vídeo multivista, por ejemplo, según la 20 implementación basada en Estándar MPEG-4 AVC actual de codificación de vídeo multivista. Se propone una nueva bandera en la extensión de conjunto de parámetros de secuencia (SPS) de codificación de vídeo multivista para señalar el uso de la técnica de pseudovistas. La realización está pensada para la exposición fácil y conveniente de flujos de vídeo multivista en monitores 3D que pueden usar este tipo de estructura.

Codificar/decodificar usando un estándar/recomendación de codificación/decodificación de vídeo de única vista

En la implementación actual de codificación multivídeo (MVC) basada en la Organización Internacional para Estandarización / Comisión Electrotécnica Internacional (ISO/IEC), estándar Moving Picture Experts Group-4 (MPEG-4) Codificación de Vídeo Avanzada (AVC) Parte 10 / Unión Internacional de Telecomunicaciones, Recomendación H.264 del Sector de Telecomunicaciones (ITU-T) (más adelante en esta memoria la "Estándar 25 MPEG-4 AVC"), el software de referencia logra predicción multivista al codificar cada vista con un único codificador y teniendo en consideración las referencias de vista en cruz. Cada vista es codificada por el codificador como flujo de bits separado en su resolución original y más tarde todos los flujos de bits se combinan para formar un único flujo de bits que luego se decodifica. Cada vista produce una salida decodificada YUV separada.

Otro planteamiento para la predicción multivista implica organizar en mosaico las imágenes de cada vista (muestreadas al mismo tiempo) en un fotograma más grande o un superfotograma con una posible operación de submuestreo. Cambiando a la FIG. 1, un ejemplo de cuatro vistas organizadas en mosaico en un único fotograma se indica generalmente con el numeral de referencia 100. Cambiando a la FIG. 2, un ejemplo de cuatro vistas volteadas 35 y organizadas en mosaico en un único fotograma se indica generalmente con el numeral de referencia 200. Así, si hay cuatro vistas, entonces una imagen de cada vista se dispone en un superfotograma como una pieza de mosaico. Esto resulta en una única secuencia de entrada no codificada con gran resolución. Esta señal se puede codificar entonces usando estándares existentes de codificación de vídeo tales como el Estándar ISO/IEC MPEG-2 y el Estándar MPEG-4 AVC.

40 Incluso otro planteamiento para predicción multivista simplemente implica codificar las diferentes vistas independientemente usando un nuevo estándar y, tras decodificar, organizar en mosaico las vistas según requiera el reproductor.

Muchos fabricantes de pantallas usan este tipo de trabajo de fotogramas para disponer u organizar en mosaico 45 diferentes vistas en un único fotograma y luego extraer las vistas de sus respectivas ubicaciones y representarlas. En tales casos, no hay manera estándar para determinar si el flujo de bits tiene este tipo de propiedad. Así, si un sistema usa el método de organizar en mosaico imágenes de diferentes vistas en un fotograma grande, entonces el método de extraer las diferentes vistas es en propiedad.

Cambiando a la FIG. 3, un codificador de vídeo capaz de realizar codificación de vídeo según el Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 300.

50 El codificador de vídeo 300 incluye un almacenamiento intermedio (búfer o *buffer*) de ordenación de fotogramas 310 que tiene una salida en comunicación de señal con una entrada no inversora de un combinador 385. Una salida del combinador 385 se conecta en comunicación de señal con una primera entrada de un transformador y cuantificador 325. Una salida del transformador y el cuantificador 325 se conecta en comunicación de señal con una primera 55 entrada de un codificador de entropía 345 y una primera entrada de un transformador inverso y cuantificador inverso 350. Una salida del codificador de entropía 345 se conecta en comunicación de señal con una primera entrada no inversora de un combinador 390. Una salida del combinador 390 se conecta en comunicación de señal con una primera entrada de un almacenamiento intermedio de salida 335.

- Una primera salida de un controlador de codificador 305 se conecta en comunicación de señal con una segunda entrada del almacenamiento intermedio de ordenación de fotogramas 310, una segunda entrada del transformador inverso y el cuantificador inverso 350, una entrada de un módulo de decisión de tipo imagen 315, una entrada de un módulo de decisión tipo macrobloque (MB-tipo) 320, una segunda entrada de un módulo intra predicción 360, una 5 segunda entrada de un filtro de desbloqueo 365, una primera entrada de un compensador de movimiento 370, una primera entrada de un estimador de movimiento 375, y una segunda entrada de un almacenamiento intermedio de imagen de referencia 380.
- Una segunda salida del controlador de codificador 305 se conecta en comunicación de señal con una primera 10 entrada de un insertador de Información de Mejora Suplementaria (SEI) 330, una segunda entrada del transformador y cuantificador 325, una segunda entrada del codificador de entropía 345, una segunda entrada del almacenamiento intermedio de salida 335, y una entrada del insertador de conjunto de parámetros de secuencia (SPS) y conjunto de parámetros de imagen (PPS) 340.
- Una primera salida del módulo de decisión de tipo imagen 315 se conecta en comunicación de señal con una tercera 15 entrada de un almacenamiento intermedio de ordenación de fotogramas 310. Una segunda salida del módulo de decisión de tipo imagen 315 se conecta en comunicación de señal con una segunda entrada de un módulo de decisión de tipo macrobloque 320.
- Una salida del insertador de conjunto de parámetros de secuencia (SPS) y conjunto de parámetros de imagen (PPS) 340 se conecta en comunicación de señal con una tercera entrada no inversora del combinador 390. Una salida del insertador SEI 330 se conecta en comunicación de señal con una segunda entrada no inversora del combinador 390. 20
- Una salida del cuantificador inverso y el transformador inverso 350 se conecta en comunicación de señal con una primera entrada no inversora de un combinador 319. Una salida del combinador 319 se conecta en comunicación de señal con una primera entrada del módulo intra predicción 360 y una primera entrada del filtro de desbloqueo 365. Una salida del filtro de desbloqueo 365 se conecta en comunicación de señal con una primera entrada de un 25 almacenamiento intermedio de imagen de referencia 380. Una salida del almacenamiento intermedio de imagen de referencia 380 se conecta en comunicación de señal con una segunda entrada del estimador de movimiento 375 y con una primera entrada de un compensador de movimiento 370. Una primera salida del estimador de movimiento 375 se conecta en comunicación de señal con una segunda entrada del compensador de movimiento 370. Una segunda salida del estimador de movimiento 375 se conecta en comunicación de señal con una tercera entrada del codificador de entropía 345.
- Una salida del compensador de movimiento 370 se conecta en comunicación de señal con una primera entrada de un 30 comutador 397. Una salida del módulo intra predicción 360 se conecta en comunicación de señal con una segunda entrada del comutador 397. Una salida del módulo de decisión de tipo macrobloque 320 se conecta en comunicación de señal con una tercera entrada del comutador 397 a fin de proporcionar una entrada de control al comutador 397. La tercera entrada del comutador 397 determina si la entrada de "datos" del comutador (en 35 comparación con la entrada de control, es decir, la tercera entrada) va a ser proporcionada por el compensador de movimiento 370 o el módulo intra predicción 360. La salida del comutador 397 se conecta en comunicación de señal con una segunda entrada no inversora del combinador 319 y con una entrada inversora del combinador 385.
- Entradas del almacenamiento intermedio de ordenación de fotogramas 310 y el controlador de codificador 105 están 40 disponibles como entrada del codificador 300, para recibir una imagen de entrada 301. Además, una entrada del insertador de Información de Mejora Suplementaria (SEI) 330 está disponible como entrada del codificador 300, para recibir metadatos. Una salida del almacenamiento intermedio de salida 335 está disponible como salida del codificador 300, para tener como salida un flujo de bits.
- Cambiando a la FIG. 4, un decodificador de vídeo capaz de realizar decodificación de vídeo según el Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 400. 45
- El decodificador de vídeo 400 incluye un almacenamiento intermedio de entrada 410 que tiene una salida conectada en comunicación de señal con una primera entrada del decodificador de entropía 445. Una primera salida del decodificador de entropía 445 se conecta en comunicación de señal con una primera entrada de un transformador inverso y el cuantificador inverso 450. Una salida del transformador inverso y el cuantificador inverso 450 se conecta en comunicación de señal con una segunda entrada no inversora de un combinador 425. Una salida del combinador 50 425 se conecta en comunicación de señal con una segunda entrada de un filtro de desbloqueo 465 y una primera entrada de un módulo intra predicción 460. Una segunda salida del filtro de desbloqueo 465 se conecta en comunicación de señal con una primera entrada de un almacenamiento intermedio de imagen de referencia 480. Una salida del almacenamiento intermedio de imagen de referencia 480 se conecta en comunicación de señal con una segunda entrada de un compensador de movimiento 470.
- Una segunda salida del decodificador de entropía 445 se conecta en comunicación de señal con una tercera entrada del compensador de movimiento 470 y una primera entrada del filtro de desbloqueo 465. Una tercera salida del decodificador de entropía 445 se conecta en comunicación de señal con una entrada de un controlador de decodificador 405. Una primera salida del controlador de decodificador 405 se conecta en comunicación de señal 55

con una segunda entrada del decodificador de entropía 445. Una segunda salida del controlador de decodificador 405 se conecta en comunicación de señal con una segunda entrada del transformador inverso y el cuantificador inverso 450. Una tercera salida del controlador de decodificador 405 se conecta en comunicación de señal con una tercera entrada del filtro de desbloqueo 465. Una cuarta salida del controlador de decodificador 405 se conecta en comunicación de señal con una segunda entrada del módulo intra predicción 460, con una primera entrada del compensador de movimiento 470, y con una segunda entrada del almacenamiento intermedio de imagen de referencia 480.

Una salida del compensador de movimiento 470 se conecta en comunicación de señal con una primera entrada de un conmutador 497. Una salida del módulo intra predicción 460 se conecta en comunicación de señal con una segunda entrada del conmutador 497. Una salida del conmutador 497 se conecta en comunicación de señal con una primera entrada no inversora del combinador 425.

Una entrada del almacenamiento intermedio de entrada 410 está disponible como entrada del decodificador 400, para recibir un flujo de bits de entrada. Una primera salida del filtro de desbloqueo 465 está disponible como salida del decodificador 400, para tener como salida una imagen de salida.

15 Cambiando a la FIG. 5, un método ejemplar para codificar imágenes para una pluralidad de vistas usando el Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 500.

El método 500 incluye un bloque de inicio 502 que pasa el control a un bloque de función 504. El bloque de función 504 dispone cada vista en un instante de tiempo particular como subimagen en formato mosaico, y pasa el control a un bloque de función 506. El bloque de función 506 establece un elemento de sintaxis num_coded_views_minus1, y pasa el control a un bloque de función 508. El bloque de función 508 establece elementos de sintaxis org_pic_width_in_mbs_minus1 y org_pic_height_in_mbs_minus1, y pasa el control a un bloque de función 510. El bloque de función 510 establece una variable i igual a cero, y pasa el control a un bloque de decisión 512. El bloque de decisión 512 determina si la variable i es menor que el número de vistas. Si es así, entonces se pasa el control a un bloque de función 514. De otro modo, se pasa el control a un bloque de función 524.

25 El bloque de función 514 establece un elemento de sintaxis view_id[i], y pasa el control a un bloque de función 516. El bloque de función 516 establece un elemento de sintaxis num_parts[view_id[i]], y pasa el control a un bloque de función 518. El bloque de función 518 establece una variable j igual a cero, y pasa el control a un bloque de decisión 520. El bloque de decisión 520 determina si el valor actual de la variable j es menor que el valor actual del elemento de sintaxis num_parts[view_id[i]]. Si es así, entonces se pasa el control a un bloque de función 522. De otro modo, 30 se pasa el control a un bloque de función 528.

El bloque de función 522 establece los siguientes elementos de sintaxis, incrementa la variable j, y entonces devuelve control al bloque de decisión 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]; y

35 frame_crop_bottom_offset[view_id[i]][j].

El bloque de función 528 establece un elemento de sintaxis upsample_view_flag[view_id[i]], y pasa el control a un bloque de decisión 530. El bloque de decisión 530 determina si el valor actual del elemento de sintaxis upsample_view_flag[view_id[i]] es igual a uno. Si es así, entonces se pasa el control a un bloque de función 532. De otro modo, se pasa el control a un bloque de decisión 534.

40 El bloque de función 532 establece un elemento de sintaxis upsample_filter[view_id[i]], y pasa el control al bloque de decisión 534.

El bloque de decisión 534 determina si el valor actual del elemento de sintaxis upsample_filter[view_id[i]] es igual a tres. Si es así, entonces se pasa el control a un bloque de función 536. De otro modo, se pasa el control a un bloque de función 540.

45 El bloque de función 536 establece los siguientes elementos de sintaxis y pasa el control a un bloque de función 538: vert_dim[view_id[i]]; hor_dim[view_id[i]]; y quantizer[view_id[i]].

El bloque de función 538 establece los coeficientes de filtro para cada componente YUV, y pasa el control al bloque de función 540.

El bloque de función 540 incrementa la variable i, y devuelve control al bloque de decisión 512.

50 El bloque de función 524 escribe estos elementos de sintaxis en al menos uno del conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS), el mensaje de Información de Mejora Suplementaria (SEI), el encabezado de unidad de Capa de Abstracción de Rede (NAL), y el encabezado de segmento, y pasa el control a un bloque de función 526.

El bloque de función 526 codifica cada imagen usando el Estándar MPEG-4 AVC u otro códec de única vista, y pasa el control a un bloque final 599.

Cambiando a la FIG. 6, un método ejemplar para decodificar imágenes para una pluralidad de vistas usando el Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 600.

- 5 El método 600 incluye un bloque de inicio 602 que pasa el control a un bloque de función 604. El bloque de función 604 analiza sintácticamente los siguientes elementos de sintaxis de al menos uno del conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS), el mensaje de Información de Mejora Suplementaria (SEI), el encabezado de unidad de Capa de Abstracción de Red (NAL), y el encabezado de segmento, y pasa el control a un bloque de función 606. El bloque de función 606 analiza sintácticamente un elemento de sintaxis num_coded_views_minus1, y pasa el control a un bloque de función 608. El bloque de función 608 analiza sintácticamente elementos de sintaxis org_pic_width_in_mbs_minus1 y org_pic_height_in_mbs_minus1, y pasa el control a un bloque de función 610. El bloque de función 610 establece una variable i igual a cero, y pasa el control a un bloque de decisión 612. El bloque de decisión 612 determina si la variable i es menor que el número de vistas. Si es así, entonces se pasa el control a un bloque de función 614. De otro modo, se pasa el control a un bloque de función 624.

El bloque de función 614 analiza sintácticamente un elemento de sintaxis view_id[i], y pasa el control a un bloque de función 616. El bloque de función 616 analiza sintácticamente un elemento de sintaxis num_parts_minus1[view_id[i]], y pasa el control a un bloque de función 618. El bloque de función 618 establece una variable j igual a cero, y pasa el control a un bloque de decisión 620.

- 20 El bloque de decisión 620 determina si el valor actual de la variable j es menor que el valor actual del elemento de sintaxis num_parts[view_id[i]]. Si es así, entonces se pasa el control a un bloque de función 622. De otro modo, se pasa el control a un bloque de función 628.

El bloque de función 622 analiza sintácticamente los siguientes elementos de sintaxis, incrementa la variable j, y entonces devuelve control al bloque de decisión 620: depth_flag[view_id[i]][j]; flip_dir[view_id[i]][j];

- 25 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]; y
frame_crop_bottom_offset[view_id[i]][j].

- 30 El bloque de función 628 analiza sintácticamente un elemento de sintaxis upsample_view_flag[view_id[i]], y pasa el control a un bloque de decisión 630. El bloque de decisión 630 determina si el valor actual del elemento de sintaxis upsample_view_flag[view_id[i]] es igual a uno. Si es así, entonces se pasa el control a un bloque de función 632. De otro modo, se pasa el control a un bloque de decisión 634.

El bloque de función 632 analiza sintácticamente un elemento de sintaxis upsample_filter[view_id[i]], y pasa el control al bloque de decisión 634.

- 35 El bloque de decisión 634 determina si el valor actual del elemento de sintaxis upsample_filter[view_id[i]] es igual a tres. Si es así, entonces se pasa el control a un bloque de función 636. De otro modo, se pasa el control a un bloque de función 640.

El bloque de función 636 analiza sintácticamente los siguientes elementos de sintaxis y pasa el control a un bloque de función 638: vert_dim[view_id[i]]; hor_dim[view_id[i]]; y quantizer[view_id[i]].

- 40 El bloque de función 638 analiza sintácticamente los coeficientes de filtro para cada componente YUV, y pasa el control al bloque de función 640.

El bloque de función 640 incrementa la variable i, y devuelve control al bloque de decisión 612.

El bloque de función 624 decodifica cada imagen usando el Estándar MPEG-4 AVC u otro códec de única vista, y pasa el control a un bloque de función 626. El bloque de función 626 separa cada vista de la imagen usando la sintaxis de alto nivel, y pasa el control a un bloque final 699.

- 45 Cambiando a la FIG. 7, un método ejemplar para codificar imágenes para una pluralidad de vistas y profundidades usando el Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 700.

El método 700 incluye un bloque de inicio 702 que pasa el control a un bloque de función 704. El bloque de función 704 dispone cada vista y correspondiente profundidad en un instante de tiempo particular como subimagen en formato mosaico, y pasa el control a un bloque de función 706. El bloque de función 706 establece un elemento de sintaxis num_coded_views_minus1, y pasa el control a un bloque de función 708. El bloque de función 708 establece elementos de sintaxis org_pic_width_in_mbs_minus1 y org_pic_height_in_mbs_minus1, y pasa el control a un

bloque de función 710. El bloque de función 710 establece una variable *i* igual a cero, y pasa el control a un bloque de decisión 712.

El bloque de decisión 712 determina si la variable *i* es menor que el número de vistas. Si es así, entonces se pasa el control a un bloque de función 714. De otro modo, se pasa el control a un bloque de función 724.

- 5 El bloque de función 714 establece un elemento de sintaxis *view_id[i]*, y pasa el control a un bloque de función 716. El bloque de función 716 establece un elemento de sintaxis *num_parts[view_id[i]]*, y pasa el control a un bloque de función 718. El bloque de función 718 establece una variable *j* igual a cero, y pasa el control a un bloque de decisión 720. El bloque de decisión 720 determina si el valor actual de la variable *j* es menor que el valor actual del elemento de sintaxis *num_parts[view_id[i]]*. Si es así, entonces se pasa el control a un bloque de función 722. De otro modo,
10 se pasa el control a un bloque de función 728.

El bloque de función 722 establece los siguientes elementos de sintaxis, incrementa la variable *j*, y entonces devuelve control al bloque de decisión 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]*; y

- 15 *frame_crop_bottom_offset[view_id[i]][j]*.

El bloque de función 728 establece un elemento de sintaxis *upsample_view_flag[view_id[i]]*, y pasa el control a un bloque de decisión 730. El bloque de decisión 730 determina si el valor actual del elemento de sintaxis *upsample_view_flag[view_id[i]]* es igual a uno. Si es así, entonces se pasa el control a un bloque de función 732. De otro modo, se pasa el control a un bloque de decisión 734.

- 20 El bloque de función 732 establece un elemento de sintaxis *upsample_filter[view_id[i]]*, y pasa el control al bloque de decisión 734.

El bloque de decisión 734 determina si el valor actual del elemento de sintaxis *upsample_filter[view_id[i]]* es igual a tres. Si es así, entonces se pasa el control a un bloque de función 736. De otro modo, se pasa el control a un bloque de función 740.

- 25 El bloque de función 736 establece los siguientes elementos de sintaxis y pasa el control a un bloque de función 738: *vert_dim[view_id[i]]*; *hor_dim[view_id[i]]*; y *quantizer[view_id[i]]*.

El bloque de función 738 establece los coeficientes de filtro para cada componente YUV, y pasa el control al bloque de función 740.

El bloque de función 740 incrementa la variable *i*, y devuelve control al bloque de decisión 712.

- 30 El bloque de función 724 escribe estos elementos de sintaxis en al menos uno del conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS), el mensaje de Información de Mejora Suplementaria (SEI), el encabezado de unidad de Capa de Abstracción de Rede (NAL), y el encabezado de segmento, y pasa el control a un bloque de función 726.

- 35 El bloque de función 726 codifica cada imagen usando el Estándar MPEG-4 AVC u otro códec de única vista, y pasa el control a un bloque final 799.

Cambiando a la FIG. 8, un método ejemplar para decodificar imágenes para una pluralidad de vistas y profundidades usando el Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 800.

- 40 El método 800 incluye un bloque de inicio 802 que pasa el control a un bloque de función 804. El bloque de función 804 analiza sintácticamente los siguientes elementos de sintaxis de al menos uno del conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS), el mensaje de Información de Mejora Suplementaria (SEI), el encabezado de unidad de Capa de Abstracción de Red (NAL), y el encabezado de segmento, y pasa el control a un bloque de función 806. El bloque de función 806 analiza sintácticamente un elemento de sintaxis *num_coded_views_minus1*, y pasa el control a un bloque de función 808. El bloque de función 808 analiza sintácticamente elementos de sintaxis *org_pic_width_in_mbs_minus1* y *org_pic_height_in_mbs_minus1*, y pasa el control a un bloque de función 810. El bloque de función 810 establece una variable *i* igual a cero, y pasa el control a un bloque de decisión 812. El bloque de decisión 812 determina si la variable *i* es menor que el número de vistas. Si es así, entonces se pasa el control a un bloque de función 814. De otro modo, se pasa el control a un bloque de función 824.

- 45 El bloque de función 814 analiza sintácticamente un elemento de sintaxis *view_id[i]*, y pasa el control a un bloque de función 816. El bloque de función 816 analiza sintácticamente un elemento de sintaxis *num_parts_minus1[view_id[i]]*, y pasa el control a un bloque de función 818. El bloque de función 818 establece una variable *j* igual a cero, y pasa el control a un bloque de decisión 820. El bloque de decisión 820 determina si el valor actual de la variable *j* es menor

que el valor actual del elemento de sintaxis `num_parts[view_id[i]]`. Si es así, entonces se pasa el control a un bloque de función 822. De otro modo, se pasa el control a un bloque de función 828.

El bloque de función 822 analiza sintácticamente los siguientes elementos de sintaxis, incrementa la variable `j`, y entonces devuelve control al bloque de decisión 820: `depth_flag[view_id[i]][j]`; `flip_dir[view_id[i]][j]`;

- 5 `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]`; `y`
`frame_crop_bottom_offset[view_id[i]][j]`.

El bloque de función 828 analiza sintácticamente un elemento de sintaxis `upsample_view_flag[view_id[i]]`, y pasa el control a un bloque de decisión 830. El bloque de decisión 830 determina si el valor actual del elemento de sintaxis `upsample_view_flag[view_id[i]]` es igual a uno. Si es así, entonces se pasa el control a un bloque de función 832. De otro modo, se pasa el control a un bloque de decisión 834.

El bloque de función 832 analiza sintácticamente un elemento de sintaxis `upsample_filter[view_id[i]]`, y pasa el control al bloque de decisión 834.

El bloque de decisión 834 determina si el valor actual del elemento de sintaxis `upsample_filter[view_id[i]]` es igual a tres. Si es así, entonces se pasa el control a un bloque de función 836. De otro modo, se pasa el control a un bloque de función 840.

El bloque de función 836 analiza sintácticamente los siguientes elementos de sintaxis y pasa el control a un bloque de función 838: `vert_dim[view_id[i]]`; `hor_dim[view_id[i]]`; `y quantizer[view_id[i]]`.

- 20 El bloque de función 838 analiza sintácticamente los coeficientes de filtro para cada componente YUV, y pasa el control al bloque de función 840.

El bloque de función 840 incrementa la variable `i`, y devuelve control al bloque de decisión 812.

El bloque de función 824 decodifica cada imagen usando el Estándar MPEG-4 AVC u otro códec de única vista, y pasa el control a un bloque de función 826. El bloque de función 826 separa cada vista y correspondiente profundidad de la imagen usando la sintaxis de alto nivel, y pasa el control a un bloque de función 827. El bloque de función 827 potencialmente realiza síntesis de vista usando la vista extraída y señales de profundidad, y pasa el control a un bloque final 899.

