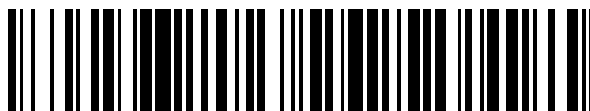


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 675 164**

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 (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **11.04.2008 E 15153812 (1)**

97 Fecha y número de publicación de la concesión europea: **16.05.2018 EP 2887671**

54 Título: **Presentación en mosaico en la codificación y descodificación 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:
09.07.2018

73 Titular/es:

DOLBY INTERNATIONAL AB (100.0%)
Apollo Building, 3E, Herikerbergweg 1-35
1101 CN Amsterdam Zuidoost, NL

72 Inventor/es:

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

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 675 164 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

Presentación en mosaico en la codificación y descodificación de vídeo

5 Campo técnico

Los presentes principios se refieren, en general, a la codificación y/o descodificación de vídeo.

Antecedentes

10 Los fabricantes de dispositivos de visualización de vídeo pueden usar un marco de disposición o de presentación en mosaico de diferentes vistas en un único cuadro. Las vistas pueden extraerse después de sus respectivas ubicaciones y renderizarse.

15 El documento EP 1 581 003 A1 describe un sistema para supervisar una pluralidad de señales de vídeo en una red de vídeo interna, tal como un entorno de grabación de radiodifusión, o una red de cámaras de seguridad. El sistema comprende un monitor de imágenes que presenta una pantalla en la que se muestra una imagen de vídeo en mosaico que contiene las señales de vídeo de una pluralidad de cámaras o de otros dispositivos de vídeo. La selección de una señal de vídeo de la imagen en mosaico puede realizarse para mostrar una versión a pantalla completa de esa señal.

20 El documento de MEESEN J ET AL: "*Content browsing and semantic context viewing through JPEG 2000-based scalable video summary*", IEE PROCEEDINGS: VISION, IMAGE AND SIGNAL PROCESSING, INSTITUTION OF ELECTRICAL ENGINEERS, GB, vol. 153, n.º 3, 8 de junio de 2006 (08/06/2006), páginas 274 a 283, describe un procedimiento y una plataforma de software para explorar de manera remota e interactiva un resumen de largas secuencias de vídeo, así como para revelar los enlaces semánticos entre tomas y escenas en su contexto temporal.

30 El documento US 5.915.091 describe un sistema de colaboración multimedia que integra diferentes redes en tiempo real y asincrónicas (las primeras para audio y vídeo en tiempo real, y las segundas para señales de control y datos de texto, de gráficos y de otro tipo), de manera que puede hacerse funcionar entre diferentes plataformas de sistemas operativos de ordenadores y redes y que se acerca en gran medida a la experiencia de una colaboración cara a cara, a la vez que se libera a los participantes de las limitaciones de tiempo y distancia.

35 El documento de EMIN MARTINIAN ET AL: "*Extensions of H.264/AVC for Multiview Video Compression*", IMAGE PROCESSING, 2006 IEEE INTERNATIONAL CONFERENCE ON, IEEE, PI, 1 de octubre de 2006 (01/10/2006), páginas 2981 a 2984, considera una compresión de vídeo con múltiples vistas: problema en chip de comprimir conjuntamente múltiples vistas de una escena grabada con diferentes cámaras. Para aprovechar la correlación en chip entre las vistas, Martinian *et al.* proponen usar una predicción de vistas y una síntesis de vistas compensadas en disparidad, y describen cómo estas características pueden implementarse ampliando la norma de compresión H.264/AVC.

Resumen

45 La presente invención se define mediante procedimientos según las reivindicaciones 1 y 7, mediante aparatos según las reivindicaciones 3 y 9, así como mediante medios legibles por ordenador según las reivindicaciones 5 y 11. Formas de realización preferidas se definen mediante las reivindicaciones dependientes.

50 Según un ejemplo general, se accede a una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen. Se accede a información que indica cómo están combinadas las múltiples imágenes en la imagen de vídeo accedida. La imagen de vídeo se descodifica para proporcionar una representación descodificada de las múltiples imágenes combinadas. La información accedida y la imagen de vídeo descodificada se proporcionan como salida.

55 Según otro aspecto general, se genera información que indica cómo múltiples imágenes incluidas en una imagen de vídeo se combinan en una única imagen. 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.

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

65 Según otro ejemplo general, se accede a una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen. Se accede a información que indica cómo están combinadas las múltiples imágenes en la imagen de vídeo accedida. La imagen de vídeo se descodifica para proporcionar una representación descodificada de al menos

una de las múltiples imágenes. La información accedida y la representación descodificada se proporcionan como salida.

5 Según otro ejemplo general, se accede a una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen. Se accede a información que indica cómo están combinadas las múltiples imágenes en la imagen de vídeo accedida. La imagen de vídeo se descodifica para proporcionar una representación descodificada de las múltiples imágenes combinadas. Se recibe una entrada de usuario que selecciona al menos una de las múltiples imágenes para su visualización. Se proporciona una salida descodificada de la al menos una imagen seleccionada, donde la salida descodificada se proporciona en función de la información accedida, la representación descodificada y la entrada de usuario.

15 Los detalles de una o más implementaciones se exponen en los dibujos adjuntos y en la siguiente descripción. Incluso si se describen de una manera particular, debe quedar claro que las implementaciones pueden configurarse o realizarse de varias maneras. Por ejemplo, una implementación puede realizarse como un procedimiento, o realizarse como un aparato configurado para realizar un conjunto de operaciones, o realizarse como un aparato que almacena instrucciones para realizar un conjunto de operaciones, o representarse como una señal. Otros aspectos y características resultarán evidentes a partir de la siguiente descripción detallada considerada junto con los dibujos adjuntos y las reivindicaciones.

20 Breve descripción de los dibujos

La FIG. 1 es un diagrama que muestra un ejemplo de cuatro vistas presentadas en mosaico en un único cuadro.

La FIG. 2 es un diagrama que muestra un ejemplo de cuatro vistas giradas y presentadas en mosaico en un único cuadro.

25 La FIG. 3 muestra un diagrama de bloques de un codificador de vídeo al que pueden aplicarse los presentes principios, según una forma de realización de los presentes principios.

La FIG. 4 muestra un diagrama de bloques de un descodificador de vídeo al que pueden aplicarse los presentes principios, según una forma de realización de los presentes principios.

30 La FIG. 5 es un diagrama de flujo de un procedimiento para codificar imágenes para una pluralidad de vistas usando la norma MPEG-4 AVC, según una forma de realización de los presentes principios.

La FIG. 6 es un diagrama de flujo de un procedimiento para descodificar imágenes para una pluralidad de vistas usando la norma MPEG-4 AVC, según una forma de realización de los presentes principios.

La FIG. 7 es un diagrama de flujo de un procedimiento para codificar imágenes para una pluralidad de vistas y profundidades usando la norma MPEG-4 AVC, según una forma de realización de los presentes principios.

35 La FIG. 8 es un diagrama de flujo de un procedimiento para descodificar imágenes para una pluralidad de vistas y profundidades usando la norma MPEG-4 AVC, según una forma de realización de los presentes principios.

La FIG. 9 es un diagrama que muestra un ejemplo de una señal de profundidad, según una forma de realización de los presentes principios.

40 La FIG. 10 es un diagrama que muestra un ejemplo de una señal de profundidad añadida como un mosaico, según una forma de realización de los presentes principios.

La FIG. 11 es un diagrama que muestra un ejemplo de 5 vistas presentadas en mosaico en un único cuadro, según una forma de realización de los presentes principios.

La FIG. 12 es un diagrama de bloques de un codificador de codificación de vídeo de múltiples vistas (MVC) al que pueden aplicarse los presentes principios, según una forma de realización de los presentes principios.

45 La FIG. 13 es un diagrama de bloques de un descodificador de codificación de vídeo de múltiples vistas (MVC) al que pueden aplicarse los presentes principios, según una forma de realización de los presentes principios.

La FIG. 14 es un diagrama de flujo de un procedimiento para procesar imágenes para una pluralidad de vistas como preparación a la codificación de las imágenes usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC, según una forma de realización de los presentes principios.

50 La FIG. 15 es un diagrama de flujo de un procedimiento para codificar imágenes para una pluralidad de vistas usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC, según una forma de realización de los presentes principios.