Con respecto a la profundidad usada en las FIG. 7 y 8, la FIG. 9 muestra un ejemplo de una señal de profundidad 900, donde profundidad se proporciona como valor de píxel para cada correspondiente ubicación de una ilustración (no se muestra). Además, la FIG. 10 muestra un ejemplo de dos señales de profundidad incluidas en una pieza de mosaico 1000. La zona superior-derecha de la pieza de mosaico 1000 es una señal de profundidad que tiene valores de profundidad correspondientes a la ilustración en la zona superior-izquierda de la pieza de mosaico 1000. La zona inferior-derecha de la pieza de mosaico 1000 es una señal de profundidad que tiene valores de profundidad correspondientes a la ilustración en la zona inferior-izquierda de la pieza de mosaico 1000.

35 Cambiando a la FIG. 11, un ejemplo de 5 vistas organizadas en mosaico en un único fotograma se indica generalmente con el numeral de referencia 1100. Las cuatro vistas superiores están en una orientación normal. La quinta vista también está en una orientación normal, pero está partida en dos zonas a lo largo de la zona inferior de la pieza de mosaico 1100. Una zona izquierda de la quinta vista muestra la "zona superior" de la quinta vista, y una zona derecha de la quinta vista muestra la "zona inferior" de la quinta vista.

Codificar/decodificar usando un estándar/recomendación de codificación/decodificación de vídeo multivista

- 40 Cambiando a la FIG. 12, un codificador ejemplar de Codificación de vídeo multivista (MVC) se indica generalmente con el numeral de referencia 1200. El codificador 1200 incluye un combinador 1205 que tiene una salida conectada en comunicación de señal con una entrada de un transformador 1210. Una salida del transformador 1210 se conecta en comunicación de señal con una entrada de cuantificador 1215. Una salida del cuantificador 1215 se conecta en comunicación de señal con una entrada de un codificador de entropía 1220 y una entrada de un cuantificador inverso 1225. Una salida del cuantificador inverso 1225 se conecta en comunicación de señal con una entrada de un transformador inverso 1230. Una salida del transformador inverso 1230 se conecta en comunicación de señal con una primera entrada no inversora de un combinador 1235. Una salida del combinador 1235 se conecta en comunicación de señal con una entrada de un intra predictor 1245 y una entrada de un filtro de desbloqueo 1250. Una salida del filtro de desbloqueo 1250 se conecta en comunicación de señal con una entrada de un almacén de imágenes de referencia 1255 (para la vista `i`). Una salida del almacén de imágenes de referencia 1255 se conecta en comunicación de señal con una primera entrada de un compensador de movimiento 1275 y una primera entrada de un estimador de movimiento 1280. Una salida del estimador de movimiento 1280 se conecta en comunicación de señal con una segunda entrada del compensador de movimiento 1275.

Una salida de un almacén de imágenes de referencia 1260 (para otras vistas) se conecta en comunicación de señal con una primera entrada de un estimador de disparidad 1270 y una primera entrada de un compensador de disparidad 1265. Una salida del estimador de disparidad 1270 se conecta en comunicación de señal con una segunda entrada del compensador de disparidad 1265.

- 5 Una salida del decodificador de entropía 1220 está disponible como salida del codificador 1200. Una entrada no inversora del combinador 1205 está disponible como entrada del codificador 1200, y se conecta en comunicación de señal con una segunda entrada del estimador de disparidad 1270, y una segunda entrada del estimador de movimiento 1280. Una salida de un conmutador 1285 se conecta en comunicación de señal con una segunda entrada no inversora del combinador 1235 y con una entrada inversora del combinador 1205. El conmutador 1285 incluye una primera entrada conectada en comunicación de señal con una salida del compensador de movimiento 1275, una segunda entrada conectada en comunicación de señal con una salida del compensador de disparidad 1265, y una tercera entrada conectada en comunicación de señal con una salida del intra predictor 1245.

Un módulo de decisión de modo 1240 tiene una salida conectada al conmutador 1285 para controlar qué entrada es seleccionada por el conmutador 1285.

- 15 Cambiando a la FIG. 13, un decodificador exemplar de Codificación de vídeo multivista (MVC) se indica generalmente con el numeral de referencia 1300. El decodificador 1300 incluye un decodificador de entropía 1305 que tiene una salida conectada en comunicación de señal con una entrada de un cuantificador inverso 1310. Una salida del cuantificador inverso se conecta en comunicación de señal con una entrada de un transformador inverso 1315. Una salida del transformador inverso 1315 se conecta en comunicación de señal con una primera entrada no inversora de un combinador 1320. Una salida del combinador 1320 se conecta en comunicación de señal con una entrada de un filtro de desbloqueo 1325 y una entrada de un intra predictor 1330. Una salida del filtro de desbloqueo 1325 se conecta en comunicación de señal con una entrada de un almacén de imágenes de referencia 1340 (para la vista i). Una salida del almacén de imágenes de referencia 1340 se conecta en comunicación de señal con una primera entrada de un compensador de movimiento 1335.

- 25 Una salida de un almacén de imágenes de referencia 1345 (para otras vistas) se conecta en comunicación de señal con una primera entrada de un compensador de disparidad 1350.

Una entrada del codificador de entropía 1305 está disponible como entrada al decodificador 1300, para recibir un flujo de bits residual. Además, una entrada de un módulo de modo 1360 también está disponible como entrada al decodificador 1300, para recibir sintaxis de control para controlar cuya entrada es seleccionada por el conmutador 30 1355. Además, una segunda entrada del compensador de movimiento 1335 está disponible como entrada del decodificador 1300, para recibir vectores de movimiento. También, una segunda entrada del compensador de disparidad 1350 está disponible como entrada al decodificador 1300, para recibir vectores de disparidad.

- 35 Una salida de un conmutador 1355 se conecta en comunicación de señal con una segunda entrada no inversora del combinador 1320. Una primera entrada del conmutador 1355 se conecta en comunicación de señal con una salida del compensador de disparidad 1350. Una segunda entrada del conmutador 1355 se conecta en comunicación de señal con una salida del compensador de movimiento 1335. Una tercera entrada del conmutador 1355 se conecta en comunicación de señal con una salida del intra predictor 1330. Una salida del módulo de modo 1360 se conecta en comunicación de señal con el conmutador 1355 para controlar qué entrada es seleccionada por el conmutador 1355. Una salida del filtro de desbloqueo 1325 está disponible como salida del decodificador 1300.

- 40 Cambiando a la FIG. 14, un método exemplar para procesar imágenes para una pluralidad de vistas en preparación para codificar las imágenes usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 1400.

El método 1400 incluye un bloque de inicio 1405 que pasa el control a un bloque de función 1410. El bloque de función 1410 dispone cada N vistas, entre un total de M vistas, en un instante de tiempo particular como superimagen en formato mosaico, y pasa el control a un bloque de función 1415. El bloque de función 1415 establece un elemento de sintaxis num_coded_views_minus1, y pasa el control a un bloque de función 1420. El bloque de función 1420 establece un elemento de sintaxis view_id[i] para todas vistas (num_coded_views_minus1 + 1), y pasa el control a un bloque de función 1425. El bloque de función 1425 establece la información de dependencia de referencia intervista para imágenes de anclaje, y pasa el control a un bloque de función 1430. El bloque de función 1430 establece la información de dependencia de referencia intervista para imágenes no de anclaje, y pasa el control a un bloque de función 1435. El bloque de función 1435 establece un elemento de sintaxis pseudo_view_present_flag, y pasa el control a un bloque de decisión 1440. El bloque de decisión 1440 determina si el valor actual del elemento de sintaxis pseudo_view_present_flag es igual a verdadero. Si es así, entonces se pasa el control a un bloque de función 1445. De otro modo, se pasa el control a un bloque final 1499.

- 55 El bloque de función 1445 establece los siguientes elementos de sintaxis, y pasa el control a un bloque de función 1450: tiling_mode; org_pic_width_in_mbs_minus1; y org_pic_height_in_mbs_minus1. El bloque de función 1450 llama a un elemento de sintaxis pseudo_view_info(view_id) para cada vista codificada, y pasa el control al bloque final 1499.

Cambiando a la FIG. 15, un método ejemplar para codificar imágenes para una pluralidad de vistas usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 1500.

- 5 El método 1500 incluye un bloque de inicio 1502 que tiene un parámetro de entrada `pseudo_view_id` y pasa el control a un bloque de función 1504. El bloque de función 1504 establece un elemento de sintaxis `num_sub_views_minus1`, y pasa el control a un bloque de función 1506. El bloque de función 1506 establece una variable `i` igual a cero, y pasa el control a un bloque de decisión 1508. El bloque de decisión 1508 determina si la variable `i` es menor que el número de `sub_views`. Si es así, entonces se pasa el control a un bloque de función 1510. De otro modo, se pasa el control a un bloque de función 1520.
- 10 El bloque de función 1510 establece un elemento de sintaxis `sub_view_id[i]`, y pasa el control a un bloque de función 1512. El bloque de función 1512 establece un elemento de sintaxis `num_parts_minus1 [sub_view_id[i]]`, y pasa el control a un bloque de función 1514. El bloque de función 1514 establece una variable `j` igual a cero, y pasa el control a un bloque de decisión 1516. El bloque de decisión 1516 determina si la variable `j` es menor que el elemento de sintaxis `num_parts_minus1 [sub_view_id[i]]`. Si es así, entonces se pasa el control a un bloque de función 1518. De otro modo, se pasa el control a un bloque de decisión 1522.
- 15 El bloque de función 1518 establece los siguientes elementos de sintaxis, incrementa la variable `j`, y devuelve el control al bloque de decisión 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]`; y `frame_crop_bottom_offset[sub_view_id[i]][j]`.
- 20 El bloque de función 1520 codifica la imagen actual para la vista actual usando codificación de vídeo multivista (MVC), y pasa el control a un bloque final 1599.
- El bloque de decisión 1522 determina si un elemento de sintaxis `tiling_mode` es igual a cero. Si es así, entonces se pasa el control a un bloque de función 1524. De otro modo, se pasa el control a un bloque de función 1538.
- 25 El bloque de función 1524 establece un elemento de sintaxis `flip_dir[sub_view_id[i]]` y un elemento de sintaxis `upsample_view_flag[sub_view_id[i]]`, y pasa el control a un bloque de decisión 1526. El bloque de decisión 1526 determina si el valor actual del elemento de sintaxis `upsample_view_flag[sub_view_id[i]]` es igual a uno. Si es así, entonces se pasa el control a un bloque de función 1528. De otro modo, se pasa el control a un bloque de decisión 1530.
- 30 El bloque de función 1528 establece un elemento de sintaxis `upsample_filter[sub_view_id[i]]`, y pasa el control al bloque de decisión 1530. El bloque de decisión 1530 determina si un valor del elemento de sintaxis `upsample_filter[sub_view_id[i]]` es igual a tres. Si es así, el control se pasa a un bloque de función 1532. De otro modo, se pasa el control a un bloque de función 1536.
- 35 El bloque de función 1532 establece los siguientes elementos de sintaxis, y pasa el control a un bloque de función 1534: `vert_dim[sub_view_id[i]]`; `hor_dim[sub_view_id[i]]`; y `quantizer[sub_view_id[i]]`. El bloque de función 1534 establece los coeficientes de filtro para cada componente YUV, y pasa el control al bloque de función 1536.
- 40 El bloque de función 1536 incrementa la variable `i`, y devuelve control al bloque de decisión 1508.
- 45 El bloque de función 1538 establece un elemento de sintaxis `pixel_dist_x[sub_view_id[i]]` y el elemento de sintaxis `flip_dist_y[sub_view_id[i]]`, y pasa el control a un bloque de función 1540. El bloque de función 1540 establece la variable `j` igual a cero, y pasa el control a un bloque de decisión 1542. El bloque de decisión 1542 determina si el valor actual de la variable `j` es menor que el valor actual del elemento de sintaxis `num_parts[sub_view_id[i]]`. Si es así, entonces se pasa el control a un bloque de función 1544. De otro modo, se pasa el control al bloque de función 1536.
- El bloque de función 1544 establece un elemento de sintaxis `num_pixel_tiling_filter_coeffs_minus1 [sub_view_id[i]]`, y pasa el control a un bloque de función 1546. El bloque de función 1546 establece los coeficientes para todos los filtros de organización en mosaico por píxeles, y pasa el control al bloque de función 1536.
- Cambiando a la FIG. 16, un método ejemplar para procesar imágenes para una pluralidad de vistas en preparación para decodificar las imágenes usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 1600.
- 50 El método 1600 incluye un bloque de inicio 1605 que pasa el control a un bloque de función 1615. El bloque de función 1615 analiza sintácticamente un elemento de sintaxis `num_coded_views_minus1`, y pasa el control a un bloque de función 1620. El bloque de función 1620 analiza sintácticamente un elemento de sintaxis `view_id[i]` para todas vistas (`num_coded_views_minus1 + 1`), y pasa el control a un bloque de función 1625. El bloque de función 1625 analiza sintácticamente la información de dependencia de referencia intervista para imágenes de anclaje, y pasa el control a un bloque de función 1630. El bloque de función 1630 analiza sintácticamente la información de

dependencia de referencia intervista para imágenes no de anclaje, y pasa el control a un bloque de función 1635. El bloque de función 1635 analiza sintácticamente un elemento de sintaxis `pseudo_view_present_flag`, y pasa el control a un bloque de decisión 1640. El bloque de decisión 1640 determina si el valor actual del elemento de sintaxis `pseudo_view_present_flag` es igual a verdadero. Si es así, entonces se pasa el control a un bloque de función 1645.

5 De otro modo, se pasa el control a un bloque final 1699.

El bloque de función 1645 analiza sintácticamente los siguientes elementos de sintaxis, y pasa el control a un bloque de función 1650: `tiling_mode`; `org_pic_width_in_mbs_minus1`; y `org_pic_height_in_mbs_minus1`. El bloque de función 1650 llama a un elemento de sintaxis `pseudo_view_info(view_id)` para cada vista codificada, y pasa el control al bloque final 1699.

10 Cambiando a la FIG. 17, un método ejemplar para decodificar imágenes para una pluralidad de vistas usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 1700.

15 El método 1700 incluye un bloque de inicio 1702 que empieza con parámetro de entrada `pseudo_view_id` y pasa el control a un bloque de función 1704. El bloque de función 1704 analiza sintácticamente un elemento de sintaxis `num_sub_views_minus1`, y pasa el control a un bloque de función 1706. El bloque de función 1706 establece una variable `i` igual a cero, y pasa el control a un bloque de decisión 1708. El bloque de decisión 1708 determina si la variable `i` es menor que el número de `sub_views`. Si es así, entonces se pasa el control a un bloque de función 1710. De otro modo, se pasa el control a un bloque de función 1720.

20 El bloque de función 1710 analiza sintácticamente un elemento de sintaxis `sub_view_id[i]`, y pasa el control a un bloque de función 1712. El bloque de función 1712 analiza sintácticamente un elemento de sintaxis `num_parts_minus1[sub_view_id[i]]`, y pasa el control a un bloque de función 1714.

25 El bloque de función 1714 establece una variable `j` igual a cero, y pasa el control a un bloque de decisión 1716. El bloque de decisión 1716 determina si la variable `j` es menor que el elemento de sintaxis `num_parts_minus1[sub_view_id[i]]`. Si es así, entonces se pasa el control a un bloque de función 1718. De otro modo, se pasa el control a un bloque de decisión 1722.

El bloque de función 1718 establece los siguientes elementos de sintaxis, incrementa la variable `j`, y devuelve el control al bloque de decisión 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]]`; y `frame_crop_bottom_offset[sub_view_id[i][j]]`.

30 El bloque de función 1720 decodifica la imagen actual para la vista actual usando codificación de vídeo multivista (MVC), y pasa el control a un bloque de función 1721. El bloque de función 1721 separa cada vista de la imagen usando la sintaxis de alto nivel, y pasa el control a un bloque final 1799.

35 La separación de cada vista de la imagen decodificada se hace usando la sintaxis de alto nivel indicada en el flujo de bits. Esta sintaxis de alto nivel puede indicar la ubicación exacta y posible orientación de las vistas (y posible correspondiente profundidad) presente en la imagen.

El bloque de decisión 1722 determina si un elemento de sintaxis `tiling_mode` es igual a cero. Si es así, entonces se pasa el control a un bloque de función 1724. De otro modo, se pasa el control a un bloque de función 1738.

40 El bloque de función 1724 analiza sintácticamente un elemento de sintaxis `flip_dir[sub_view_id[i]]` y un elemento de sintaxis `upsample_view_flag[sub_view_id[i]]`, y pasa el control a un bloque de decisión 1726. El bloque de decisión 1726 determina si el valor actual del elemento de sintaxis `upsample_view_flag[sub_view_id[i]]` es igual a uno. Si es así, entonces se pasa el control a un bloque de función 1728. De otro modo, se pasa el control a un bloque de decisión 1730.

45 El bloque de función 1728 analiza sintácticamente un elemento de sintaxis `upsample_filter[sub_view_id[i]]`, y pasa el control al bloque de decisión 1730. El bloque de decisión 1730 determina si un valor del elemento de sintaxis `upsample_filter[sub_view_id[i]]` es igual a tres. Si es así, el control se pasa a un bloque de función 1732. De otro modo, se pasa el control a un bloque de función 1736.

50 El bloque de función 1732 analiza sintácticamente los siguientes elementos de sintaxis, y pasa el control a un bloque de función 1734: `vert_dim[sub_view_id[i]]`; `hor_dim[sub_view_id[i]]`; y `quantizer[sub_view_id[i]]`. El bloque de función 1734 analiza sintácticamente los coeficientes de filtro para cada componente YUV, y pasa el control al bloque de función 1736.

El bloque de función 1736 incrementa la variable `i`, y devuelve control al bloque de decisión 1708.

El bloque de función 1738 analiza sintácticamente un elemento de sintaxis `pixel_dist_x[sub_view_id[i]]` y el elemento de sintaxis `flip_dist_y[sub_view_id[i]]`, y pasa el control a un bloque de función 1740. El bloque de función 1740

establece la variable *j* igual a cero, y pasa el control a un bloque de decisión 1742. El bloque de decisión 1742 determina si el valor actual de la variable *j* es menor que el valor actual del elemento de sintaxis num_parts[sub_view_id[i]]. Si es así, entonces se pasa el control a un bloque de función 1744. De otro modo, se pasa el control al bloque de función 1736.

- 5 El bloque de función 1744 analiza sintácticamente un elemento de sintaxis num_pixel_tiling_filter_coeffs_minus1 [sub_view_id[i]], y pasa el control a un bloque de función 1746. El bloque de función 1776 analiza sintácticamente los coeficientes para todas los filtros de organización en mosaico por píxeles, y pasa el control al bloque de función 1736.

10 Cambiando a la FIG. 18, un método ejemplar para procesar imágenes para una pluralidad de vistas y profundidades en preparación para codificar las imágenes usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 1800.

El método 1800 incluye un bloque de inicio 1805 que pasa el control a un bloque de función 1810. El bloque de función 1810 dispone cada N vistas y mapas de profundidad, entre un total de M vistas y mapas de profundidad, en un instante de tiempo particular como superimagen en formato mosaico, y pasa el control a un bloque de función 1815. El bloque de función 1815 establece un elemento de sintaxis num_coded_views_minus1, y pasa el control a un bloque de función 1820. El bloque de función 1820 establece un elemento de sintaxis view_id[i] para todas profundidades (num_coded_views_minus1 + 1) correspondientes a view_id[i], y pasa el control a un bloque de función 1825. El bloque de función 1825 establece la información de dependencia de referencia entrevista para imágenes de profundidad de anclaje, y pasa el control a un bloque de función 1830. El bloque de función 1830 establece la información de dependencia de referencia entrevista para imágenes de profundidad no de anclaje, y pasa el control a un bloque de función 1835. El bloque de función 1835 establece un elemento de sintaxis pseudo_view_present_flag, y pasa el control a un bloque de decisión 1840. El bloque de decisión 1840 determina si el valor actual del elemento de sintaxis pseudo_view_present_flag es igual a verdadero. Si es así, entonces se pasa el control a un bloque de función 1845. De otro modo, se pasa el control a un bloque final 1899.

- 25 El bloque de función 1845 establece los siguientes elementos de sintaxis, y pasa el control a un bloque de función 1850: tiling_mode; org_pic_width_in_mbs_minus1; y org_pic_height_in_mbs_minus1. El bloque de función 1850 llama a un elemento de sintaxis pseudo_view_info(view_id) para cada vista codificada, y pasa el control al bloque final 1899.

30 Cambiando a la FIG. 19, un método ejemplar para codificar imágenes para una pluralidad de vistas y profundidades usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 1900.

El método 1900 incluye un bloque de inicio 1902 que pasa el control a un bloque de función 1904. El bloque de función 1904 establece un elemento de sintaxis num_sub_views_minus1, y pasa el control a un bloque de función 1906. El bloque de función 1906 establece una variable *i* igual a cero, y pasa el control a un bloque de decisión 1908. El bloque de decisión 1908 determina si la variable *i* es menor que el número de sub_views. Si es así, entonces se pasa el control a un bloque de función 1910. De otro modo, se pasa el control a un bloque de función 1920.

- 40 El bloque de función 1910 establece un elemento de sintaxis sub_view_id[i], y pasa el control a un bloque de función 1912. El bloque de función 1912 establece un elemento de sintaxis num_parts_minus1[sub_view_id[i]], y pasa el control a un bloque de función 1914.

El bloque de función 1914 establece una variable *j* igual a cero, y pasa el control a un bloque de decisión 1916. El bloque de decisión 1916 determina si la variable *j* es menor que el elemento de sintaxis num_parts_minus1 [sub_view_id[i]]. Si es así, entonces se pasa el control a un bloque de función 1918. De otro modo, se pasa el control a un bloque de decisión 1922.

- 45 El bloque de función 1918 establece los siguientes elementos de sintaxis, incrementa la variable *j*, y devuelve el control al bloque de decisión 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]]; y frame_crop_bottom_offset[sub_view_id[i][j]].

50 El bloque de función 1920 codifica la profundidad actual para la vista actual usando codificación de vídeo multivista (MVC), y pasa el control a un bloque final 1999. La señal de profundidad puede ser codificada de manera similar a la manera como se codifica su correspondiente señal de vídeo. Por ejemplo, la señal de profundidad para una vista se puede incluir en una pieza de mosaico que incluye únicamente otras señales de profundidad, o únicamente señales de vídeo, o señales tanto de profundidad como de vídeo. La pieza de mosaico (pseudovista) es tratada entonces como única vista para MVC, y también hay presumiblemente otras piezas de mosaico que se tratan como otras vistas para MVC.

El bloque de decisión 1922 determina si un elemento de sintaxis `tiling_mode` es igual a cero. Si es así, entonces se pasa el control a un bloque de función 1924. De otro modo, se pasa el control a un bloque de función 1938.

5 El bloque de función 1924 establece un elemento de sintaxis `flip_dir[sub_view_id[i]]` y un elemento de sintaxis `upsample_view_flag[sub_view_id[i]]`, y pasa el control a un bloque de decisión 1926. El bloque de decisión 1926 determina si el valor actual del elemento de sintaxis `upsample_view_flag[sub_view_id[i]]` es igual a uno. Si es así, entonces se pasa el control a un bloque de función 1928. De otro modo, se pasa el control a un bloque de decisión 1930.

10 El bloque de función 1928 establece un elemento de sintaxis `upsample_filter[sub_view_id[i]]`, y pasa el control al bloque de decisión 1930. El bloque de decisión 1930 determina si un valor del elemento de sintaxis `upsample_filter[sub_view_id[i]]` es igual a tres. Si es así, el control se pasa a un bloque de función 1932. De otro modo, se pasa el control a un bloque de función 1936.

El bloque de función 1932 establece los siguientes elementos de sintaxis, y pasa el control a un bloque de función 1934: `vert_dim[sub_view_id[i]]`; `hor_dim[sub_view_id[i]]`; y `quantizer[sub_view_id[i]]`. El bloque de función 1934 establece los coeficientes de filtro para cada componente YUV, y pasa el control al bloque de función 1936.