55 La FIG. 16 es un diagrama de flujo de un procedimiento para procesar imágenes para una pluralidad de vistas como preparación a la descodificación de las imágenes usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC, según una forma de realización de los presentes principios.

La FIG. 17 es un diagrama de flujo de un procedimiento para descodificar imágenes para una pluralidad de vistas usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC, según una forma de realización de los presentes principios.

5 La FIG. 18 es un diagrama de flujo de un procedimiento para procesar imágenes para una pluralidad de vistas y profundidades como preparación a la codificación de las imágenes usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC, según una forma de realización de los presentes principios.

La FIG. 19 es un diagrama de flujo de un procedimiento para codificar imágenes para una pluralidad de vistas y profundidades usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC, según una forma de realización de los presentes principios.

10 La FIG. 20 es un diagrama de flujo de un procedimiento para procesar imágenes para una pluralidad de vistas y profundidades como preparación a la descodificación de las imágenes usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC, según una forma de realización de los presentes principios.

15 La FIG. 21 es un diagrama de flujo de un procedimiento para descodificar imágenes para una pluralidad de vistas y profundidades usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC, según una forma de realización de los presentes principios.

La FIG. 22 es un diagrama que muestra ejemplos de presentación en mosaico a nivel de píxel, según una forma de realización de los presentes principios.

La FIG. 23 muestra un diagrama de bloques de un dispositivo de procesamiento de vídeo al que pueden aplicarse los presentes principios, según una forma de realización de los presentes principios.

20 Descripción detallada

25 Varias implementaciones están dirigidas a procedimientos y aparatos para la presentación de vistas en mosaico en la codificación y la descodificación de vídeo. Por tanto, debe apreciarse que los expertos en la técnica podrán concebir varias disposiciones que, aunque no se describan o muestren de manera explícita en el presente documento, representan los presentes principios.

30 Todos los ejemplos y el lenguaje condicional incluidos en el presente documento tienen fines pedagógicos para ayudar al lector a entender los presentes principios y los conceptos aportados por el/los inventor(es) para mejorar la técnica, y no debe considerarse que están limitados a tales ejemplos y condiciones expuestos de manera específica.

35 Además, todas las afirmaciones del presente documento que enumeran principios, aspectos y formas de realización de los presentes principios, así como ejemplos específicos de los mismos, abarcan equivalencias estructurales y funcionales de los mismos. Además, tales equivalencias incluyen equivalencias actualmente conocidas, así como equivalencias desarrolladas en el futuro, es decir, cualquier elemento desarrollado que lleve a cabo la misma función, independientemente de la estructura.

40 Por tanto, por ejemplo, los expertos en la técnica apreciarán que los diagramas de bloques presentados en el presente documento representan vistas conceptuales de sistemas de circuitos ilustrativos que representan los presentes principios. Asimismo, debe apreciarse que cualquier organigrama, diagrama de flujo, diagrama de transición de estados, seudocódigo, etc., representan varios procesos que pueden representarse sustancialmente en medios legibles por ordenador y ejecutarse por tanto mediante un ordenador o procesador, independientemente de que tal ordenador o procesador se muestre explícitamente o no.

45 Las funciones de los diversos elementos mostrados en las figuras pueden proporcionarse mediante el uso de hardware dedicado así como hardware capaz de ejecutar software junto con software apropiado. Cuando se proporcionan por un procesador, las funciones pueden proporcionarse por un único procesador dedicado, por un único procesador compartido o por una pluralidad de procesadores individuales, algunos de los cuales pueden compartirse. Además, no debe considerarse que el uso explícito del término "procesador" o "controlador" se refiere exclusivamente a hardware capaz de ejecutar software, sino que puede incluir implícitamente, sin limitación, hardware de procesador de señales digitales (DSP), memoria de solo lectura (ROM) para almacenar software, memoria de acceso aleatorio (RAM) y almacenamiento no volátil.

55 También puede incluirse otro hardware, convencional y/o personalizado. Asimismo, cualquier conmutador mostrado en las figuras es solamente conceptual. Su función puede llevarse a cabo mediante operaciones de lógica de programa, mediante lógica dedicada, mediante la interacción de control de programa y lógica dedicada, o incluso manualmente, donde la técnica particular más apropiada según el contexto puede ser seleccionada por el implementador.

60 En las reivindicaciones de la invención, cualquier elemento expresado como un medio para llevar a cabo una función especificada pretende abarcar cualquier manera de llevar a cabo esa función incluido, por ejemplo, a) una combinación de elementos de circuito que lleve a cabo esa función, o b) software de cualquier tipo, incluido, por

tanto, firmware, microcódigo, etc., en combinación con un sistema de circuitos apropiado para ejecutar ese software para llevar a cabo la función. Los presentes principios definidos por tales reivindicaciones se basan en el hecho de que las funcionalidades proporcionadas por los diversos medios enumerados se combinan y se utilizan de la manera indicada en las reivindicaciones. Por tanto, se considera que cualquier medio que pueda proporcionar estas

5 funcionalidades es equivalente a los mostrados en el presente documento.

La referencia en la memoria descriptiva a "una forma de realización" (o "una implementación") de los presentes principios significa que una propiedad, estructura, característica, etc. particulares descritas en relación con la forma de realización está incluida en al menos una forma de realización de los presentes principios. Por tanto, no todas las

10 veces que aparece la expresión "en una forma de realización" en diversos puntos de la memoria descriptiva se hace referencia necesariamente a la misma forma de realización.

Debe apreciarse 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 abarcar la selección de solo la primera opción enumerada (A), de solo la selección de la segunda opción enumerada (B) o la selección de ambas opciones (A y B). Como otro ejemplo, en los casos de "A, B y/o C" y de "al menos uno de A, B y C", tal expresión tiene como objetivo abarcar la selección de solo la primera opción enumerada (A), la selección de solo la segunda opción enumerada (B), la selección de solo la tercera opción enumerada (C), la selección de solo la primera y la segunda opción enumeradas (A y B), la selección de solo la primera y la tercera opción enumeradas (A y C), la selección de solo la segunda y la tercera opción enumeradas (B y C), o la selección de las tres opciones (A, B y C). Esto puede ampliarse con respecto al número total de elementos enumerados, como les resultará fácilmente evidente a los expertos en la técnica.

15

20

Además, debe apreciarse que aunque una o más formas de realización de los presentes principios se describen en el presente documento con respecto a la norma MPEG-4 AVC, los presentes principios no están limitados solamente a esta norma y, por tanto, pueden utilizarse con respecto a otras normas, recomendaciones y extensiones de la misma, en particular a normas, recomendaciones y extensiones de codificación de vídeo de la misma, incluidas extensiones de la norma MPEG-4 AVC.

25

Además, debe apreciarse que aunque otra u otras formas de realización de los presentes principios se describen en el presente documento con respecto a la extensión de codificación de vídeo de múltiples vistas de la norma MPEG-4 AVC, los presentes principios no están limitados solamente a esta extensión y/o a esta norma y, por tanto, pueden utilizarse con respecto a otras normas, recomendaciones y extensiones de codificación de vídeo de la misma relacionadas con la codificación de vídeo de múltiples vistas. La codificación de vídeo de múltiples vistas (MVC) es el marco de compresión para la codificación de secuencias de múltiples vistas. Una secuencia de codificación de vídeo de múltiples vistas (MVC) es un conjunto de dos o más secuencias de vídeo que capturan la misma escena desde un punto de vista diferente.

30

35

Además, debe apreciarse que aunque otra u otras formas de realización de los presentes principios descritas en el presente documento usan información de profundidad con respecto al contenido de vídeo, los presentes principios no están limitados a tales formas de realización y, por tanto, pueden implementarse otras formas de realización que no usan información de profundidad.

40

Además, como se usa en el presente documento, "sintaxis de alto nivel" se refiere a la sintaxis presente en el flujo de bits que reside de manera jerárquica encima de la capa de macrobloques. Por ejemplo, sintaxis de alto nivel, tal y como se usa en el presente documento, puede referirse a, pero no está limitado a, sintaxis a nivel de cabecera de segmento, nivel de información de mejora complementaria (SEI), nivel de conjunto de parámetros de imagen (PPS), nivel de conjunto de parámetros de secuencia (SPS), conjunto de parámetros de vista (VPS) y nivel de cabecera de unidad de capa de abstracción de red (NAL).