15 El bloque de función 1936 incrementa la variable `i`, y devuelve control al bloque de decisión 1908.

El bloque de función 1938 establece un elemento de sintaxis `pixel_dist_x[sub_view_id[i]]` y el elemento de sintaxis `flip_dist_y[sub_view_id[i]]`, y pasa el control a un bloque de función 1940. El bloque de función 1940 establece la variable `j` igual a cero, y pasa el control a un bloque de decisión 1942. El bloque de decisión 1942 determina si el valor actual de la variable `j` es menor que el valor actual del elemento de sintaxis `num_parts[sub_view_id[i]]`. Si es así, entonces se pasa el control a un bloque de función 1944. De otro modo, se pasa el control al bloque de función 1936.

El bloque de función 1944 establece un elemento de sintaxis `num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]]`, y pasa el control a un bloque de función 1946. El bloque de función 1946 establece los coeficientes para todos los filtros de organización en mosaico por píxeles, y pasa el control al bloque de función 1936.

25 Cambiando a la FIG. 20, un método ejemplar para procesar imágenes para una pluralidad de vistas y profundidades en preparación para decodificar las imágenes usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 2000.

El método 2000 incluye un bloque de inicio 2005 que pasa el control a un bloque de función 2015. El bloque de función 2015 analiza sintácticamente un elemento de sintaxis `num_coded_views_minus1`, y pasa el control a un 30 bloque de función 2020. El bloque de función 2020 analiza sintácticamente un elemento de sintaxis `view_id[i]` para todas profundidades (`num_coded_views_minus1 + 1`) correspondientes a `view_id[i]`, y pasa el control a un bloque de función 2025. El bloque de función 2025 analiza sintácticamente la información de dependencia de referencia intervista para imágenes de profundidad de anclaje, y pasa el control a un bloque de función 2030. El bloque de función 2030 analiza sintácticamente la información de dependencia de referencia intervista para imágenes de profundidad no de anclaje, y pasa el control a un bloque de función 2035. El bloque de función 2035 analiza 35 sintácticamente un elemento de sintaxis `pseudo_view_present_flag`, y pasa el control a un bloque de decisión 2040. El bloque de decisión 2040 determina si el valor actual del elemento de sintaxis `pseudo_view_present_flag` es igual a verdadero. Si es así, entonces se pasa el control a un bloque de función 2045. De otro modo, se pasa el control a un bloque final 2099.

40 El bloque de función 2045 analiza sintácticamente los siguientes elementos de sintaxis, y pasa el control a un bloque de función 2050: `tiling_mode`; `org_pic_width_in_mbs_minus1`; y `org_pic_height_in_mbs_minus1`. El bloque de función 2050 llama a un elemento de sintaxis `pseudo_view_info(view_id)` para cada vista codificada, y pasa el control al bloque final 2099.

45 Cambiando a la FIG. 21, un método ejemplar para decodificar imágenes para una pluralidad de vistas y profundidades usando la extensión de codificación de vídeo multivista (MVC) del Estándar MPEG-4 AVC se indica generalmente con el numeral de referencia 2100.

El método 2100 incluye un bloque de inicio 2102 que empieza con parámetro de entrada `pseudo_view_id` y pasa el control a un bloque de función 2104. El bloque de función 2104 analiza sintácticamente un elemento de sintaxis `num_sub_views_minus1`, y pasa el control a un bloque de función 2106. El bloque de función 2106 establece una 50 variable `i` igual a cero, y pasa el control a un bloque de decisión 2108. El bloque de decisión 2108 determina si la variable `i` es menor que el número de `sub_views`. Si es así, entonces se pasa el control a un bloque de función 2110. De otro modo, se pasa el control a un bloque de función 2120.

55 El bloque de función 2110 analiza sintácticamente un elemento de sintaxis `sub_view_id[i]`, y pasa el control a un bloque de función 2112. El bloque de función 2112 analiza sintácticamente un elemento de sintaxis `num_parts_minus1[sub_view_id[i]]`, y pasa el control a un bloque de función 2114. El bloque de función 2114 establece una variable `j` igual a cero, y pasa el control a un bloque de decisión 2116. El bloque de decisión 2116

determina si la variable j es menor que el elemento de sintaxis num_parts_minus1 [sub_view_id[i]]. Si es así, entonces se pasa el control a un bloque de función 2118. De otro modo, se pasa el control a un bloque de decisión 2122.

El bloque de función 2118 establece los siguientes elementos de sintaxis, incrementa la variable j, y devuelve el control al bloque de decisión 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]; y frame_crop_bottom_offset[sub_view_id[i]][j].

El bloque de función 2120 decodifica la imagen actual usando codificación de vídeo multivista (MVC), y pasa el control a un bloque de función 2121. El bloque de función 2121 separa cada vista de la imagen usando la sintaxis de alto nivel, y pasa el control a un bloque final 2199. La separación de cada vista usando sintaxis de alto nivel es como se ha descrito anteriormente.

El bloque de decisión 2122 determina si un elemento de sintaxis tiling_mode es igual a cero. Si es así, entonces se pasa el control a un bloque de función 2124. De otro modo, se pasa el control a un bloque de función 2138.

El bloque de función 2124 analiza sintácticamente un elemento de sintaxis flip_dir[sub_view_id[i]] y un elemento de sintaxis upsample_view_flag[sub_view_id[i]], y pasa el control a un bloque de decisión 2126. El bloque de decisión 2126 determina si el valor actual del elemento de sintaxis upsample_view_flag[sub_view_id[i]] es igual a uno. Si es así, entonces se pasa el control a un bloque de función 2128. De otro modo, se pasa el control a un bloque de decisión 2130.

El bloque de función 2128 analiza sintácticamente un elemento de sintaxis upsample_filter[sub_view_id[i]], y pasa el control al bloque de decisión 2130. El bloque de decisión 2130 determina si un valor del elemento de sintaxis upsample_filter[sub_view_id[i]] es igual a tres. Si es así, el control se pasa a un bloque de función 2132. De otro modo, se pasa el control a un bloque de función 2136.

El bloque de función 2132 analiza sintácticamente los siguientes elementos de sintaxis, y pasa el control a un bloque de función 2134: vert_dim[sub_view_id[i]]; hor_dim[sub_view_id[i]]; y quantizer[sub_view_id[i]]. El bloque de función 2134 analiza sintácticamente los coeficientes de filtro para cada componente YUV, y pasa el control al bloque de función 2136.

El bloque de función 2136 incrementa la variable i, y devuelve control al bloque de decisión 2108.

El bloque de función 2138 analiza sintácticamente un elemento de sintaxis pixel_dist_x[sub_view_id[i]] y el elemento de sintaxis flip_dist_y[sub_view_id[i]], y pasa el control a un bloque de función 2140. El bloque de función 2140 establece la variable j igual a cero, y pasa el control a un bloque de decisión 2142. El bloque de decisión 2142 determina si el valor actual de la variable j es menor que el valor actual del elemento de sintaxis num_parts[sub_view_id[i]]. Si es así, entonces se pasa el control a un bloque de función 2144. De otro modo, se pasa el control al bloque de función 2136.

El bloque de función 2144 analiza sintácticamente un elemento de sintaxis num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]], y pasa el control a un bloque de función 2146. El bloque de función 2146 analiza sintácticamente los coeficientes para todas los filtros de organización en mosaico por píxeles, y pasa el control al bloque de función 2136.

Cambiando a la FIG. 22, ejemplos de organización en mosaico a nivel de píxel se indican generalmente con el numeral de referencia 2200. La FIG. 22 se describe adicionalmente más adelante.

40 ORGANIZACIÓN DE MOSAICO DE VISTAS USANDO MPEG-4 AVC O MVC

Una aplicación de codificación de vídeo multivista es TV punto de vista libre (o FTV). Esta aplicación requiere que el usuario pueda moverse libremente entre dos o más vistas. A fin de conseguir esto, las vistas "virtuales" entre dos vistas tienen que ser interpoladas o sintetizadas. Hay varios métodos para realizar interpolación de vistas. Uno de los métodos usa profundidad para la interpolación/síntesis de vistas.

Cada vista puede tener una señal de profundidad asociada. Así, la profundidad se puede considerar que es otra forma de señal de vídeo. La FIG. 9 muestra un ejemplo de una señal de profundidad 900. A fin de permitir aplicaciones tales como FTV, la señal de profundidad se transmite junto con la señal de vídeo. En la estructura propuesta de organización en mosaico, la señal de profundidad también se puede añadir como una de las piezas de mosaico. La FIG. 10 muestra un ejemplo de señales de profundidad añadidas como piezas de mosaico. Las señales de profundidad/piezas de mosaico se muestran en el lado derecho de la FIG. 10.

Una vez la profundidad se codifica como pieza de mosaico del fotograma entero, la sintaxis de alto nivel debe indicar qué pieza de mosaico es la señal de profundidad de modo que el solicitante pueda usar la señal de profundidad apropiadamente.

En el caso cuando la secuencia de entrada (tal como el mostrado en la FIG. 1) se codifica usando un codificador de Estándar MPEG-4 AVC (o un codificador correspondiente a un estándar y/o recomendación diferente de codificación de vídeo), la sintaxis propuesta de alto nivel puede estar presente en, por ejemplo, el conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS), un encabezado de segmento, y/o un mensaje de Información de Mejora Suplementaria (SEI). Una realización del método propuesto se muestra en la TABLA 1 donde la sintaxis está presente en un mensaje de Información de Mejora Suplementaria (SEI).

En el caso cuando la secuencia de entradas de las pseudovistas (tal como la mostrado en la FIG. 1) se codifica usando la extensión de codificación de vídeo multivista (MVC) del codificador de Estándar MPEG-4 AVC (o un codificador correspondiente a codificación de vídeo multivista estándar con respecto a un estándar y/o recomendación de codificación de vídeo diferente), la sintaxis propuesta de alto nivel puede estar presente en la SPS, la PPS, el encabezado de segmento, un mensaje SEI, o un perfil especificado. Una realización del método propuesto se muestra en la TABLA 1. La TABLA 1 muestra elementos de sintaxis presentes en la estructura de Conjunto de Parámetros de Secuencia (SPS), que incluye elementos de sintaxis propuestos según una realización de los presentes principios.

15

TABLA 1

	C	Descriptor
seq_parameter_set_mvc_extension() {		
num_views_minus_1		ue(v)
para (i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
para (i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_10[i]		ue(v)
para (j = 0; j < num_anchor_refs_10[i]; j++)		
anchor_ref_10[i][j]		ue(v)
num_anchor_refs_11[i]		ue(v)
para (j = 0; j < num_anchor_refs_11[i]; j++)		
anchor_ref_11[i][j]		ue(v)
}		
para (i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_10[i]		ue(v)
para (j = 0; j < num_non_anchor_refs_10[i]; j++)		
non_anchor_ref_10[i][j]		ue(v)
num_non_anchor_refs_11[i]		ue(v)
para (j = 0; j < num_non_anchor_refs_11[i]; j++)		
non_anchor_ref_11[i][j]		ue(v)
}		
pseudo_view_present_flag		u(1)
si (pseudo_view_present_flag) {		
tiling_mode		
org_pic_width_in_mbs_minus1		
org_pic_height_in_mbs_minus1		
para (i = 0; i < num_views_minus_1; i++)		
pseudo_view_info(i);		
}		
}		

La TABLA 2 muestra elementos de sintaxis para el elemento de sintaxis pseudo_view_info de la TABLA 1, según una realización de los presentes principios.

TABLA 2

pseudo_view_info (pseudo_view_id) {	C	Descriptor
num_sub_views_minus_1[pseudo_view_id]	5	ue(v)
si (num_sub_views_minus_1 != 0) {		
para (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)
para (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)
}		
si (tiling_mode == 0) {		
flip_dir[sub_view_id[i]][j]	5	u(2)
upsample_view_flag[sub_view_id[i]]	5	u(1)
si (upsample_view_flag[sub_view_id[i]])		
upsample_filter[sub_view_id[i]]	5	u(2)
si (upsample_filter[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)
para (yuv = 0; yuv < 3; yuv++) {		
para (y = 0; y < vert_dim[sub_view_id[i]] - 1; y++) {		
para (x = 0; x < hor_dim[sub_view_id[i]] - 1; x++) {		
filter_coeffs[sub_view_id[i]][yuv][y][x]	5	se(v)
}		
}		
}		
}		
}		
// si (tiling_mode == 0)		
también si (tiling_mode == 1) {		
pixel_dist_x[sub_view_id[i]]		
pixel_dist_y[sub_view_id[i]]		
para (j = 0; j <= num_parts[sub_view_id[i]]; j++) {		
num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]][j]		
para (coeff_idx = 0; coeff_idx <= num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]][j]; j++)		
pixel_tiling_filter_coeffs[sub_view_id[i]][j]		
}		
// para (j = 0; j <= num_parts[sub_view_id[i]]; j++)		
// también si (tiling_mode == 1)		
// para (i = 0; i < num_sub_views_minus_1; i++)		
// si (num_sub_views_minus_1 != 0)		
}		

Semántica de los elementos de sintaxis presentados en la TABLA 1 y la TABLA 2

pseudo_view_present_flag igual a verdadero indica que alguna vista es una supervista o múltiples subvistas.

tiling_mode igual a 0 indica que las subvistas se organizan en mosaico a nivel de imagen. Un valor de 1 indica que la organización en mosaico se hace a nivel de píxel.

- 5 El nuevo mensaje SEI podría usar un valor para la tipo de carga útil SEI que no se ha usado en el Estándar MPEG-4 AVC o una extensión del Estándar MPEG-4 AVC. El nuevo mensaje SEI incluye varios elementos de sintaxis con la siguiente semántica.

num_coded_views_minus1 más 1 indica el número de vistas codificadas soportadas por el flujo de bits. El valor de num_coded_views_minus1 está en el intervalo de 0 a 1023, inclusive.

- 10 org_pic_width_in_mbs_minus1 más 1 especifica la anchura de una imagen en cada vista en unidades de macrobloques.

La variable para la anchura de imagen en unidades de macrobloques se deriva de la siguiente manera:

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

La variable para anchura de imagen para la componente luma se deriva de la siguiente manera:

- 15 PicWidthInSamplesL = PicWidthInMbs * 16

La variable para anchura de imagen para las componentes croma se deriva de la siguiente manera:

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

org_pic_height_in_mbs_minus1 más 1 especifica la altura de una imagen en cada vista en unidades de macrobloques.

- 20 La variable para la altura de imagen en unidades de macrobloques se deriva de la siguiente manera:

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

La variable para altura de imagen para la componente luma se deriva de la siguiente manera:

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

La variable para altura de imagen para las componentes croma se deriva de la siguiente manera:

- 25 PicHeightInSamplesC = PicHeightInMbs * MbHeightC

num_sub_views_minus1 más 1 indica el número de subvistas codificadas incluidas en la vista actual. El valor de num_coded_views_minus1 está en el intervalo de 0 a 1023, inclusive.

sub_view_id[i] especifica el sub_view_id de la subvista con orden de decodificación indicado por i.

num_parts[sub_view_id[i]] especifica el número de partes en la que se parte la imagen de sub_view_id[i].

- 30 loc_left_offset[sub_view_id[i]][j] y loc_top_offset[sub_view_id[i]][j] especifican las ubicaciones en desplazamientos de píxeles izquierdo y superior, respectivamente, donde la parte j actual se ubica en la imagen reconstruida final de la vista con sub_view_id igual a sub_view_id[i].

view_id[i] especifica la view_id de la vista con orden de codificación indicado por i.

- 35 frame_crop_left_offset[view_id[i]][j], frame_crop_right_offset[view_id[i]][j], frame_crop_top_offset[view_id[i]][j], y frame_crop_bottom_offset[view_id[i]][j] especifican las muestras de las imágenes en las secuencia de vídeo codificada que son parte de num_part j y view_id i, en términos de una región rectangular especificada en coordenadas de fotograma para salida.

Las variables CropUnitX y CropUnitY se derivan de la siguiente manera:

Si chroma_format_idc es igual a 0, CropUnitX y CropUnitY se derivan de la siguiente manera:

- 40 CropUnitX = 1

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

De lo contrario (chroma_format_idc es igual a 1, 2 o 3), CropUnitX y CropUnitY se derivan de la siguiente manera:

CropUnitX = SubWidthC

CropUnitY = SubHeightC * (2 - frame_mbs_only_flag)

El rectángulo de recorte de fotograma incluye muestras luma con coordenadas horizontales de fotograma de la siguiente:

- 5 CropUnitX * frame_crop_left_offset a PicWidthInSamplesL - (CropUnitX * frame_crop_right_offset + 1) y coordenadas verticales de fotograma de CropUnitY * frame_crop_top_offset a (16 * FrameHeightInMbs) - (CropUnitY * frame_crop_bottom_offset + 1), inclusive. El valor de frame_crop_left_offset debe estar en el intervalo de 0 a (PicWidthInSamplesL / CropUnitX) - (frame_crop_right_offset + 1), inclusive; y el valor de frame_crop_top_offset debe estar en el intervalo de 0 a (16 * FrameHeightInMbs / CropUnitY) - (frame_crop_bottom_offset + 1), inclusive.
- 10 Cuando chroma_format_idc no es igual a 0, las muestras especificadas correspondientes de las dos distribuciones croma son las muestras que tienen coordenadas de fotograma (x / SubWidthC, y / SubHeightC), donde (x, y) son las coordenadas de fotograma de las muestras luma especificadas.

Para campos decodificados, las muestras especificadas del campo decodificado son las muestras que caen dentro del rectángulo especificado en coordenadas de fotograma.

- 15 num_parts[view_id[i]] especifica el número de partes en las que se parte la imagen de view_id[i].
depth_flag[view_id[i]] especifica si la parte actual es una señal de profundidad. Si depth_flag es igual a 0, entonces la parte actual no es una señal de profundidad. Si depth_flag es igual a 1, entonces la parte actual es una señal de profundidad asociada con la vista identificada por view_id[i].
- 20 flip_dir[sub_view_id[i][j]] especifica la dirección de volteo para la parte actual. flip_dir igual a 0 indica no hay volteo, flip_dir igual a 1 indica volteo en una dirección horizontal, flip_dir igual a 2 indica volteo en una dirección vertical, y flip_dir igual a 3 indica volteo en direcciones horizontal y vertical.
flip_dir[view_id[i][j]] especifica la dirección de volteo para la parte actual. flip_dir igual a 0 indica no hay volteo, flip_dir igual a 1 indica volteo en una dirección horizontal, flip_dir igual a 2 indica volteo en una dirección vertical, y flip_dir igual a 3 indica volteo en direcciones horizontal y vertical.
- 25 loc_left_offset[view_id[i][j]], loc_top_offset[view_id[i][j]] especifica la ubicación en desplazamientos de píxeles, donde la parte actual j se ubica en la imagen reconstruida final de la vista con view_id igual a view_id[i]
upsample_view_flag[view_id[i]] indica si la imagen perteneciente a la vista especificada por view_id[i] tiene que ser supermuestreada. upsample_view_flag[view_id[i]] igual a 0 indica que la imagen con view_id igual a view_id[i] no será supermuestreada. upsample_view_flag[view_id[i]] igual a 1 especifica que la imagen con view_id igual a view_id[i] será supermuestreada.
- 30 upsample_filter[view_id[i]] indica el tipo de filtro que se va a utilizar para supermuestreo. upsample_filter[view_id[i]] igual a 0 indica que se debe usar el filtro 6-tap AVC. upsample_filter[view_id[i]] igual a 1 indica que se debe usar el filtro 4-tap SVC, upsample_filter[view_id[i]] 2 indica que se debe usar filtro bilineal, upsample_filter[view_id[i]] igual a 3 indica que se trasmiten coeficientes de filtro personalizados. Cuando upsample_filter[view_id[i]] no está presente se establece a 0. En esta realización, se usa filtro personalizado 2D. Se puede extender fácilmente a filtro 1D, y algún otro filtro no lineal.
- 35 vert_dim[view_id[i]] especifica la dimensión vertical del filtro 2D personalizado.
hor_dim[view_id[i]] especifica la dimensión horizontal del filtro 2D personalizado.
quantizer[view_id[i]] especifica el factor de cuantización para cada coeficiente de filtro.
- 40 filter_coeffs[view_id[i]] [yuv][y][x] especifica los coeficientes de filtro cuantizados, yuv señala la componente para la que se aplican los coeficientes de filtro, yuv igual a 0 especifica la componente Y, yuv igual a 1 especifica la componente U, y yuv igual a 2 especifica la componente V.
- 45 pixel_dist_x[sub_view_id[i]] y pixel_dist_y[sub_view_id[i]] especifican respectivamente la distancia en la dirección horizontal y la dirección vertical en la pseudovista reconstruida final entre píxeles vecinos en la vista con sub_view_id igual a sub_view_id[i].
num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i][j]] más uno indica el número de los coeficientes de filtro cuando el modo de organización en mosaico se establece igual a 1.
pixel_tiling_filter_coeffs[sub_view_id[i][j]] señala los coeficientes de filtro que se requieren para representar un filtro que se puede usar para filtrar la imagen organizada en mosaico.

Ejemplos de organización en mosaico a nivel de píxel

Cambiando a la FIG. 22, dos ejemplos que muestran la composición de una pseudovista al organizar en mosaico píxeles de cuatro vistas se indican respectivamente con los numerales de referencia 2210 y 2220, respectivamente. Las cuatro vistas se indican colectivamente con el numeral de referencia 2250. Los valores de sintaxis para el primer ejemplo en la FIG. 22 se proporcionan en la TABLA 3 a continuación.

5

TABLA 3

	Valor
pseudo_view_info (pseudo_view_id) {	
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

Los valores de sintaxis para el segundo ejemplo en la FIG. 22 son todos los mismos excepto los siguientes dos elementos de sintaxis: loc_left_offset[3][0] igual a 5 y loc_top_offset[3][0] igual a 3.

- 10 El desplazamiento indica que los píxeles correspondientes a una vista deben comenzar en cierta ubicación desplazada. Esto se muestra en la FIG. 22 (2220). Esto se puede hacer, por ejemplo, cuando dos vistas producen imágenes en las que objetos comunes aparecen desplazados de una vista a otra. Por ejemplo, si cámaras primera y segunda (que representan vistas primera y segunda) toman imágenes de un objeto, el objeto puede parecer desplazado cinco píxeles a la derecha en la segunda vista en comparación con la primera vista. Esto significa que el pixel(i-5, j) en la primera vista corresponde al pixel(i, j) en la segunda vista. Si los píxeles de las dos vistas simplemente se organizan en mosaico píxel a píxel, entonces puede no haber mucha correlación entre píxeles vecinos en la pieza de mosaico, y las ganancias de codificación espacial pueden ser pequeñas. Por el contrario, al desplazar la organización en mosaico de modo que el pixel(i-5, j) de la vista uno se coloca a continuación del pixel(i, j) de la vista dos, la correlación espacial se puede aumentar y la ganancia de codificación espacial también se puede aumentar. Esto sigue porque, por ejemplo, los correspondientes píxeles para el objeto en las vistas primera y segunda están siendo organizados en mosaico uno junto a otro.
- 15 20 Así, la presencia de loc_left_offset y loc_top_offset puede beneficiar a la eficiencia de codificación. La información de desplazamiento puede ser obtenida por medios externos. Por ejemplo, la información de posición de las cámaras o los vectores de disparidad global entre las vistas se puede usar para determinar tal información de desplazamiento.

Como resultado del desplazamiento, a algunos píxeles en la pseudovista no se les asigna valores de píxel de ninguna vista. Continuando con el ejemplo anterior, cuando se organiza en mosaico el pixel(i-5, j) de la vista uno junto con el pixel(i, j) de la vista dos, para valores de $i = 0, \dots, 4$ no hay pixel(i-5, j) de la vista uno para organizar en mosaico, de modo que aquellos píxeles están vacíos en el mosaico. Para aquellos píxeles en la pseudovista (pieza de mosaico) que no tienen asignados valores de píxel de ninguna vista, al menos una implementación usa un procedimiento de interpolación similar al procedimiento de interpolación subpíxel en compensación de movimiento en AVC. Esto es, los píxeles de mosaico vacíos se pueden interpolar de píxeles vecinos. Tal interpolación puede resultar en mayor correlación espacial en la pieza de mosaico y mayor ganancia de codificación para la pieza de mosaico.

- 5 10 En codificación de vídeo, se puede elegir un tipo de codificación diferente para cada imagen, tales como imágenes I, P y B. Para codificación de vídeo multivista, adicionalmente, se definen imágenes de anclaje y no de anclaje. En una realización, se propone que la decisión de agrupar se pueda hacer basada en el tipo de imagen. Esta información de agrupación se señala en sintaxis de alto nivel.

15 Cambiando a la FIG. 11, un ejemplo de 5 vistas organizadas en mosaico en un único fotograma se indica generalmente con el numeral de referencia 1100. En particular, la secuencia *ballroom* se muestra con 5 vistas organizadas en mosaico en un único fotograma. Adicionalmente, se puede ver que la quinta vista está partida en dos partes de modo que se puede disponer en un fotograma rectangular. Aquí, cada vista es de tamaño QVGA por lo que la dimensión total de fotograma es de 640x600. Puesto que 600 no es múltiplo de 16 debe ser extendido a 608.

Para este ejemplo, el mensaje SEI posible podría ser como se muestra en la TABLA 4.

20

TABLA 4

	Valor
multiview_display_info(payloadSize) {	
num_coded_views_minus1	5
org_pic_width_in_mbs_minus1	40
org_pic_height_in_mbs_minus1	30
view_id[0]	0
num_parts[view_id[0]]	1
depth_flag[view_id[0]][0]	0
flip_dir[view_id[0]][0]	0
loc_left_offset[view_id[0]][0]	0
loc_top_offset[view_id[0]][0]	0
frame_crop_left_offset[view_id[0]][0]	0
frame_crop_right_offset[view_id[0]][0]	320
frame_crop_top_offset[view_id[0]][0]	0
frame_crop_bottom_offset[view_id[0]][0]	240
upsample_view_flag[view_id[0]]	1
si (upsample_view_flag[view_id[0]]) {	
vert_dimview_id[0]]	6
hor_dim[view_id[0]]	6
quantizer[view_id[0]]	32
para (yuv= 0; vuv< 3; yuv++) {	
para (y = 0; y < vert_dim[view_id[i]] - 1; y++) {	
para (x = 0; x < hor_dim[view_id[i]] - 1; x++)	
filter_coeffs[view_id[j]] [yuv] [y] [x]	XX
view_id[1]	1

num_parts[view_id[1]]	1
depth_flag[view_id[0]][0]	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
si (upsample_view_flag[view_id[1]]){	
vert_dim[view_id[1]]	6
hor_dim[view_id[1]]	6
quantizer[view_id[1]]	32
para (yuv= 0; yuv< 3; yuv++) {	
para (y = 0; y < vert dim[view_id[i]] - 1; y ++){	
para (x = 0; x < hor dim[view_id[i]] - 1; x ++)	
filter_coeffs[view_id[i]] [yuv] [y] [x]	XX
...(de manera similar para vista 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]] si (upsample_view_flag[view_id[4]]) {	1
vert dim[view_id[4]]	6
hor dim[view_id[4]]	6
quantizer[view_id[4]]	32
para (yuv= 0; yuv< 3; yuv++) {	
para (y = 0; y < vert dim[view_id[i]] - 1; y++) {	
para (x = 0; x < hor dim[view_id[i]] - 1; x++)	
filter_coeffs[view_id[i]] [yuv] [y] [x]	XX

La TABLA 5 muestra la estructura de sintaxis general para transmitir información multivista para el ejemplo mostrado en la TABLA 4.

TABLA 5

multiview_display_info(payloadSize) {	C	Descriptor
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)
para (i = 0; i <= mim_coded_views_minus1; i++) {		
view_id[i]	5	ue(v)
num_parts[view_id[i]]	5	ue(v)
para (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)
si (upsample_view_flag[view_id[i]]))		
upsample_filter[view_id[i]]	5	u(2)
si (upsample filter[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)
para (yuv= 0; yuv< 3; yuv++) {		
para (y = 0; y < vert dim[view id[i]] - 1; y++) {		
para (x = 0; x < hor dim[view id[i]] - 1; x++)		
filter_coeffs[view_id[i]] [yuv][y][x]	5	se(v)
}		
}		

}		
}		
}		

Haciendo referencia a la FIG. 23, se muestra un dispositivo de procesamiento de vídeo 2300. El dispositivo de procesamiento de vídeo 2300 puede ser, por ejemplo, un aparato descodificador u otro dispositivo que recibe vídeo codificado y proporciona, por ejemplo, vídeo decodificado para exponer a un usuario o para almacenamiento. Así, el dispositivo 2300 puede proporcionar su salida a una televisión, monitor de ordenador, o un ordenador u otro dispositivo de procesamiento.

El dispositivo 2300 incluye un decodificador 2310 que recibir una señal de datos 2320. La señal de datos 2320 puede incluir, por ejemplo, un flujo compatible AVC o un MVC. El decodificador 2310 decodifica toda o parte de la señal recibida 2320 y proporciona como salida una señal de vídeo decodificada 2330 e información de organización en mosaico 2340. El vídeo decodificado 2330 y la información de organización en mosaico 2340 se proporcionan a un selector 2350. El dispositivo 2300 también incluye una interfaz de usuario 2360 que recibe una entrada de usuario 2370. La interfaz de usuario 2360 proporciona una señal de selección de imagen 2380, basada en la entrada de usuario 2370, al selector 2350. La señal de selección de imagen 2380 y la entrada de usuario 2370 indican cuál de múltiples imágenes un usuario desea tener expuesta. El selector 2350 proporciona la(s) imagen/imágenes seleccionada(s) como salida 2390. El selector 2350 usa la información de selección de imagen 2380 para seleccionar cuál de las imágenes en el vídeo decodificado 2330 proporcionar como salida 2390. El selector 2350 usa la información de organización en mosaico 2340 para ubicar la(s) imagen/imágenes seleccionada(s) en el vídeo decodificado 2330.

En diversas implementaciones, el selector 2350 incluye la interfaz de usuario 2360, y en otras implementaciones no se necesita interfaz de usuario 2360 porque el selector 2350 recibe la entrada de usuario 2370 directamente sin realizarse una función separada de función de interfaz. El selector 2350 se puede implementar en software o como circuito integrado, por ejemplo. El selector 2350 también puede incorporar el decodificador 2310.

Más generalmente, los descodificadores de diversas implementaciones descritas en esta solicitud pueden proporcionar una salida decodificada que incluye una pieza de mosaico entera. Adicionalmente o como alternativa, los descodificadores pueden proporcionar una salida decodificada que incluye únicamente una o más imágenes seleccionadas (ilustraciones o señales de profundidad, por ejemplo) de la pieza de mosaico.

Como se señala anteriormente, se puede usar sintaxis de alto nivel para realizar señalización según una o más realizaciones de los presentes principios. La sintaxis de alto nivel se puede usar, por ejemplo, aunque sin limitación a esto, señalando cualquiera de lo siguiente: el número de vistas codificadas presentes en el fotograma más grande; la anchura y altura originales de todas las vistas; para cada vista codificada, el identificador de vista correspondiente a la vista; para cada vista codificada, el número de partes en las que está partido el fotograma de una vista; para cada parte de la vista, la dirección de volteo (que puede ser, por ejemplo, no hay volteo, volteo horizontal únicamente, volteo vertical únicamente o volteo horizontal y vertical); para cada parte de la vista, la posición izquierda en píxeles o el número de macrobloques donde la parte actual pertenece en el fotograma final para la vista; para cada parte de la vista, la posición superior de la parte en píxeles o el número de macrobloques donde la parte actual pertenece en el fotograma final para la vista; para cada parte de la vista, la posición izquierda, en el fotograma grande actual decodificado/codificado, de la ventana de recorte en píxeles o número de macrobloques; para cada parte de la vista, la posición derecha, en el fotograma grande actual decodificado/codificado, de la ventana de recorte en píxeles o número de macrobloques; para cada parte de la vista, la posición superior, en el fotograma grande actual decodificado/codificado, de la ventana de recorte en píxeles o número de macrobloques; y para cada parte de la vista, la posición inferior, en el fotograma grande actual decodificado/codificado, de la ventana de recorte en píxeles o número de macrobloques; para cada vista codificada si la vista necesita ser supermuestreada antes de la salida (donde si se tiene que realizar el supermuestreo, se puede usar un sintaxis de alto nivel para indicar el método de supermuestreo (que incluye, pero sin limitación a esto, filtro AVC 6-tap, filtro SVC 4-tap, filtro bilineal o un filtro 1D personalizado, 2D lineal o no lineal).

Cabe señalar que los términos "codificador" y "decodificador" connotan estructuras generales y no se limitan a funciones o rasgos particulares. Por ejemplo, un decodificador puede recibir una portadora modulada que lleva un flujo de bits codificado, y demodular el flujo de bits codificado, así como decodificar el flujo de bits.

Se han descrito diversos métodos. Muchos de estos métodos se detallan para proporcionar abundante divulgación. Cabe señalar, sin embargo, que se contemplan variaciones que pueden variar uno o muchos de los rasgos específicos descritos para estos métodos. Además, muchos de los rasgos que se mencionan son conocidos en la técnica y, por consiguiente, no se describen en gran detalle.

Además, se ha hecho referencia al uso de sintaxis de alto nivel para enviar cierta información en varias implementaciones. Se tiene que entender, sin embargo, que otras implementaciones, no cubiertas por la presente invención, usan sintaxis de menor nivel, ciertamente otros mecanismos completamente (tales como, por ejemplo,

enviar información como parte de datos codificados) para proporcionar la misma información (o variaciones de esa información).

Diversas implementaciones proporcionan organización en mosaico y señalización apropiada para permitir que múltiples vistas (imágenes, más generalmente) sean organizadas en mosaico en una única imagen, codificadas como única imagen, y enviadas como única imagen. La información de señalización puede permitir a un procesador posterior separar las vistas/ímágenes. Estas implementaciones pueden proporcionar una o más ventajas. Por ejemplo, usuarios pueden desear exponer múltiples vistas a modo organizado en mosaico, y estas diversas implementaciones proporcionan una manera eficiente para codificar y trasmisitir o almacenar tales vistas al organizarlas en mosaico antes de codificar y trasmisitirlas/almacenarlas a modo organizado en mosaico.

Implementaciones que organizan en mosaico múltiples vistas en el contexto de AVC y/o MVC también proporcionan ventajas adicionales. AVC se usa ostensiblemente únicamente para una única vista, por tanto no se espera vista adicional. Sin embargo, tales implementaciones basadas en AVC pueden proporcionar múltiples vistas en un entorno AVC porque las vistas organizadas en mosaico se pueden disponer de modo que, por ejemplo, un decodificador sabe que esas imágenes organizadas en mosaico pertenecen a diferentes vistas (por ejemplo, la imagen superior izquierda en la pseudovista es la vista 1, la imagen superior derecha es la vista 2, etc.).

Adicionalmente, MVC ya incluye múltiples vistas, por lo que no se espera que se incluyan múltiples vistas en una única pseudovista. Además, MVC tiene un límite sobre el número de vistas que se pueden soportar, y tales implementaciones basadas en MVC aumentan eficazmente el número de vistas que se pueden soportar al permitir (como en las implementaciones basadas en AVC) organizar en mosaico vistas adicionales. Por ejemplo, cada pseudovista puede corresponder a una de las vistas soportadas de MVC, y el decodificador puede saber que cada "vista soportada" realmente incluye cuatro vistas en un orden predispuesto organizado en mosaico. Así, en este tipo de implementación, el número de vistas posibles es cuatro veces el número de "vistas soportadas".

Las implementaciones descritas en esta memoria pueden ser implementadas en, por ejemplo, un método o un proceso, un aparato, o un programa de software. Incluso si únicamente se trata en el contexto de una única forma de implementación (por ejemplo, tratada únicamente como método), la implementación de rasgos tratados también se puede implementar en otra forma (por ejemplo, un aparato o un programa). Un aparato se puede implementar en, por ejemplo, hardware, software y firmware apropiados. Los métodos se pueden implementar en, por ejemplo, un aparato tal como, por ejemplo, un procesador, que se refiere a dispositivos de procesamiento en general, incluido, por ejemplo, un ordenador, un microprocesador, un circuito integrado, o un dispositivo lógico programable. Dispositivos de procesamiento también incluyen dispositivos de comunicación, tales como, por ejemplo, ordenadores, teléfonos móviles, ayudantes portátiles/digitales personales ("PDA"), y otros dispositivos que facilitan la comunicación de información entre usuarios finales.

Implementaciones de los diversos procesos y rasgos descritos en esta memoria se pueden incorporar en una variedad de diferentes equipos o aplicaciones, particularmente, por ejemplo, equipos o aplicaciones asociados con codificación y decodificación de datos. Ejemplos de equipos incluyen codificadores de vídeo, descodificadores de vídeo, códecs de vídeo, servidores web, descodificadores, ordenadores portátiles, ordenadores personales, teléfonos móviles, PDA, y otros dispositivos de comunicación. Como debe quedar claro, los equipos pueden ser móviles e incluso instalados en un vehículo móvil.

Adicionalmente, los métodos pueden ser implementados por instrucciones que son realizadas por un procesador, y tales instrucciones se pueden almacenar en un medio legible por procesador tal como, por ejemplo, un circuito integrado, un portador de software u otro dispositivo de almacenamiento tal como, por ejemplo, un disco duro, un diskette compacto, una memoria de acceso aleatorio ("RAM"), o una memoria de solo lectura ("ROM"). Las instrucciones pueden formar un programa de aplicación incorporado tangiblemente en un medio legible por procesador. Como debe quedar claro, un procesador puede incluir un medio legible por procesador que tiene, por ejemplo, instrucciones para llevar a cabo un proceso. Tales programas de aplicación se pueden cargar en una máquina, y ser ejecutados por esta, que comprende cualquier arquitectura adecuada. Preferiblemente, la máquina se implementa en una plataforma informática que tiene hardware tal como una o más unidades de procesamiento central ("CPU"), una memoria de acceso aleatorio ("RAM"), e interfaces de entrada/salida ("E/S") . La plataforma informática también puede incluir un sistema operativo y código de microinstrucciones. Los diversos procesos y funciones descritos en esta memoria pueden ser parte del código de microinstrucciones o parte del programa de aplicación, o cualquier combinación de los mismos, que puede ser ejecutado por una CPU. Adicionalmente, otras diversas unidades periféricas se pueden conectar a la plataforma informática tal como una unidad de almacenamiento de datos y una unidad de impresión adicionales.

Como debe ser evidente para el experto en la técnica, implementaciones también pueden producir una señal formateada para llevar información que puede ser, por ejemplo, almacenada o trasmisitida. La información puede incluir, por ejemplo, instrucciones para realizar un método, o datos producidos por una de las implementaciones descritas. Este tipo de señal puede ser formateado, por ejemplo, como onda electromagnética (por ejemplo, usando una zona radiofrecuencia del espectro) o como señal de banda de base. El formateo puede incluir, por ejemplo, codificar un flujo de datos, producir sintaxis, y modular un portadora con el flujo de datos codificado y la sintaxis. La

información que la señal lleva puede ser, por ejemplo, información analógica o digital. La señal puede ser transmitida por una variedad de diferentes enlaces cableados o inalámbricos, como se conoce.

- Además se tiene que entender que, debido a que algunos de los componentes constituyentes de sistema y métodos representados en los dibujos adjuntos se implementan preferiblemente en software, las conexiones reales entre los componentes de sistema o el bloque de funciones de proceso pueden diferir dependiendo de la manera en la que se programen los presentes principios. Dadas las enseñanzas en esta memoria, un experto en la técnica pertinente podrá contemplar estas implementaciones o configuraciones y similares de los presentes principios.

5

REIVINDICACIONES

1. Un programa de ordenador que tiene instrucciones que, cuando son ejecutadas por un sistema o dispositivo de informática, hace que dicho sistema o dispositivo de informática:

- reciba un flujo de vídeo;
- 5 - acceda a una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen (826), incluyendo las múltiples imágenes una primera imagen de una primera vista de un vídeo multivista y una segunda imagen de una segunda vista del vídeo multivista, siendo parte la imagen de vídeo del flujo de vídeo recibido;
- 10 - acceda a información que indica cómo se combinan las múltiples imágenes en la imagen de vídeo accedida, en donde la información accedida indica que al menos una de las múltiples imágenes se voltean individualmente en uno o más de una dirección horizontal o una dirección vertical,
y en donde la información accedida está comprendida en al menos uno de un encabezado de segmento, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un encabezado de unidad de capa de abstracción de red, y un mensaje de información de mejora suplementaria,
- 15 no volteándose la primera imagen de la primera vista y volteándose la segunda imagen de la segunda vista; y
 - decodifique la imagen de vídeo para proporcionar una representación decodificada de las múltiples imágenes combinadas (824, 826), en donde la información accedida (824, 826) se puede usar para procesar posteriormente la representación decodificada para voltear individualmente la al menos una de las múltiples imágenes.
- 20 2. Un flujo de vídeo que comprende:
 - una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen, incluyendo las múltiples imágenes una primera imagen de una primera vista de un vídeo multivista y una segunda imagen de una segunda vista del vídeo multivista; e
 - información que indica cómo se combinan las múltiples imágenes en la imagen de vídeo, en donde la información indica que al menos una de las múltiples imágenes se voltean individualmente en una o más de una dirección horizontal o una dirección vertical,
y en donde la información está comprendida en al menos uno de un encabezado de segmento, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un encabezado de unidad de capa de abstracción de red, y un mensaje de información de mejora suplementaria, no volteándose la primera imagen de la primera vista y volteándose la segunda imagen de la segunda vista.
- 30 3. Un programa de ordenador que tiene instrucciones que, cuando son ejecutadas por un sistema o dispositivo de informática, hace que dicho sistema o dispositivo de informática:
 - genere una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen, incluyendo las múltiples imágenes una primera imagen de una primera vista de un vídeo multivista y una segunda imagen de una segunda vista de un vídeo multivista;
 - genere información que indica cómo se combinan las múltiples imágenes en la imagen de vídeo generada, en donde la información generada indica que al menos una de las múltiples imágenes se voltean individualmente en una o más de una dirección horizontal o una dirección vertical,
y en donde la información generada está comprendida en al menos uno de un encabezado de segmento, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un encabezado de unidad de capa de abstracción de red,
y un mensaje de información de mejora suplementaria,
no volteándose la primera imagen de la primera vista y volteándose la segunda imagen de la segunda vista;
 - codifique la imagen de vídeo generada y la información generada; y
- 45 - proporcione un flujo de vídeo que incluya la imagen de vídeo codificada y la información codificada.