45

En la implementación actual de codificación de múltiples vídeos (MVC) basada en la recomendación H.264 de la Unión Internacional de Telecomunicaciones, Sector de Telecomunicaciones (ITU-T) / norma del Grupo 4 de Expertos en Imágenes en Movimiento (MPEG-4), parte 10 (codificación de vídeo avanzada (AVC)), de la Organización Internacional de Normalización/Comisión Electrotécnica Internacional (ISO/IEC) (lo que se denomina en lo sucesivo "norma MPEG-4 AVC), el software de referencia consigue una predicción de múltiples vistas codificando cada vista con un único codificador y teniendo en cuenta las referencias de vistas cruzadas. Cada vista es codificada como un flujo de bits diferente por el codificador en su resolución original y, después, todos los flujos de bits se combinan para formar un único flujo de bits que posteriormente se descodifica. Cada vista produce una salida descodificada YUV diferente.

50

55

Otro enfoque para la predicción de múltiples vistas implica agrupar un conjunto de vistas en seudovistas. En un ejemplo de este enfoque, las imágenes de cada N vistas de un total de M vistas (muestreadas al mismo tiempo) pueden presentarse en mosaico en un cuadro más grande o en supercuadro con un posible muestreo descendente u otras operaciones. Haciendo referencia a la FIG. 1, un ejemplo de cuatro vistas presentadas en mosaico en un único cuadro se indica de manera genérica mediante el número de referencia 100. Las cuatro vistas están en su orientación normal.

60

65

- Haciendo referencia a la FIG. 2, un ejemplo de cuatro vistas giradas y presentadas en mosaico en un único cuadro se indica de manera genérica mediante el número de referencia 200. La vista superior izquierda está en su orientación normal. La vista superior derecha está girada en horizontal. La vista inferior izquierda está girada en vertical. La vista inferior derecha está girada en vertical y en horizontal. Por tanto, si hay cuatro vistas, entonces una imagen de cada vista está dispuesta en un supercuadro como un mosaico. Esto da como resultado una única secuencia de entrada no codificada con una gran resolución.
- Como alternativa, puede muestrearse la imagen de manera descendente para producir una menor resolución. Por tanto, se crean múltiples secuencias, donde cada una incluye diferentes vistas que se presentan en mosaico conjuntamente. Cada secuencia de este tipo forma entonces una seudovista, donde cada seudovista incluye N vistas en mosaico diferentes. La FIG. 1 muestra una seudovista, y la FIG. 2 muestra otra seudovista. Estas seudovistas pueden codificarse después usando normas de codificación de vídeo existentes, tales como la norma ISO/IEC MPEG-2 y la norma MPEG-4 AVC.
- Otro enfoque adicional para la predicción de múltiples vistas simplemente implica codificar las diferentes vistas de manera independiente usando una nueva norma y, después, descodificar y presentar en mosaico las vistas según requiera el reproductor.
- Además, en otro enfoque, las vistas también pueden presentarse en mosaico de una manera basada en píxeles. Por ejemplo, en una supervista que está formada por cuatro vistas, el 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.
- Muchos fabricantes de dispositivos de visualización usan este marco de disposición o de presentación en mosaico de diferentes vistas en un único cuadro, y después extraen las vistas desde sus respectivas ubicaciones y las renderizan. En tales casos, no hay una manera estándar de determinar si el flujo de bits tiene tal propiedad. Por tanto, si un sistema usa el procedimiento de presentar en mosaico imágenes de diferentes vistas en un gran cuadro, entonces el procedimiento de extracción de las diferentes vistas es obligatorio.
- Sin embargo, no hay una manera estándar de determinar si el flujo de bits tiene tal propiedad. Se propone una sintaxis de alto nivel para facilitar al renderizador o al reproductor que extraiga tal información con el fin de ayudar en la visualización o en otro procesamiento posterior. También es posible que las subimágenes tengan diferentes resoluciones y puede ser necesario cierto muestreo ascendente para renderizar finalmente la vista. El usuario puede desear que el procedimiento de muestreo ascendente también esté indicado en la sintaxis de alto nivel. Además, también pueden transmitirse parámetros para cambiar el foco de profundidad.
- En una forma de realización se propone un nuevo mensaje de información de mejora complementaria (SEI) para señalar información de múltiples vistas en un flujo de bits compatible con la norma MPEG-4 AVC, donde cada imagen incluye subimágenes que pertenecen a una vista diferente. La forma de realización tiene como objetivo, por ejemplo, visualizar de manera fácil y cómoda flujos de vídeo de múltiples vistas en monitores tridimensionales (3D) que pueden usar este marco. El concepto puede ampliarse a otras normas de codificación de vídeo y recomendaciones que señalan tal información usando una sintaxis de alto nivel.
- Además, en una forma de realización, se propone un procedimiento de señalización que indica cómo disponer vistas antes de que se envíen al codificador y/o al descodificador de vídeo de múltiples vistas. De manera ventajosa, la forma de realización puede dar lugar a una implementación simplificada de la codificación de múltiples vistas, y puede mejorar la eficacia de codificación. Ciertas vistas pueden juntarse y formar una seudovista o supervista y, después, la supervista presentada en mosaico se trata como una vista normal mediante un codificador y/o un descodificador de vídeo común de múltiples vistas, por ejemplo, en lo que respecta a la implementación de la codificación de vídeo de múltiples vistas basada en la norma MPEG-4 AVC actual. Se propone un nuevo indicador en la extensión del conjunto de parámetros de secuencia (SPS) de la codificación de vídeo de múltiples vistas para señalar el uso de la técnica de seudovistas. La forma de realización tiene como objetivo visualizar de manera fácil y cómoda flujos de vídeo de múltiples vistas en monitores 3D que pueden usar este marco.
- Codificación/descodificación usando una norma/recomendación de codificación/descodificación de vídeo de una sola vista
- En la implementación actual de codificación de múltiples vídeos (MVC) basada en la recomendación H.264 de la Unión Internacional de Telecomunicaciones, Sector de Telecomunicaciones (ITU-T) / norma del Grupo 4 de Expertos en Imágenes en Movimiento (MPEG-4), parte 10 (codificación de vídeo avanzada (AVC)), de la Organización Internacional de Normalización/Comisión Electrotécnica Internacional (ISO/IEC) (lo que se denomina en lo sucesivo "norma MPEG-4 AVC), el software de referencia consigue una predicción de múltiples vistas codificando cada vista con un único codificador y teniendo en cuenta las referencias de vistas cruzadas. Cada vista es codificada como un flujo de bits diferente por el codificador en su resolución original y, después, todos los flujos de bits se combinan para formar un único flujo de bits que posteriormente se descodifica. Cada vista produce una salida descodificada YUV diferente.