100



FIG. 1

200

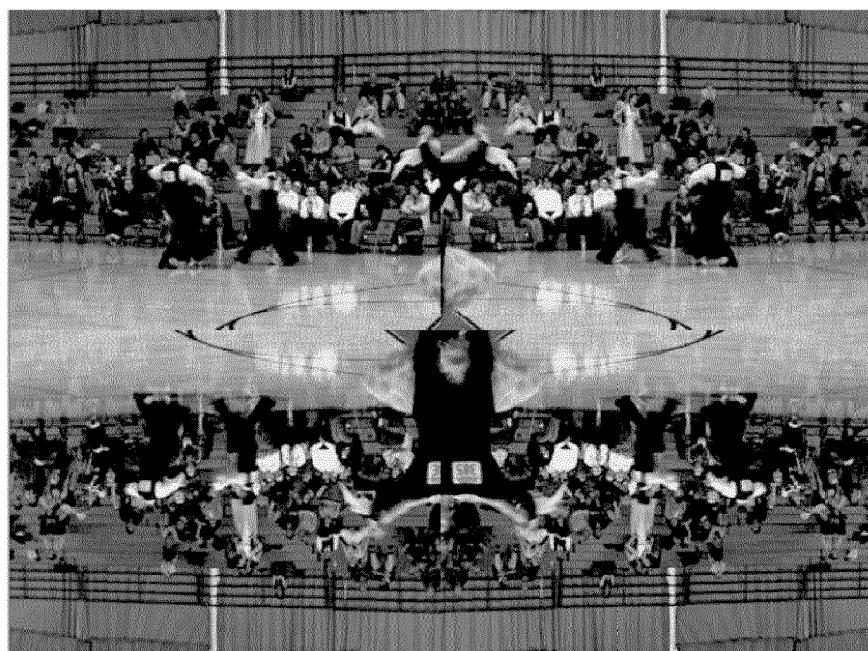


FIG. 2

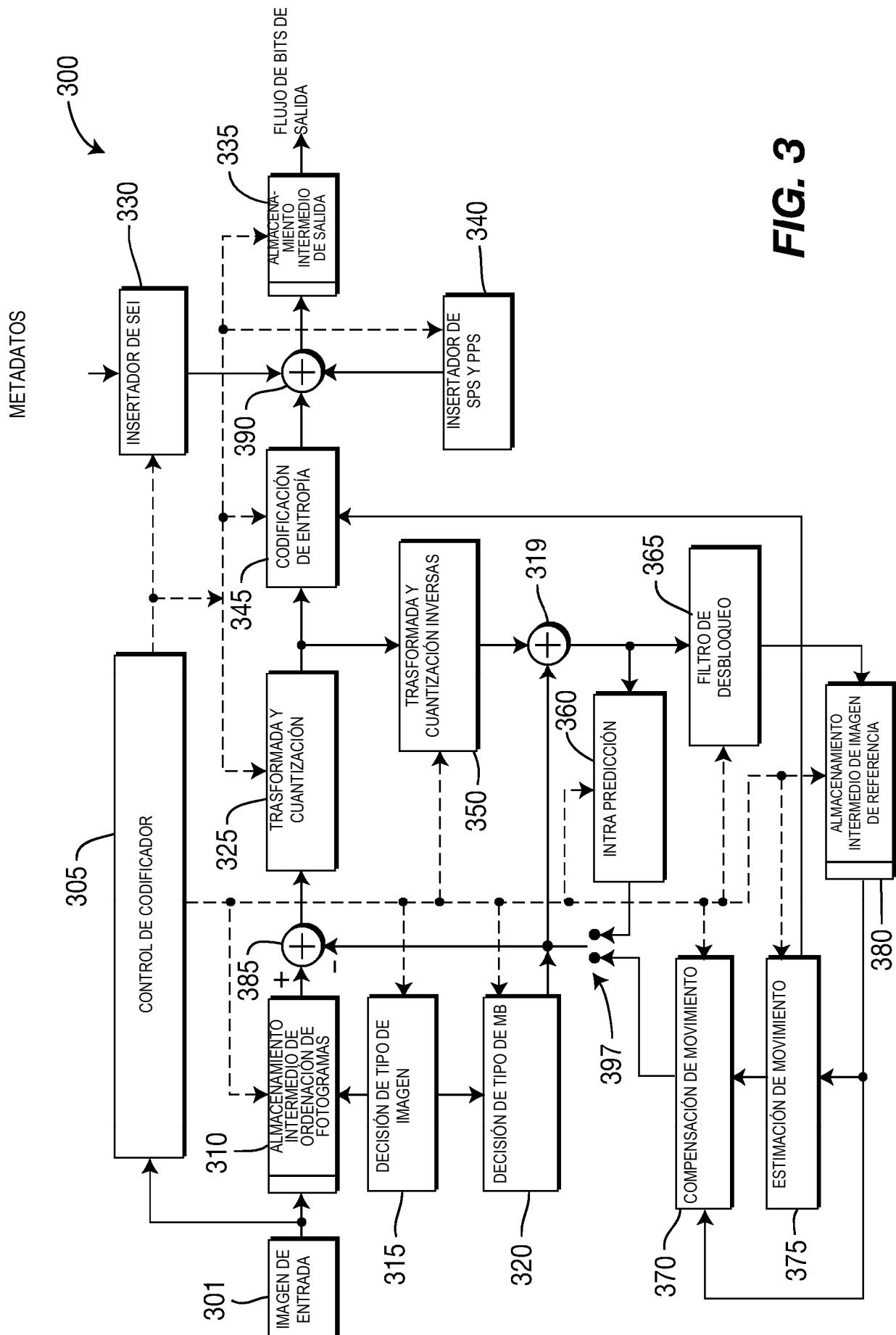


FIG. 3

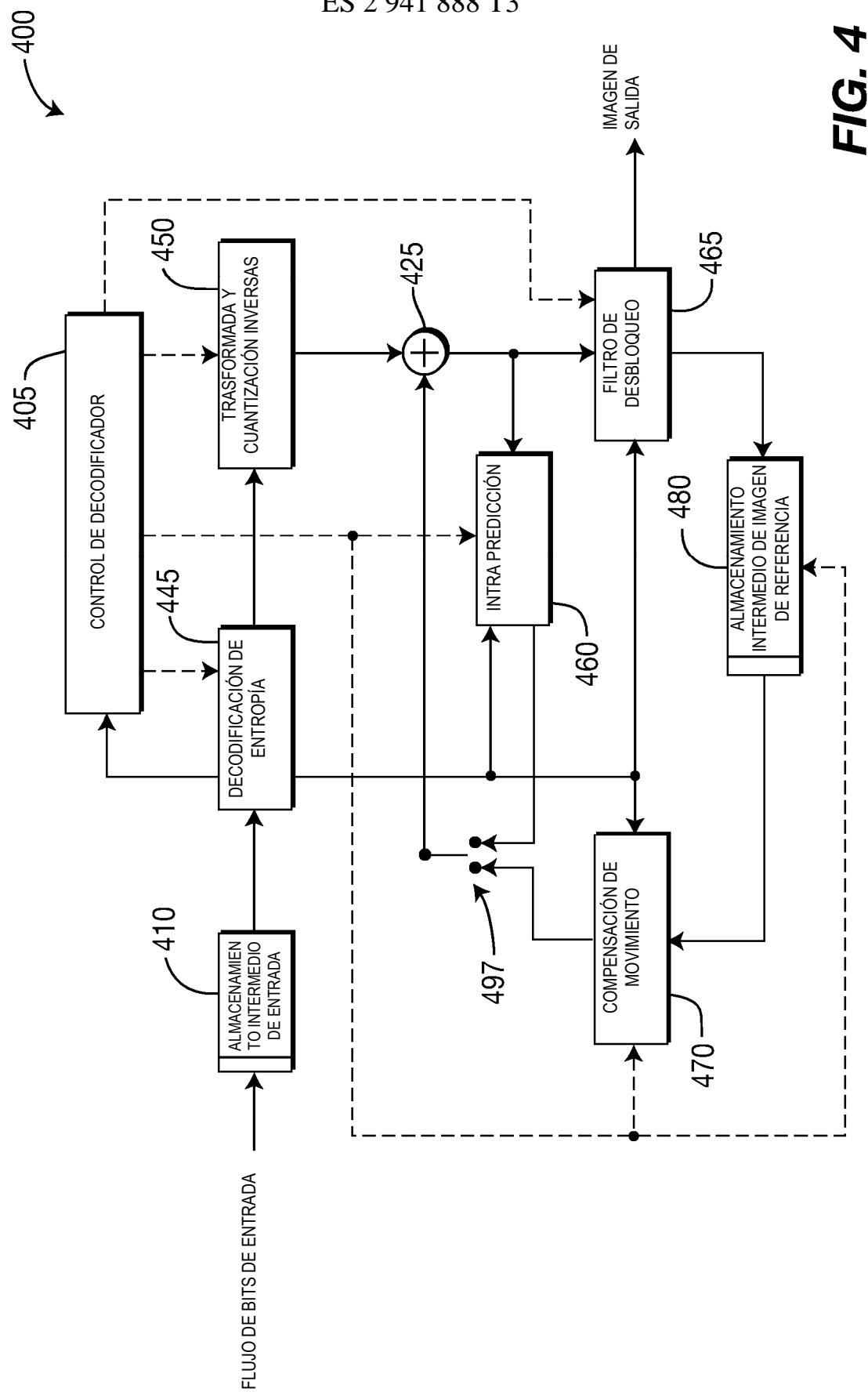
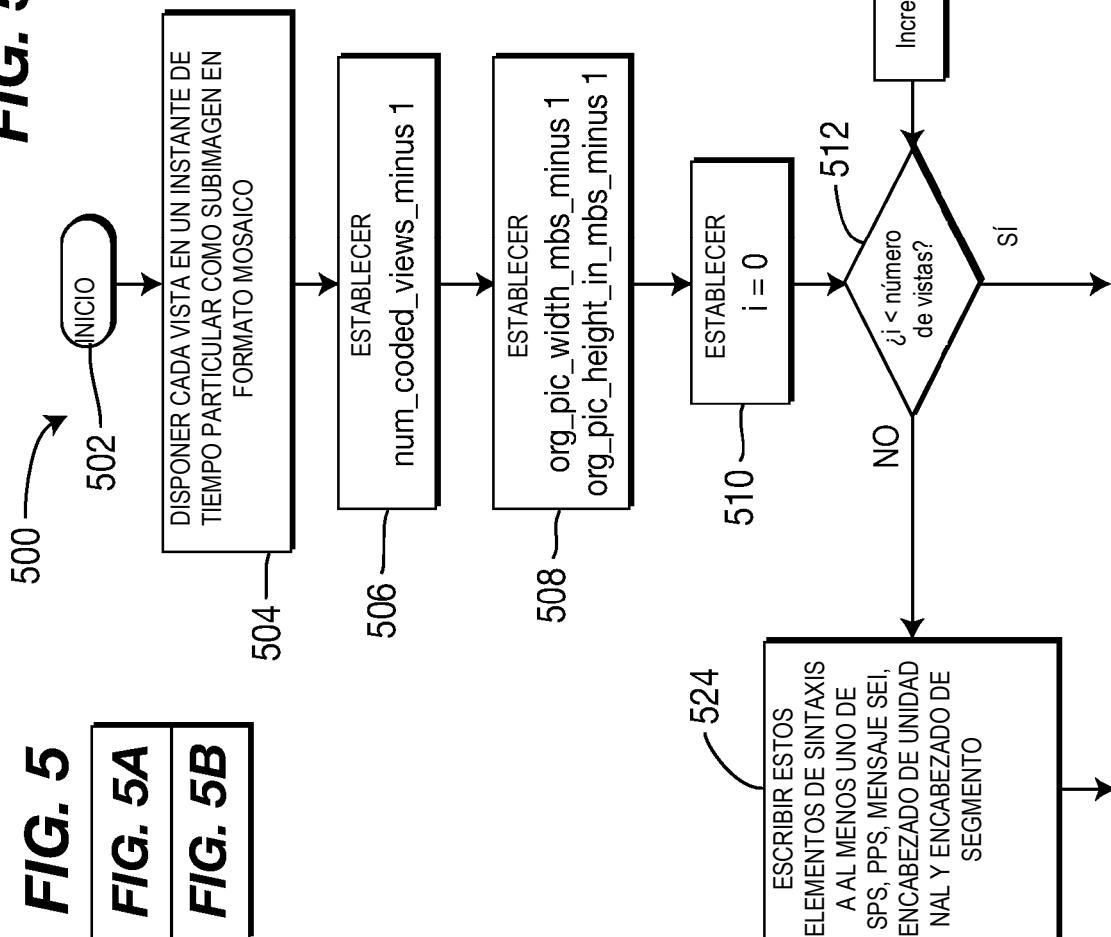
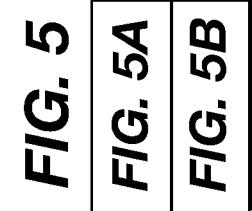
**FIG. 4**

FIG. 5**FIG. 5A****FIG. 5B**

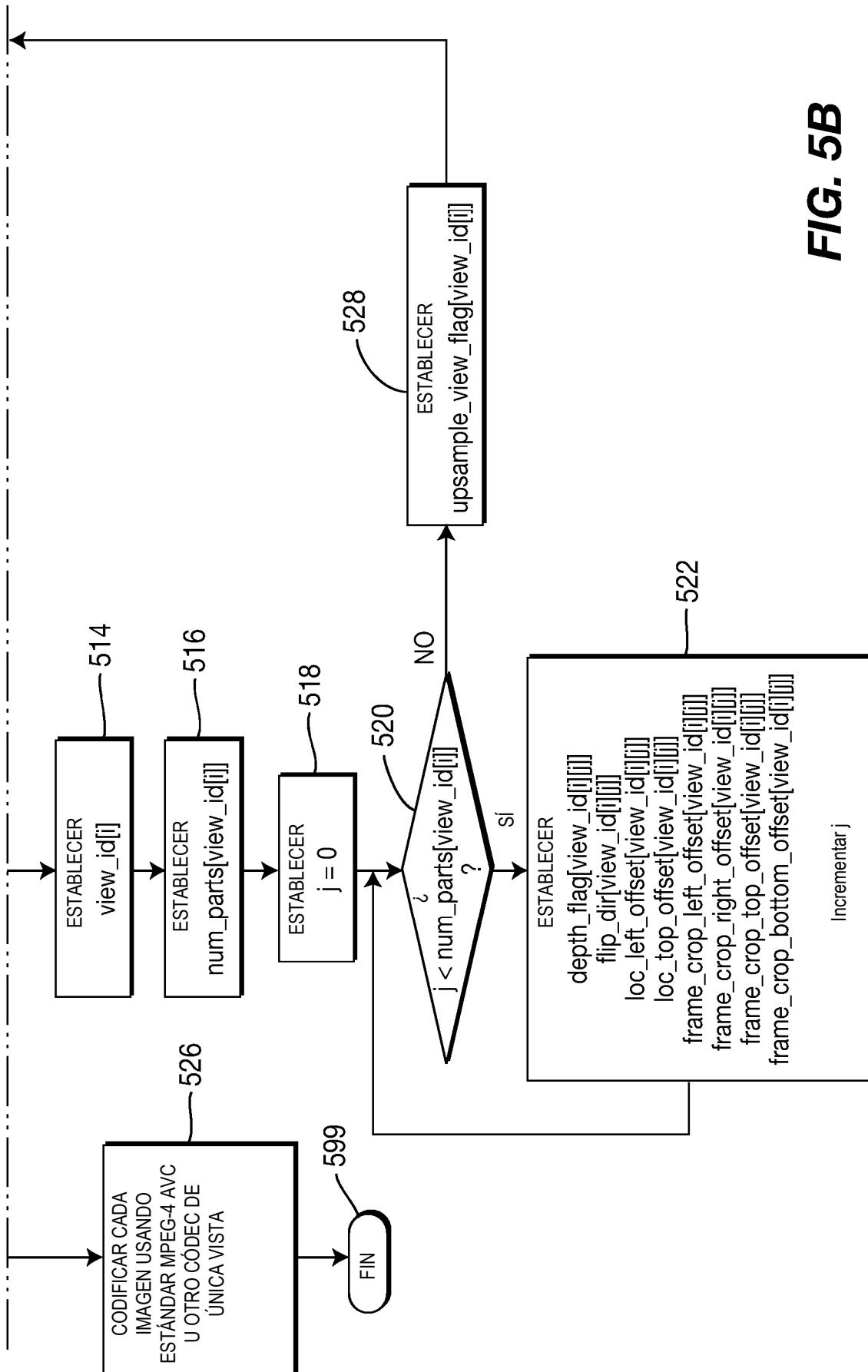
**FIG. 5B**

FIG. 6

FIG. 6A

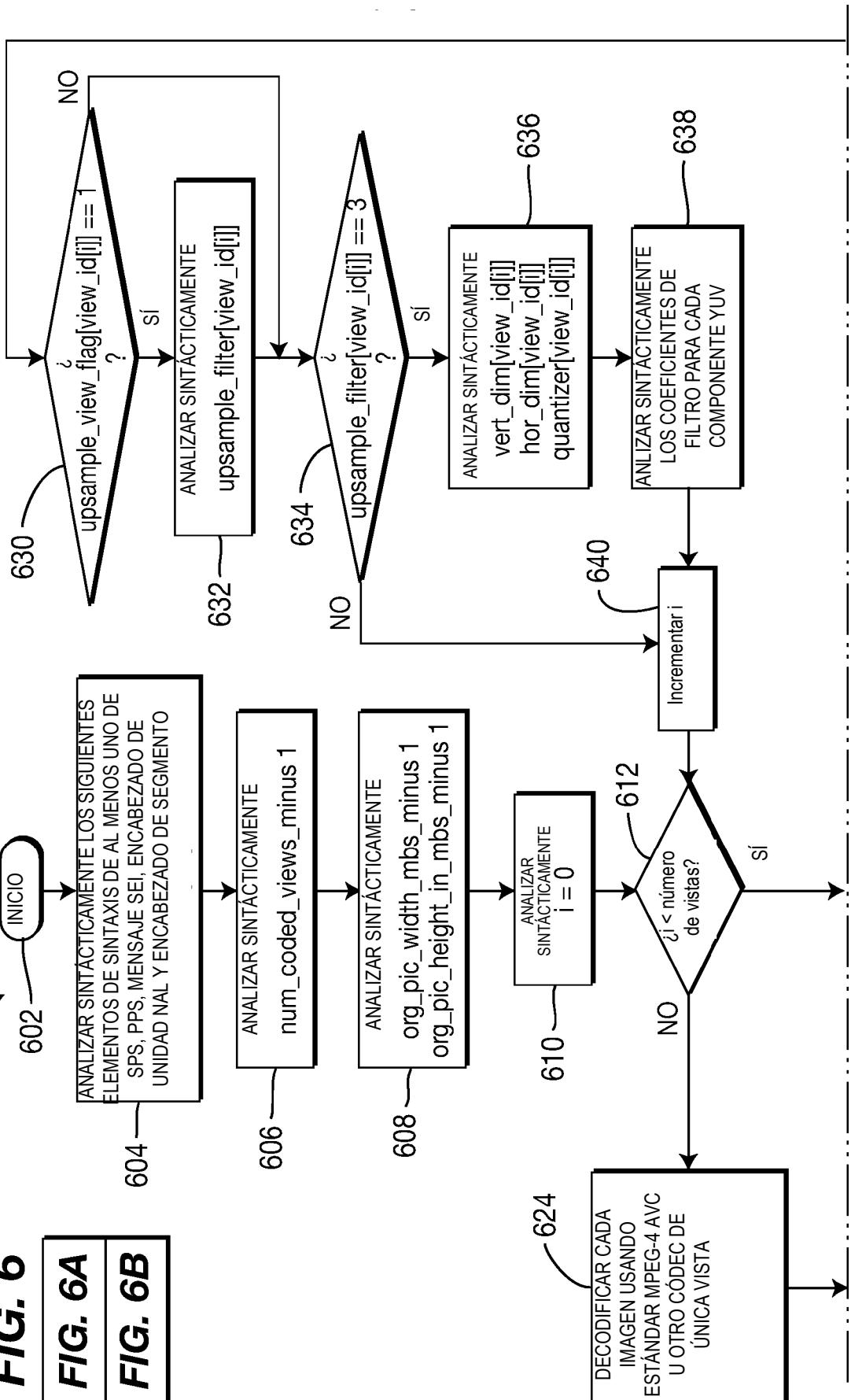
600 → INICIO
602 → ANALIZAR SINTÁCTICAMENTE LOS SIGUIENTES ELEMENTOS DE SINTAXIS DE AL MENOS UNO DE SPS, PPS, MENSAJE SEI, ENCABEZADO DE UNIDADNAL Y ENCABEZADO DE SEGMENTO
604 → num_coded_views_minus 1
606 → ANALIZAR SINTÁCTICAMENTE org_pic_width_mbs_minus 1
608 → org_pic_height_in_mbs_minus 1
610 → ANALIZAR SINTÁCTICAMENTE i = 0
612 → ¿i < número de vistas?
624 → DECODIFICAR CADA IMAGEN USANDO ESTÁNDAR MPEG-4 AVC U OTRO CÓDÉC DE ÚNICA VISTA

FIG. 6B

600 → INICIO
602 → ANALIZAR SINTÁCTICAMENTE LOS SIGUIENTES ELEMENTOS DE SINTAXIS DE AL MENOS UNO DE SPS, PPS, MENSAJE SEI, ENCABEZADO DE UNIDADNAL Y ENCABEZADO DE SEGMENTO
604 → num_coded_views_minus 1
606 → ANALIZAR SINTÁCTICAMENTE org_pic_width_mbs_minus 1
608 → org_pic_height_in_mbs_minus 1
610 → ANALIZAR SINTÁCTICAMENTE i = 0
612 → ¿i < número de vistas?
624 → DECODIFICAR CADA IMAGEN USANDO ESTÁNDAR MPEG-4 AVC U OTRO CÓDÉC DE ÚNICA VISTA

FIG. 6A

600 → INICIO
602 → ANALIZAR SINTÁCTICAMENTE LOS SIGUIENTES ELEMENTOS DE SINTAXIS DE AL MENOS UNO DE SPS, PPS, MENSAJE SEI, ENCABEZADO DE UNIDADNAL Y ENCABEZADO DE SEGMENTO
604 → num_coded_views_minus 1
606 → ANALIZAR SINTÁCTICAMENTE org_pic_width_mbs_minus 1
608 → org_pic_height_in_mbs_minus 1
610 → ANALIZAR SINTÁCTICAMENTE i = 0
612 → ¿i < número de vistas?
624 → DECODIFICAR CADA IMAGEN USANDO ESTÁNDAR MPEG-4 AVC U OTRO CÓDÉC DE ÚNICA VISTA



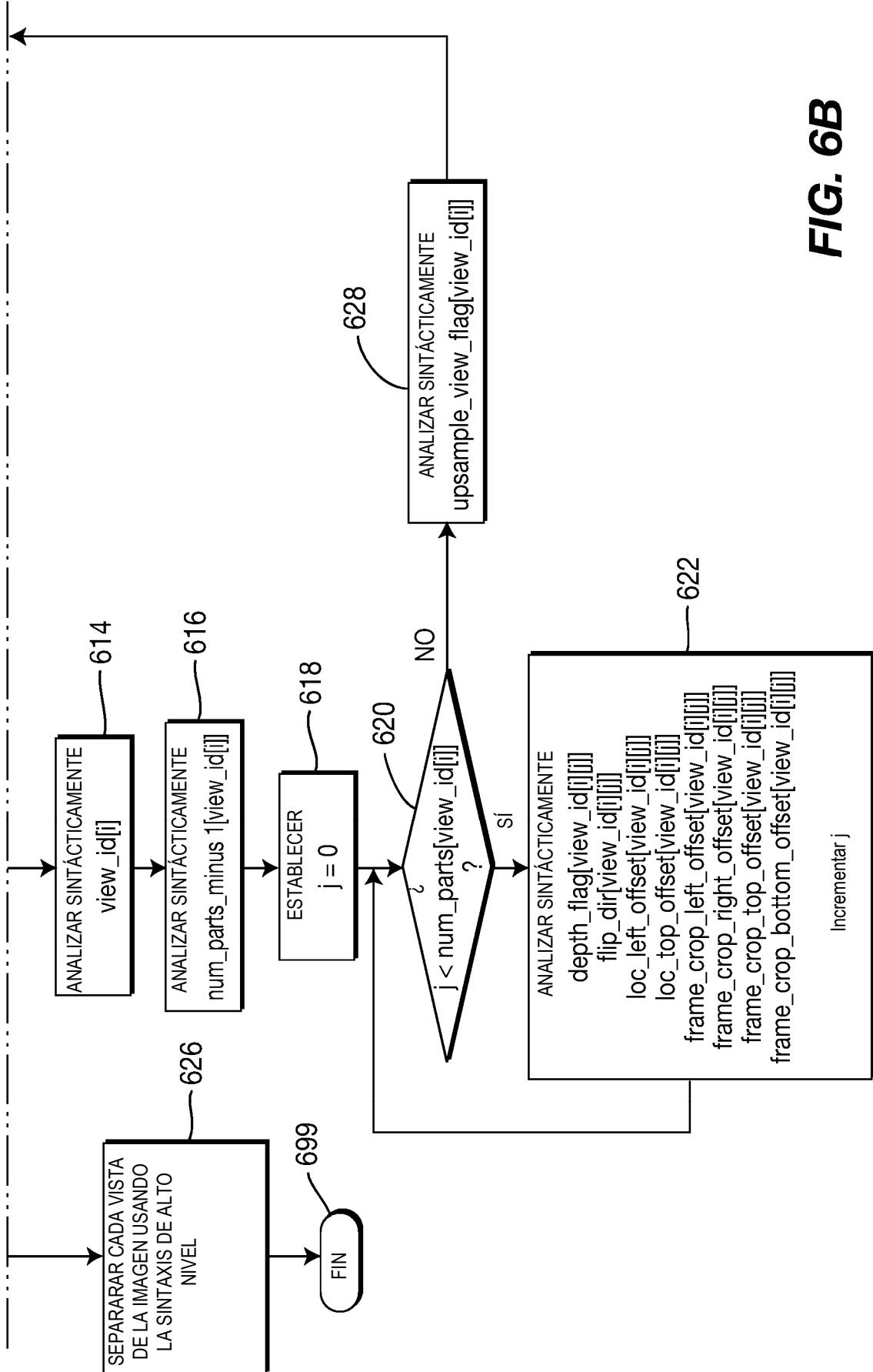
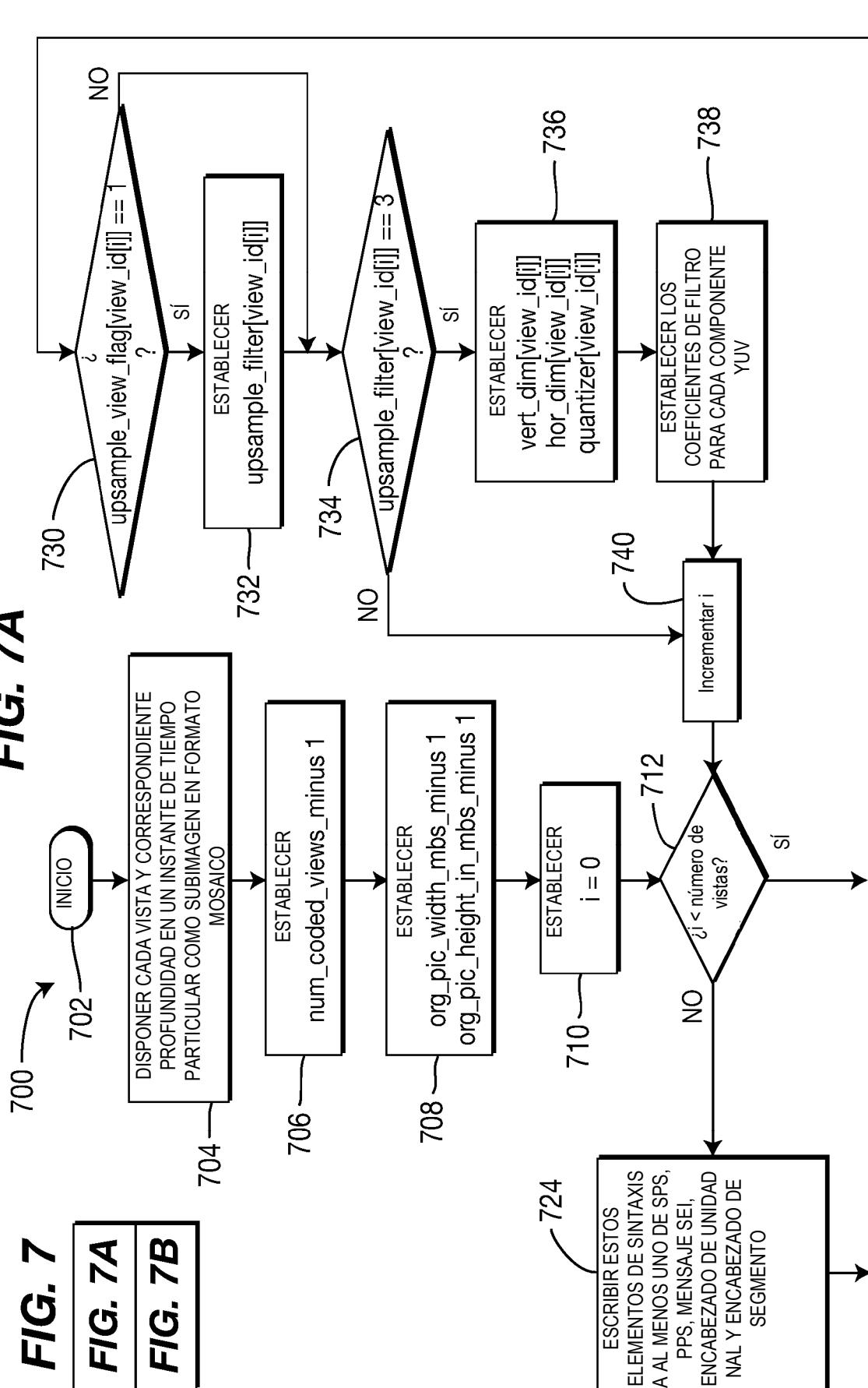
**FIG. 6B**

FIG. 7**FIG. 7A**

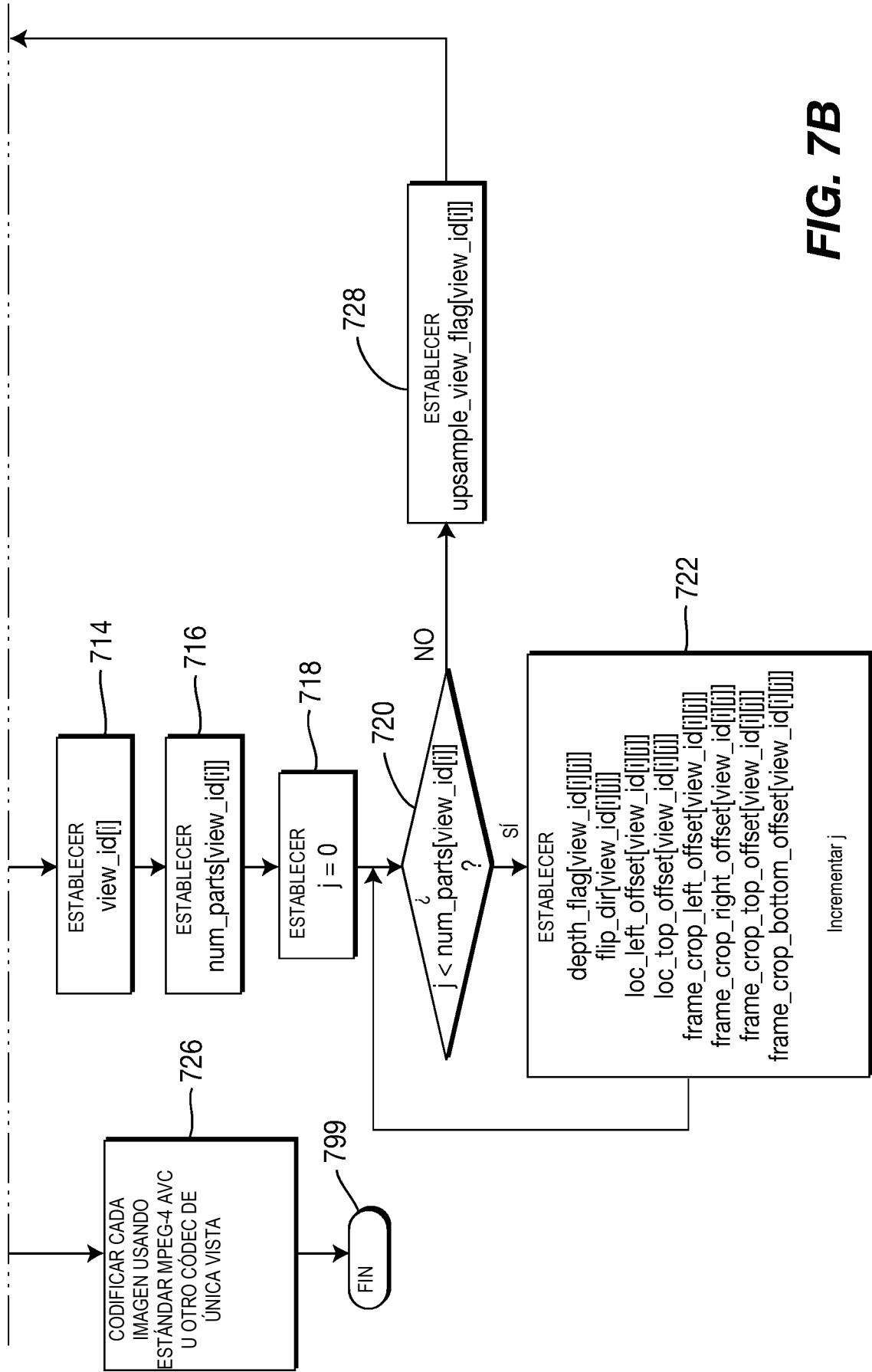
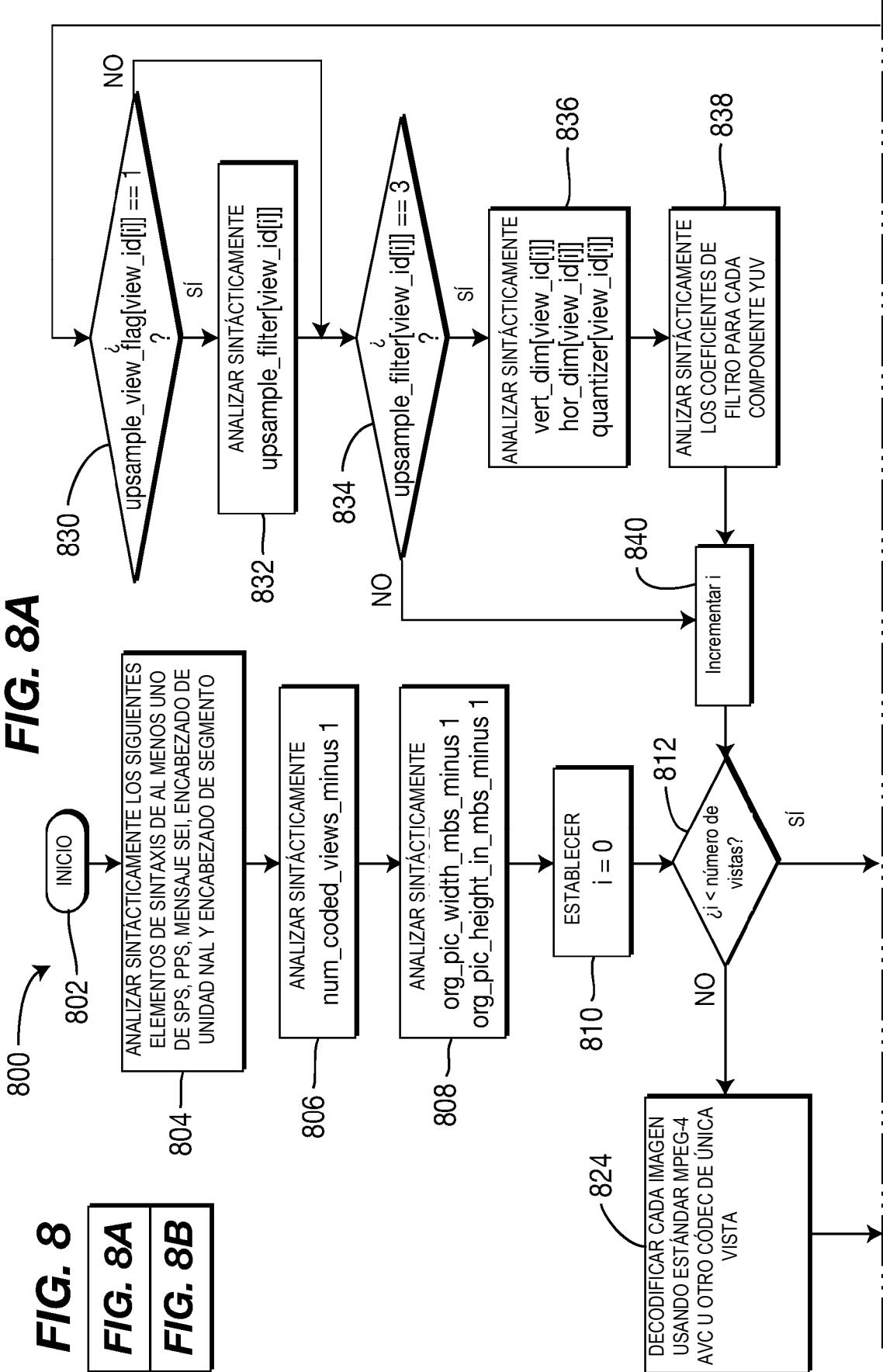
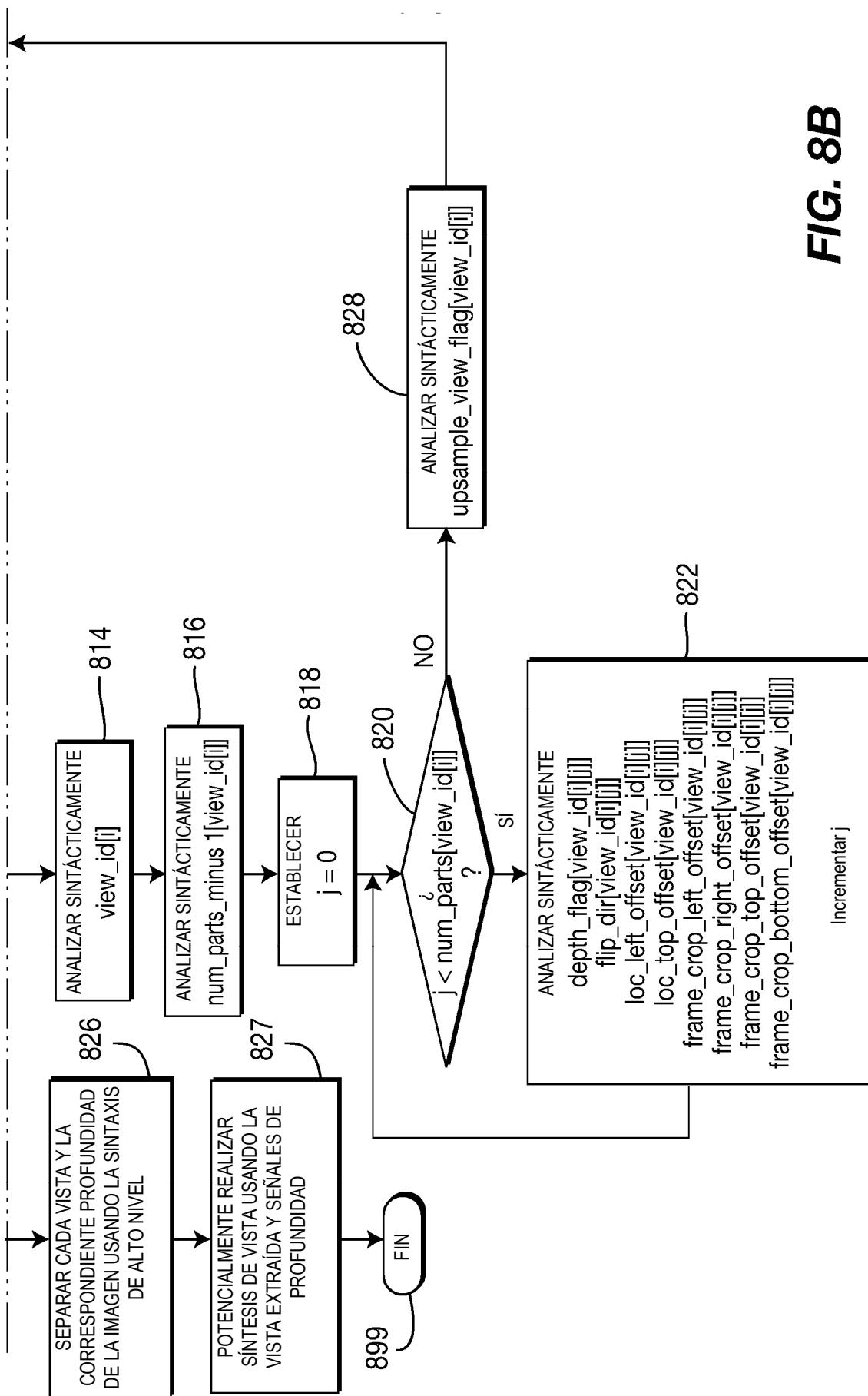
**FIG. 7B**

FIG. 8**FIG. 8A****FIG. 8B**



900

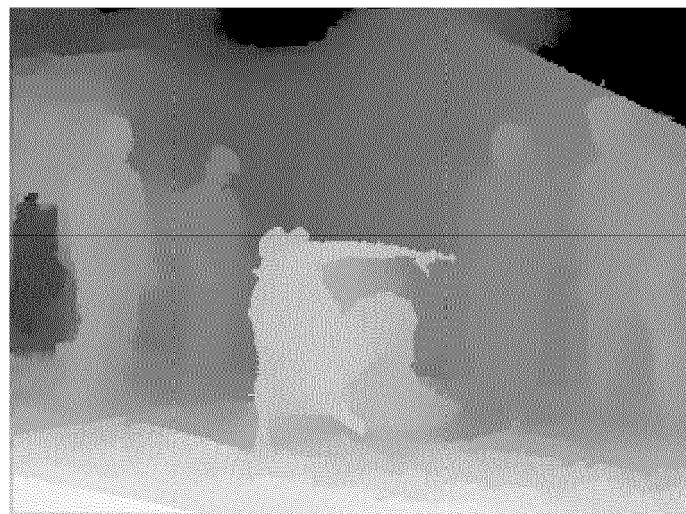


FIG. 9

1000

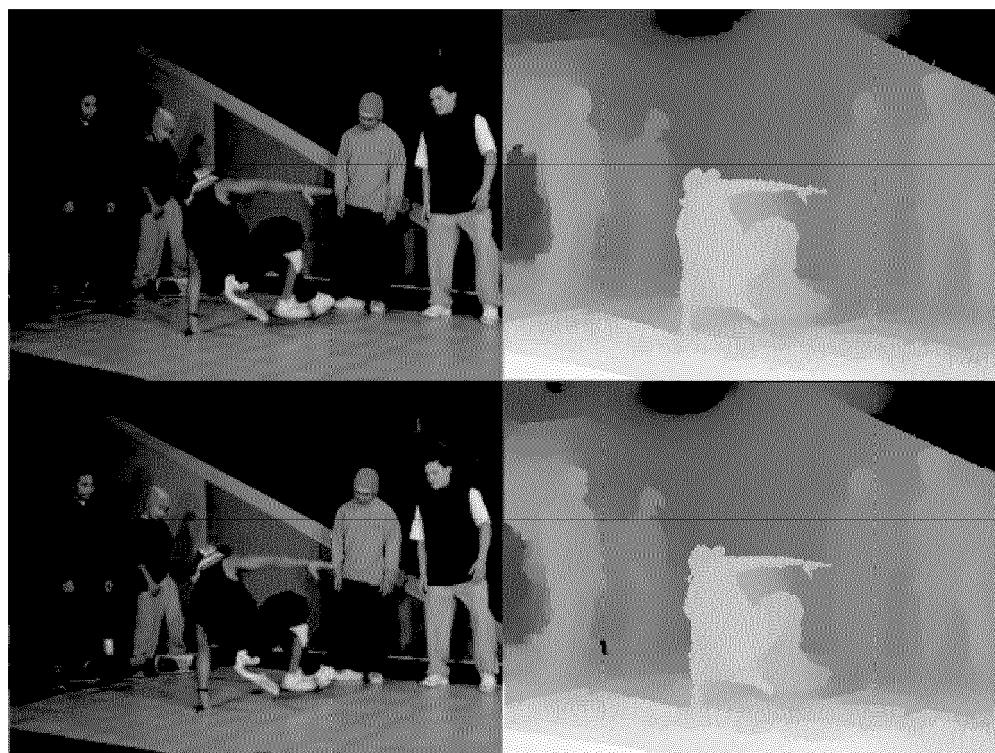


FIG. 10

1100

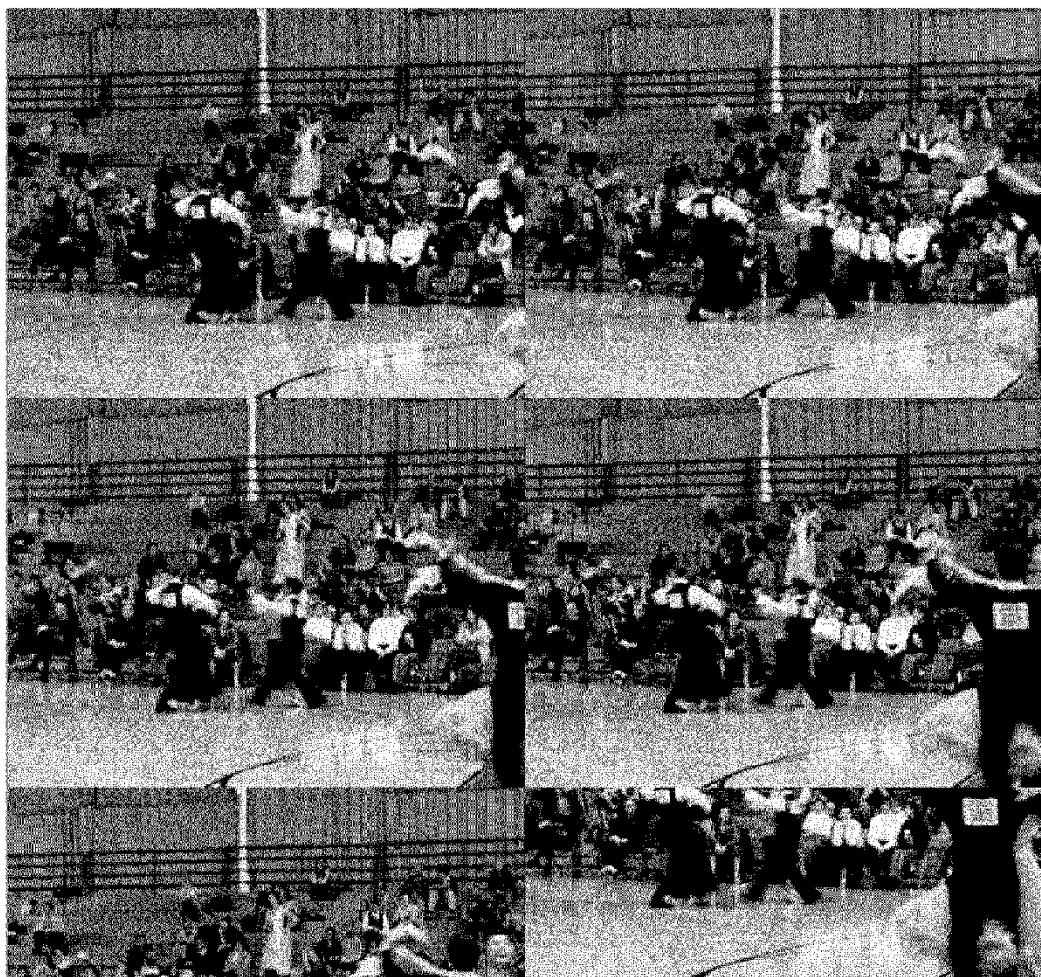


FIG. 11

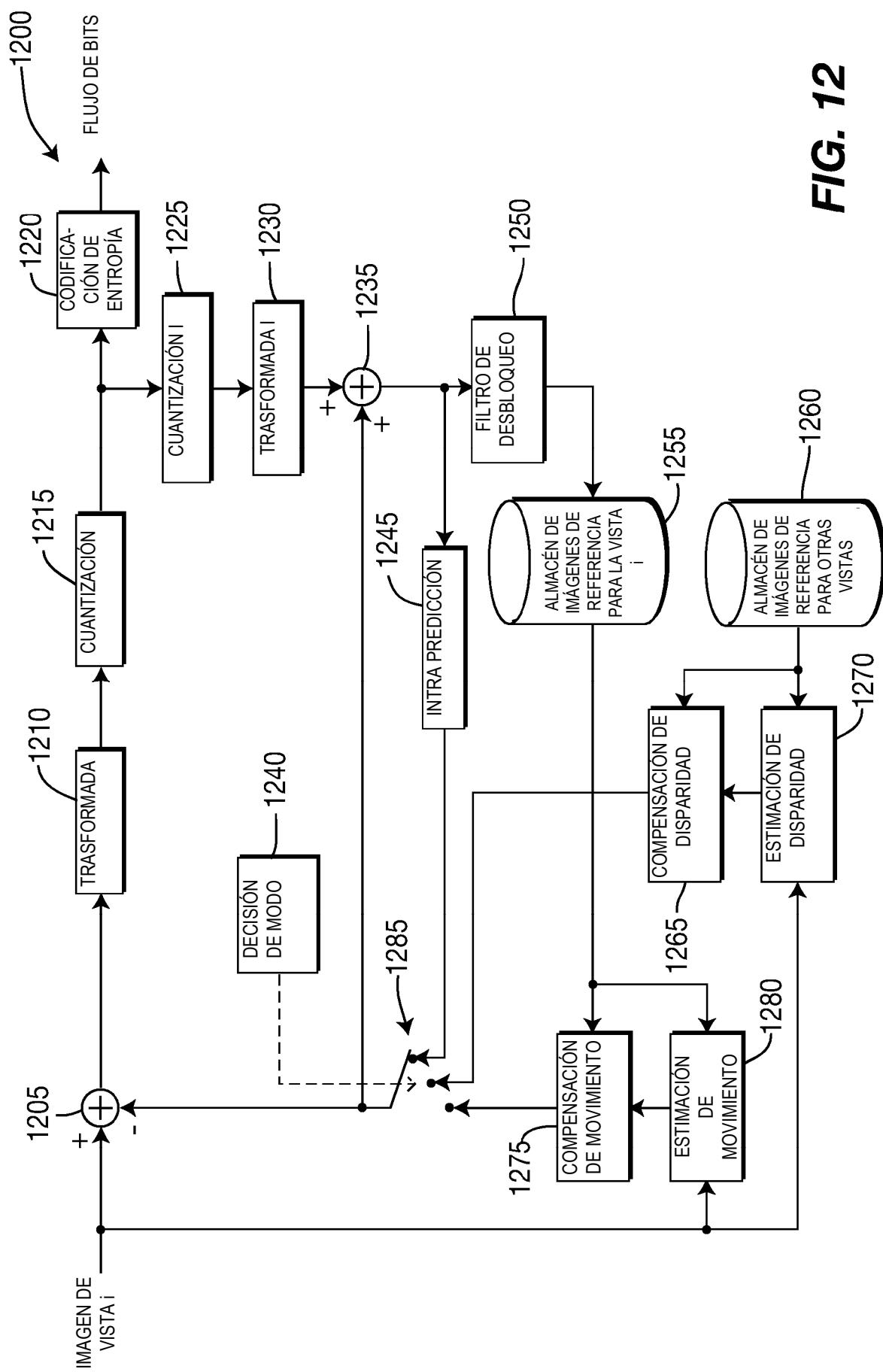
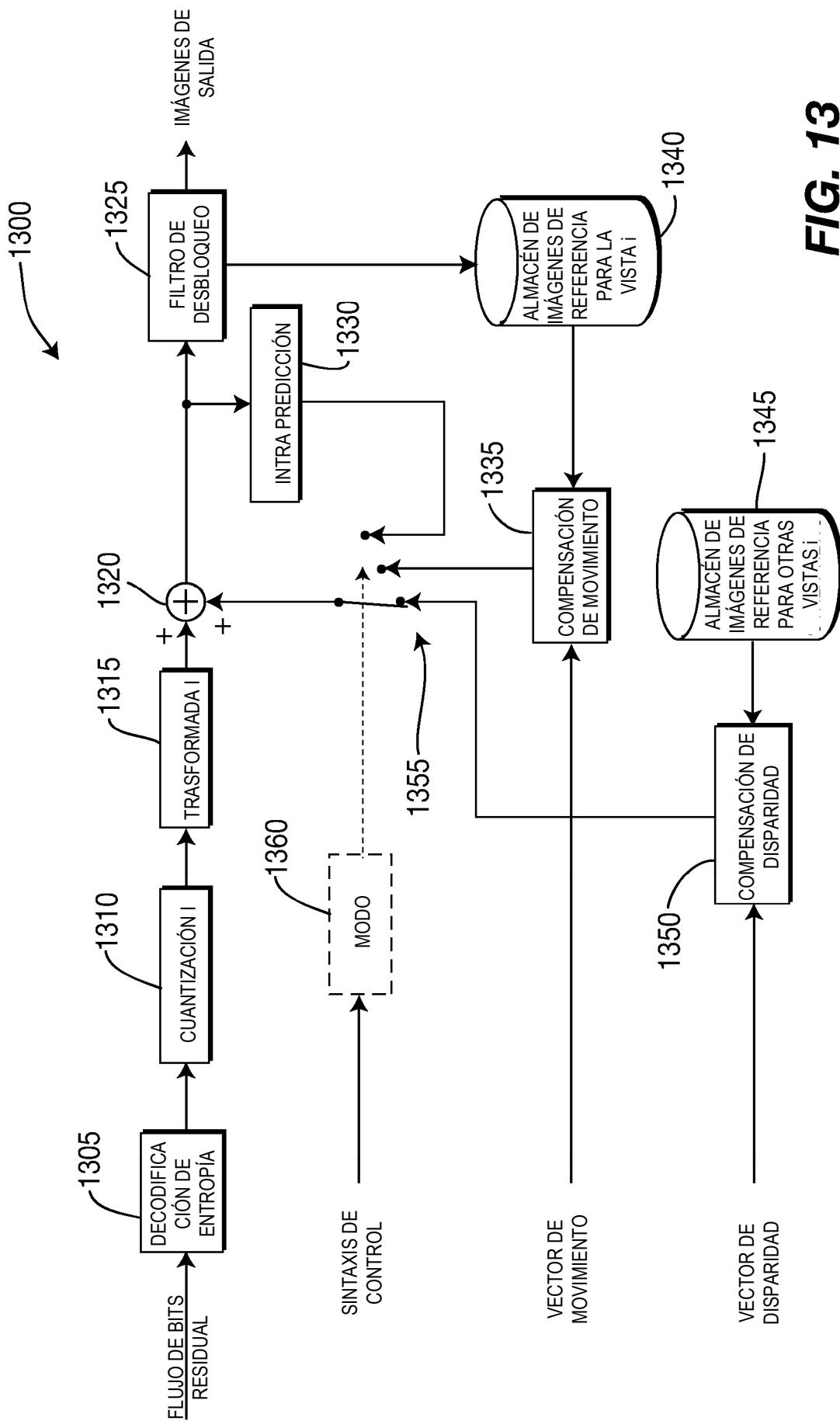