Otro enfoque para una predicción de múltiples vistas implica presentar en mosaico las imágenes de cada vista (muestreadas al mismo tiempo) en un cuadro más grande o un supercuadro con una posible operación de muestreo descendente. Haciendo referencia a la FIG. 1, un ejemplo de cuatro vistas presentadas en mosaico en un único cuadro se indica de manera genérica mediante el número de referencia 100. Haciendo referencia a la FIG. 2, un ejemplo de cuatro vistas giradas y presentadas en mosaico en un único cuadro se indica de manera genérica mediante el número de referencia 200. Por tanto, si hay cuatro vistas, entonces una imagen de cada vista está dispuesta en un supercuadro como un mosaico. Esto da como resultado una única secuencia de entrada no codificada con una gran resolución. Esta señal puede codificarse después usando normas de codificación de vídeo existentes, tales como la norma ISO/IEC MPEG-2 y la norma MPEG-4 AVC.

Otro enfoque adicional para la predicción de múltiples vistas simplemente incluye codificar las diferentes vistas de manera independiente usando una nueva norma y, después, descodificar y presentar en mosaico las vistas según requiera el reproductor.

Muchos fabricantes de dispositivos de visualización usan este marco de disposición o de presentación en mosaico de diferentes vistas en un único cuadro, y después extraen las vistas desde sus respectivas ubicaciones y las renderizan. En tales casos, no hay una manera estándar de determinar si el flujo de bits tiene tal propiedad. Por tanto, si un sistema usa el procedimiento de presentar en mosaico imágenes de diferentes vistas en un gran cuadro, entonces el procedimiento de extracción de las diferentes vistas es obligatorio.

Haciendo referencia a la FIG. 3, un codificador de vídeo que puede realizar una codificación de vídeo según la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 300.