FIG. 12

**FIG. 13**

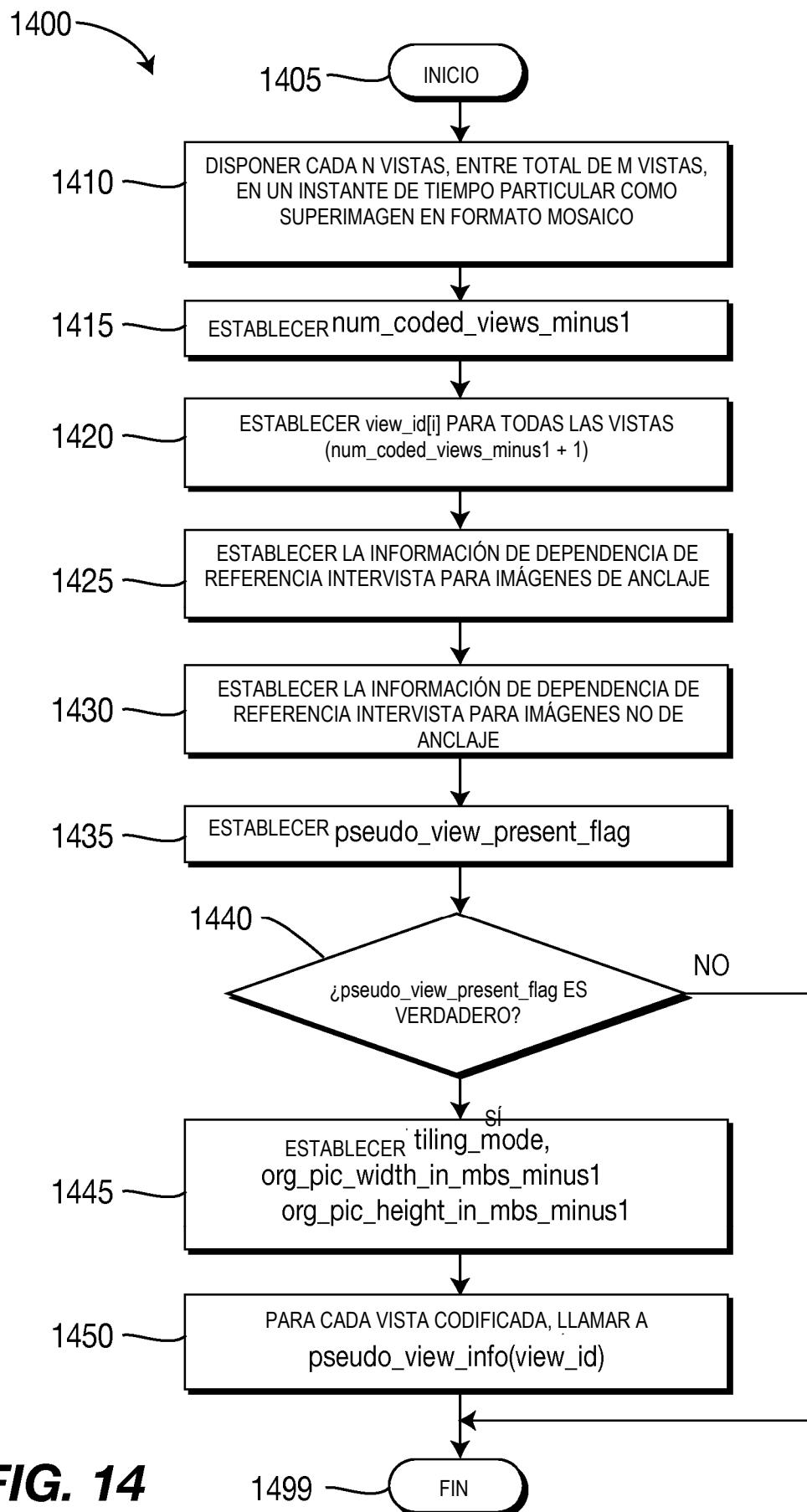
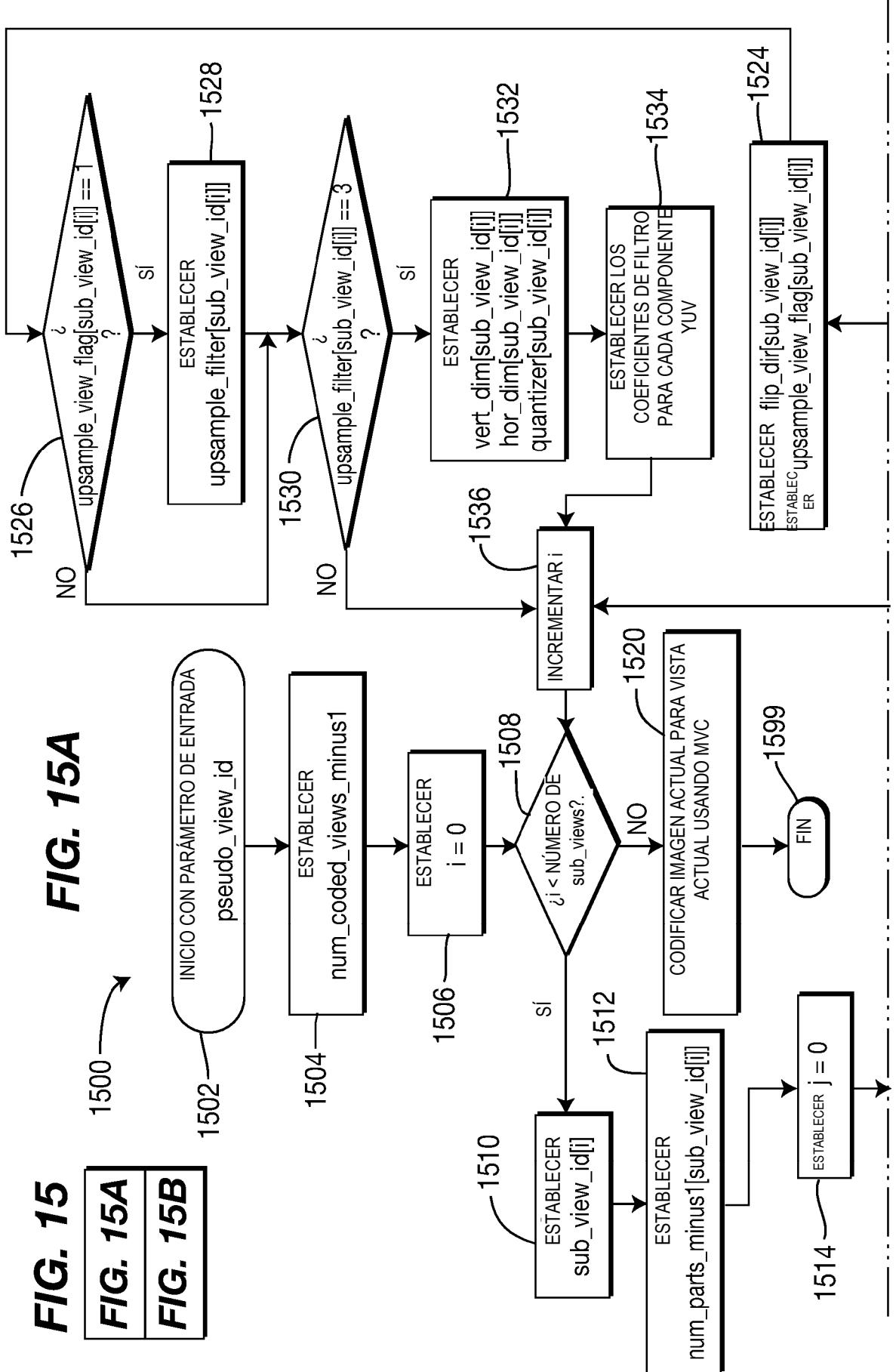
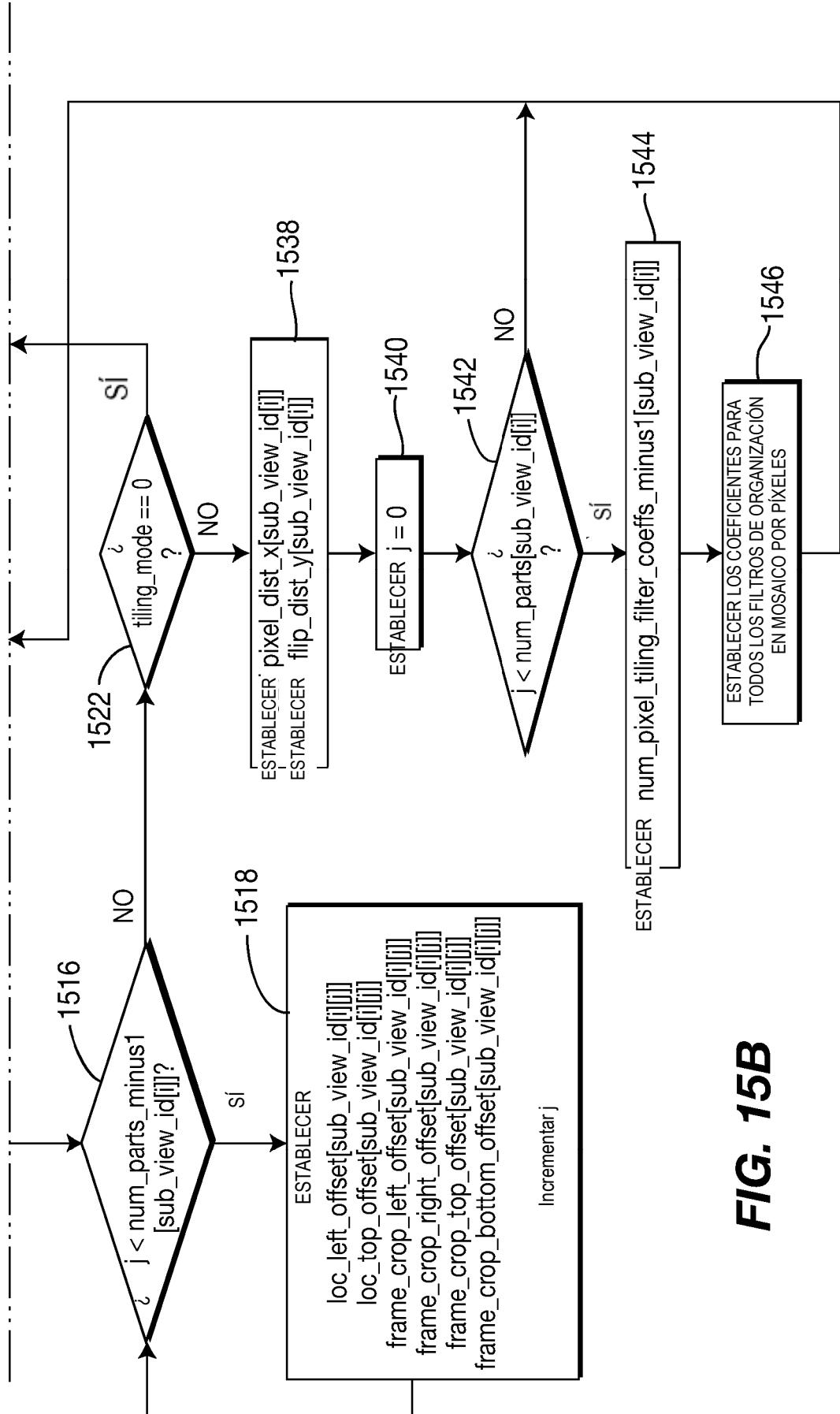
**FIG. 14**

FIG. 15**FIG. 15A****FIG. 15B**

**FIG. 15B**

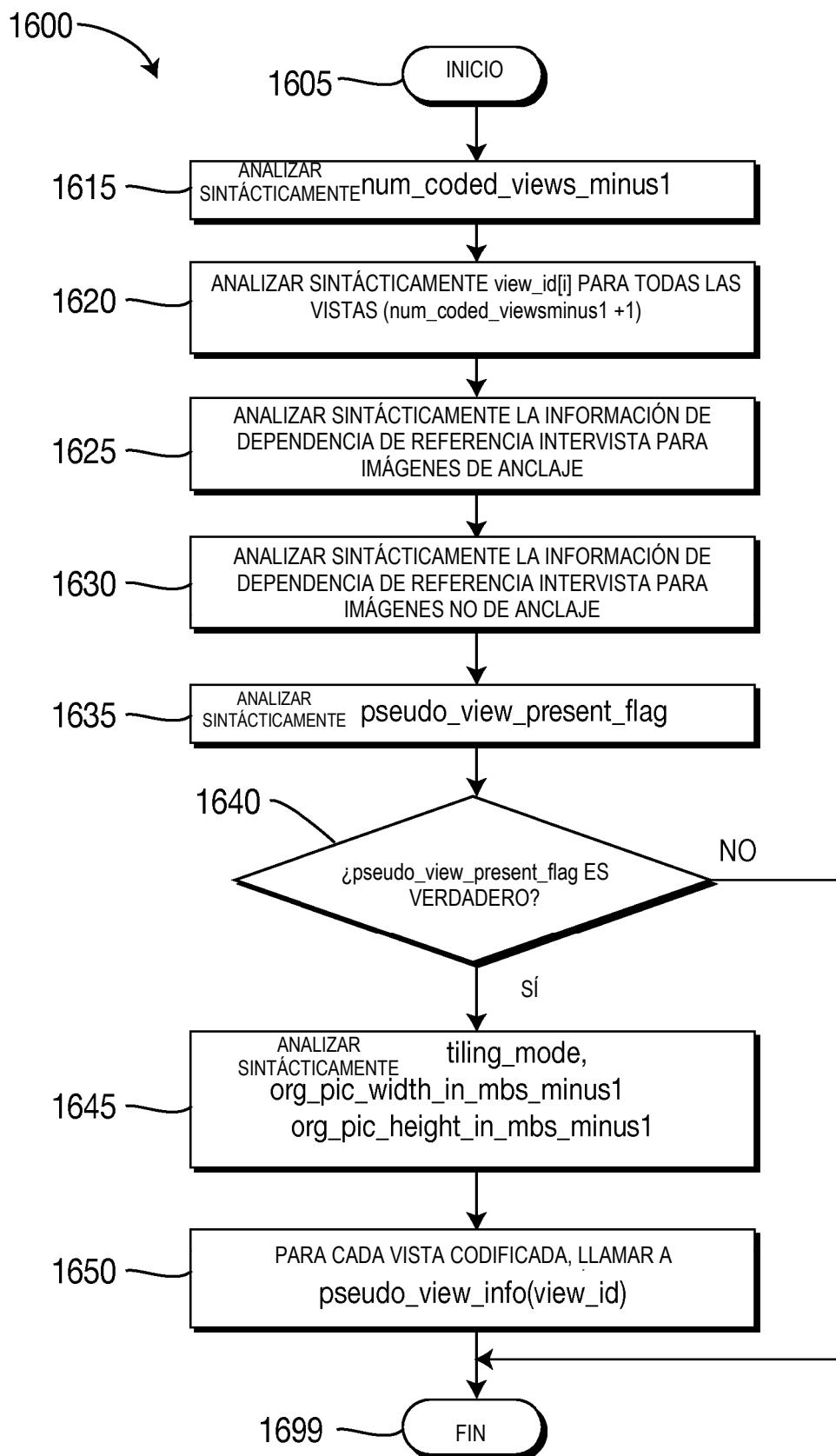
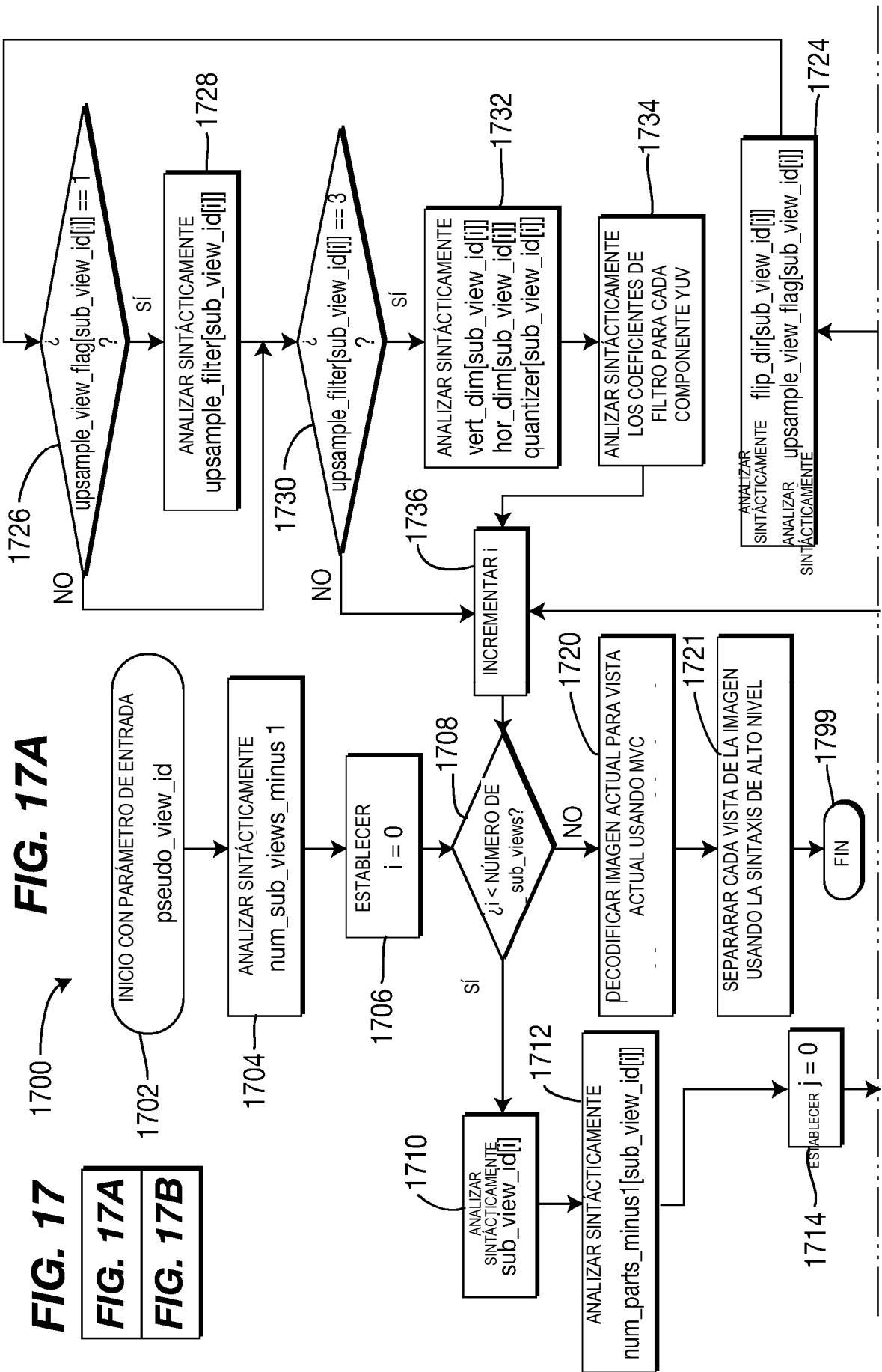
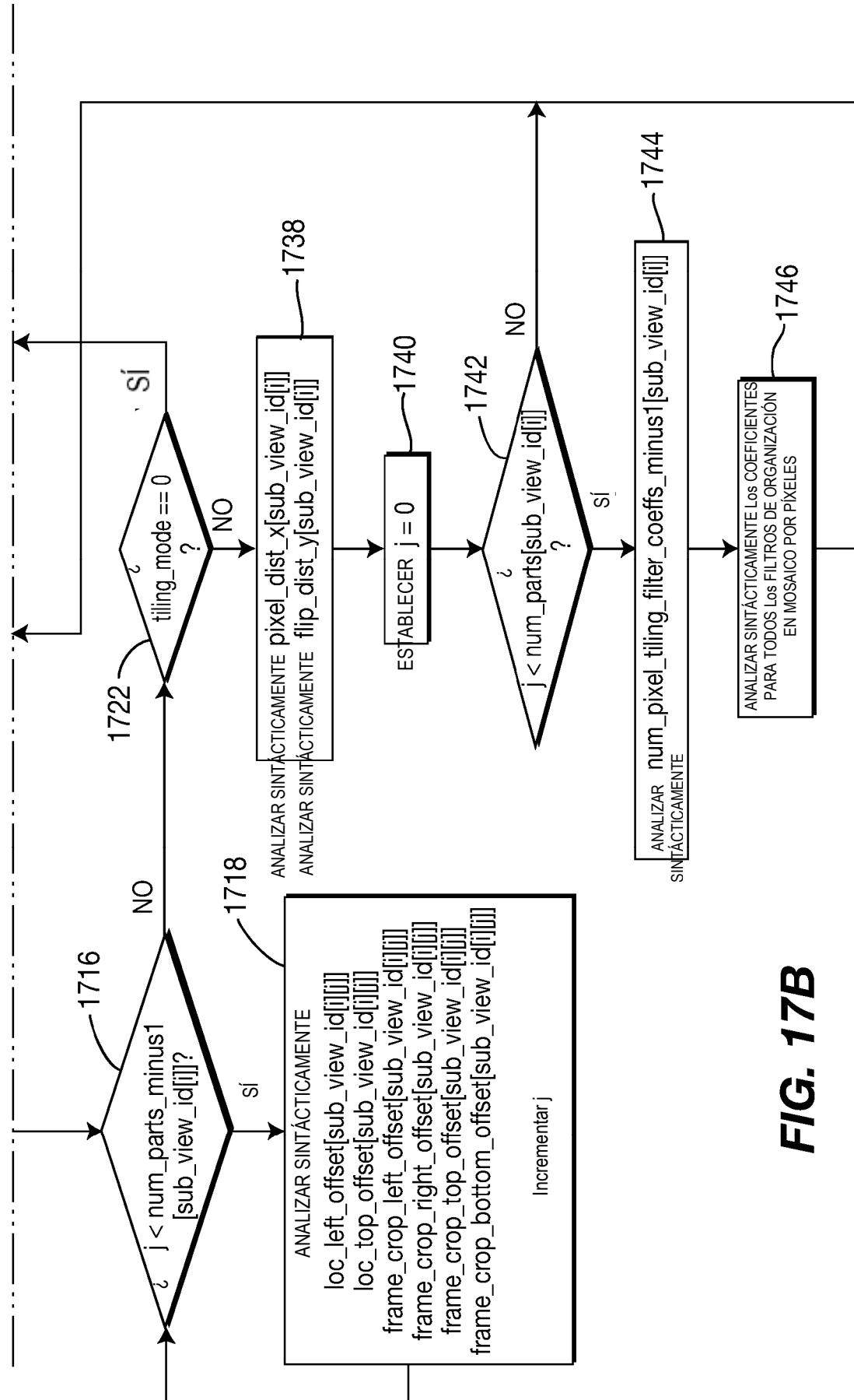
**FIG. 16**

FIG. 17**FIG. 17A****FIG. 17B**

**FIG. 17B**

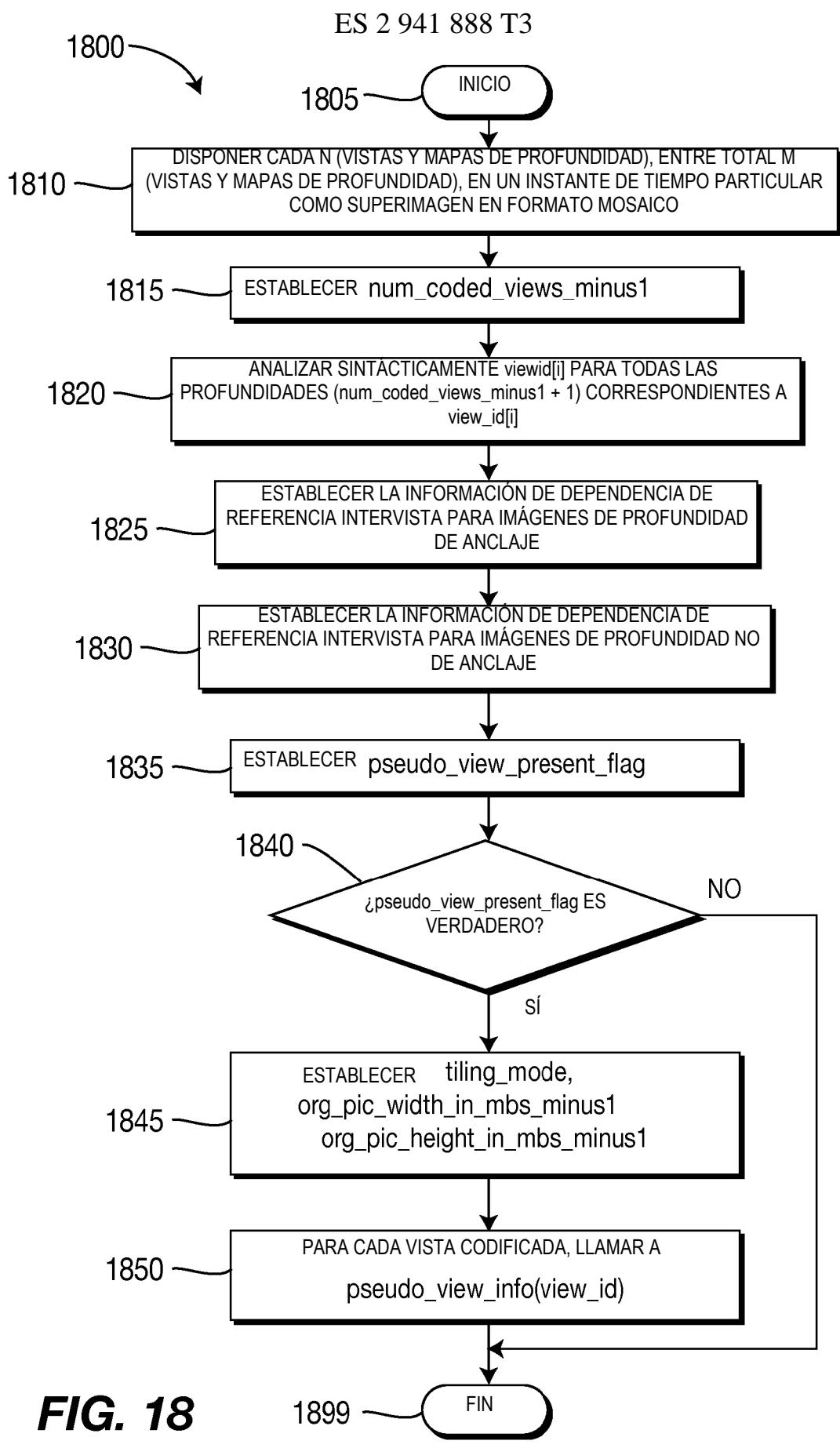
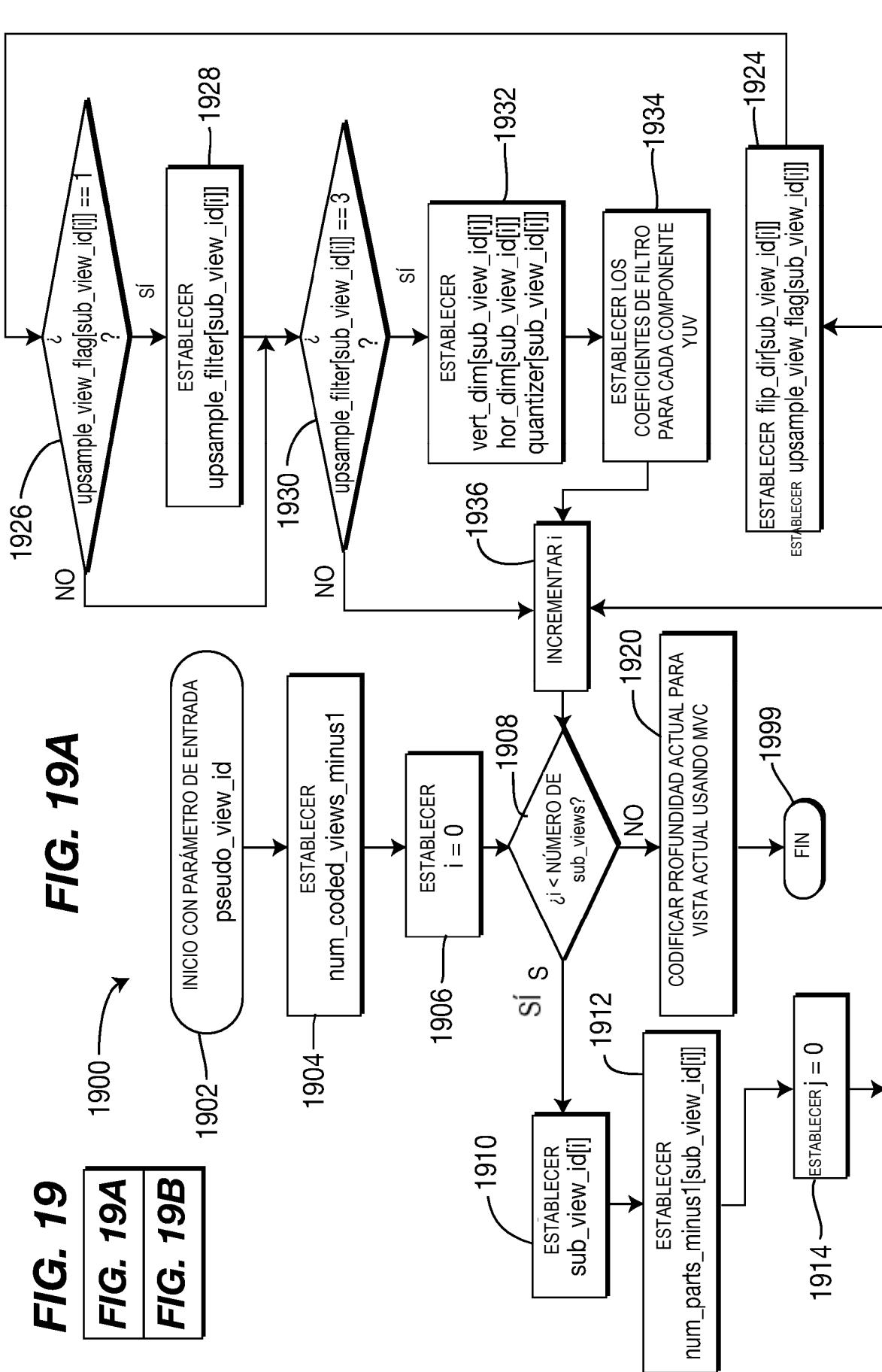
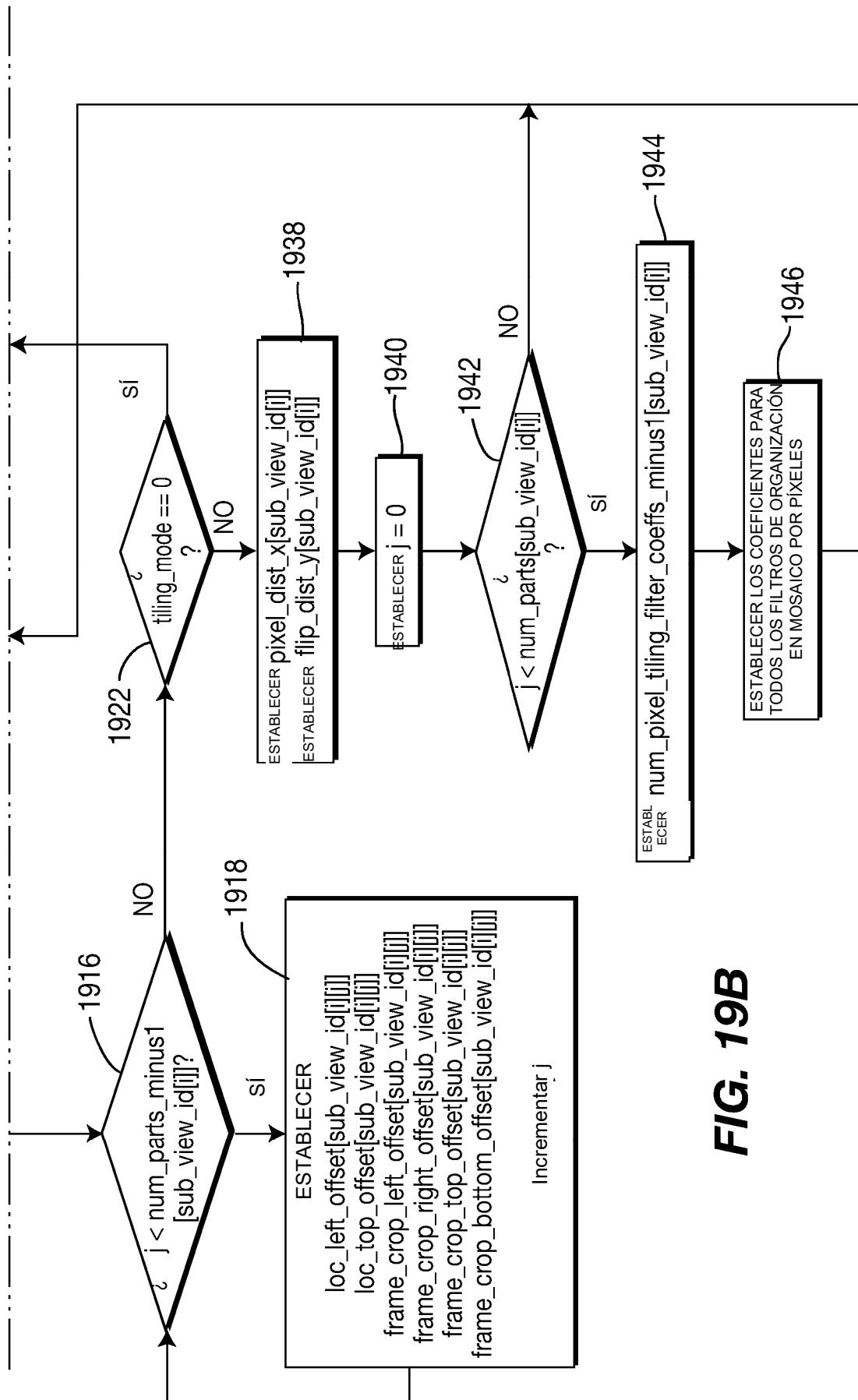


FIG. 18

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

**FIG. 19B**

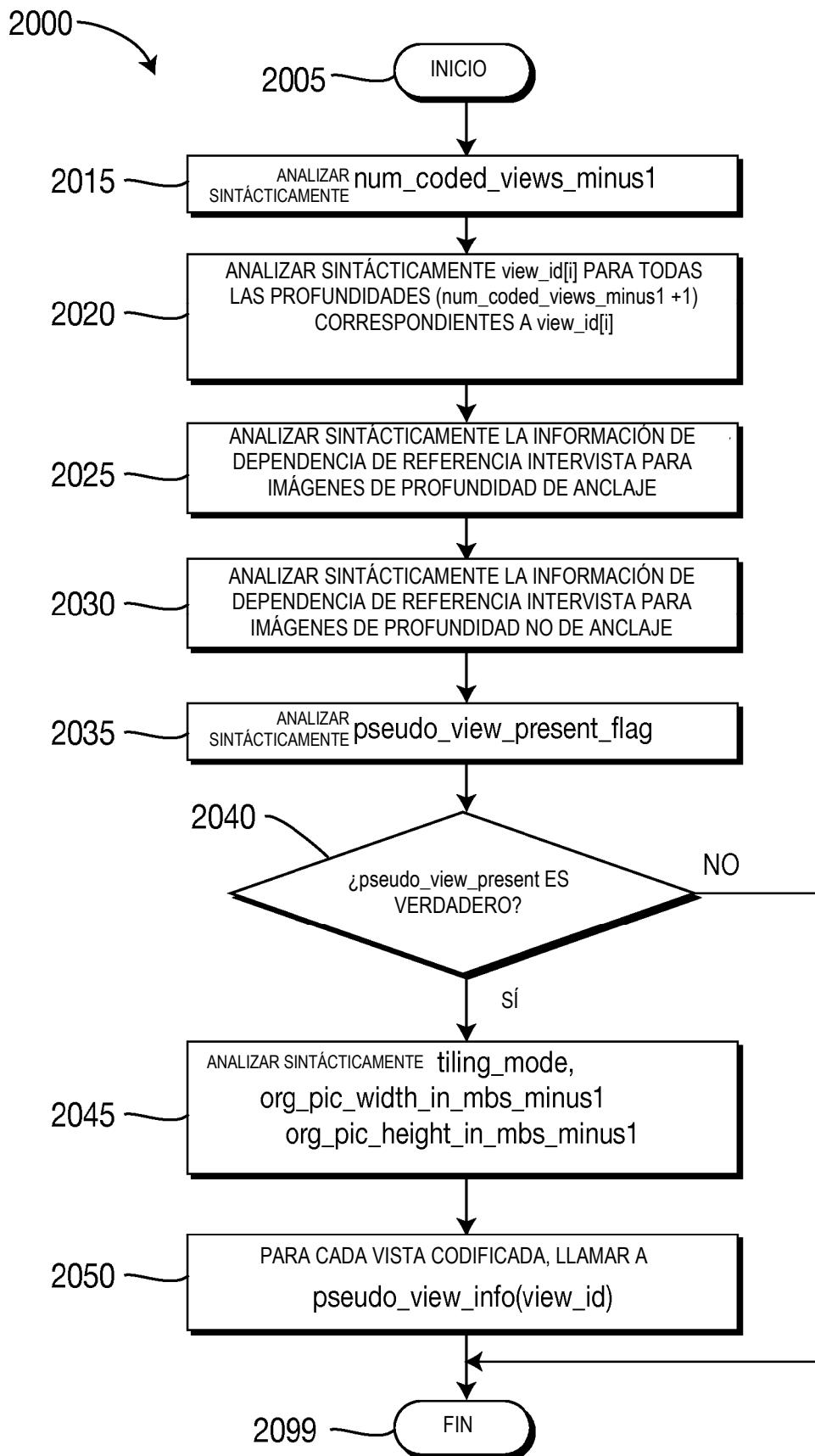
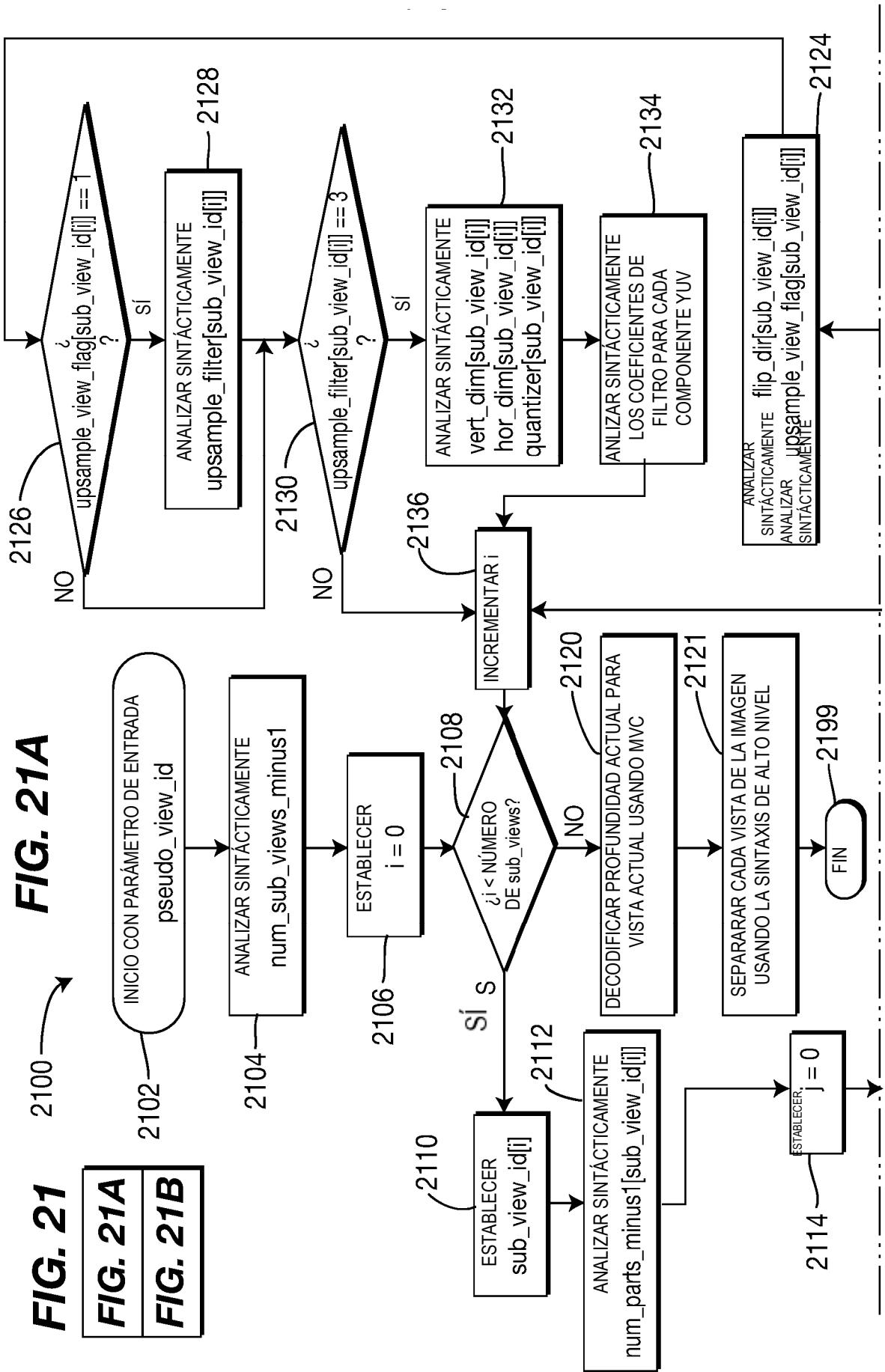
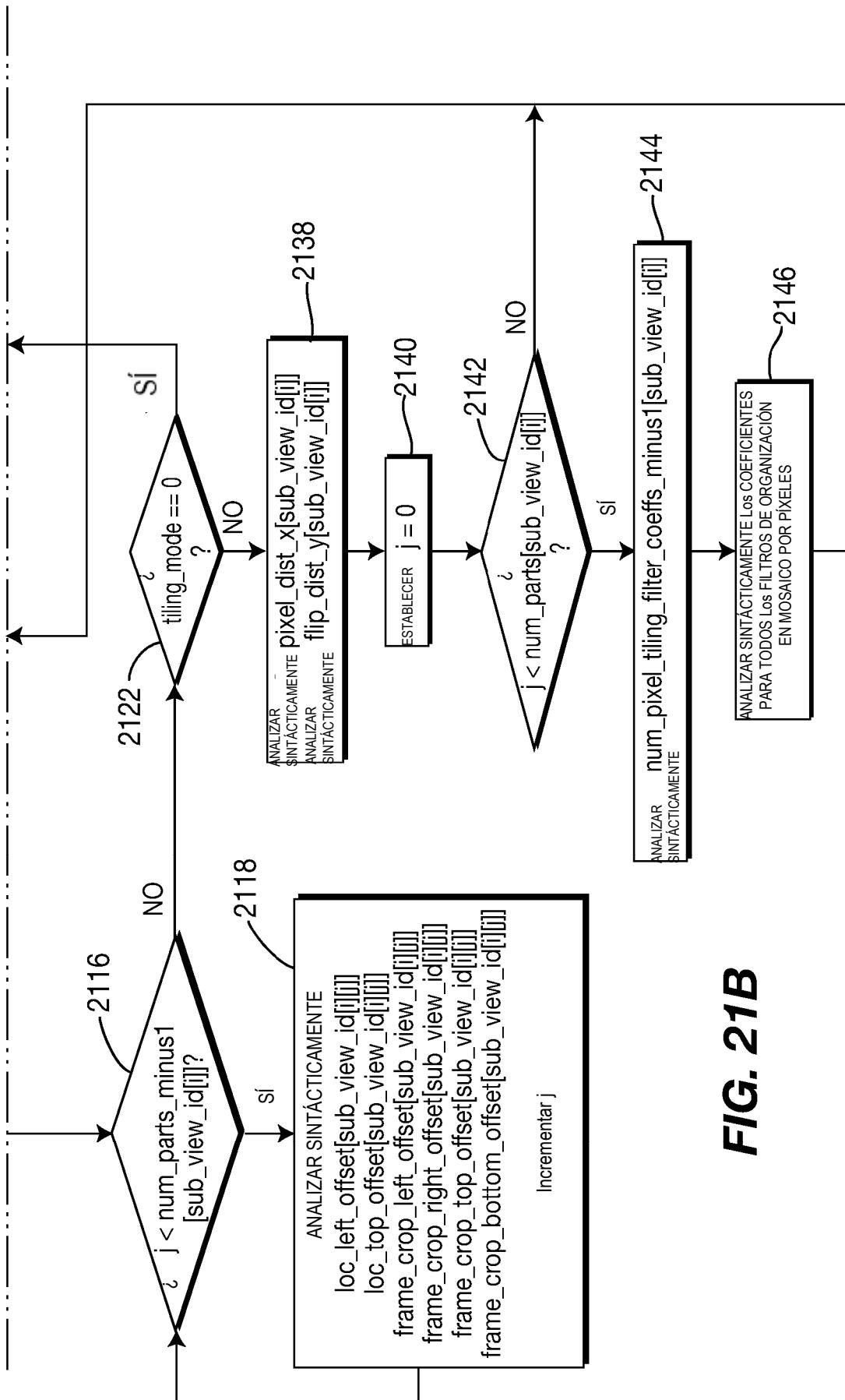
**FIG. 20**

FIG. 21A**FIG. 21B**

**FIG. 21B**