El codificador de vídeo 300 incluye una memoria intermedia de ordenación de cuadros 310 que presenta una salida en comunicación de señales con una entrada sin inversión de un combinador 385. Una salida del combinador 385 está conectada en comunicación de señales a una primera entrada de un transformador y cuantificador 325. Una salida del transformador y cuantificador 325 está conectada en comunicación de señal a una primera entrada de un codificador de entropía 345 y a una primera entrada de un transformador inverso y cuantificador inverso 350. Una salida del codificador de entropía 345 está conectada en comunicación de señales a una primera entrada sin inversión de un combinador 390. Una salida del combinador 390 está conectada en comunicación de señales a una primera entrada de una memoria intermedia de salida 335.

Una primera salida de un controlador de codificador 305 está conectada en comunicación de señales a una segunda entrada de la memoria intermedia de ordenación de cuadros 310, a una segunda entrada del transformador inverso y cuantificador inverso 350, a una entrada de un módulo de decisión de tipo de imagen 315, a una entrada de un módulo de decisión de tipo de macrobloque (tipo MB) 320, a una segunda entrada de un módulo de intrapredicción 360, a una segunda entrada de un filtro de desbloqueo 365, a una primera entrada de un compensador de movimiento 370, a una primera entrada de un estimador de movimiento 375 y a una segunda entrada de una memoria intermedia de imágenes de referencia 380.

Una segunda salida del controlador de codificador 305 está conectada en comunicación de señales a una primera entrada de un insertador de información de mejora complementaria (SEI) 330, a una segunda entrada del transformador y cuantificador 325, a una segunda entrada del codificador de entropía 345, a una segunda entrada de la memoria intermedia de salida 335, y a una entrada del insertador de conjunto de parámetros de secuencia (SPS) y de conjunto de parámetros de imagen (PPS) 340.

Una primera salida del módulo de decisión de tipo de imagen 315 está conectada en comunicación de señales a una tercera entrada de una memoria intermedia de ordenación de cuadros 310. Una segunda salida del módulo de decisión de tipo de imagen 315 está conectada en comunicación de señales a una segunda entrada de un módulo de decisión de tipo de macrobloque 320.

Una salida del insertador de conjunto de parámetros de secuencia (SPS) y de conjunto de parámetros de imagen (PPS) 340 está conectado en comunicación de señales a una tercera entrada sin inversión del combinador 390. Una salida del insertador de SEI 330 está conectada en comunicación de señales a una segunda entrada sin inversión del combinador 390.

Una salida del cuantificador inverso y transformador inverso 350 está conectada en comunicación de señales a una primera entrada sin inversión de un combinador 319. Una salida del combinador 319 está conectada en comunicación de señales a una primera entrada del módulo de intrapredicción 360 y a una primera entrada del filtro de desbloqueo 365. Una salida del filtro de desbloqueo 365 está conectada en comunicación de señales a una primera entrada de una memoria intermedia de imágenes de referencia 380. Una salida de la memoria intermedia de imágenes de referencia 380 está conectada en comunicación de señales a una segunda entrada del estimador de movimiento 375 y a una primera entrada de un compensador de movimiento 370. Una primera salida del estimador de movimiento 375 está conectada en comunicación de señales a una segunda entrada del compensador de movimiento 370. Una segunda salida del estimador de movimiento 375 está conectada en comunicación de señales a una tercera entrada del codificador de entropía 345.

Una salida del compensador de movimiento 370 está conectada en comunicación de señales a una primera entrada de un conmutador 397. Una salida del módulo de intrapredicción 360 está conectada en comunicación de señales a una segunda entrada del conmutador 397. Una salida del módulo de decisión de tipo de macrobloque 320 está conectada en comunicación de señales a una tercera entrada del conmutador 397 para proporcionar una entrada de control al conmutador 397. La tercera entrada del conmutador 397 determina si la entrada de "datos" del conmutador (en comparación con la unidad de control, es decir, la tercera entrada) va a proporcionarse o no por el compensador de movimiento 370 o el módulo de intrapredicción 360. La salida del conmutador 397 está conectada en comunicación de señales a una segunda entrada sin inversión del combinador 319 y a una entrada con inversión del combinador 385.

Entradas de la memoria intermedia de ordenación de cuadros 310 y del controlador de codificador 105 están disponibles como entradas del codificador 300 para recibir una imagen de entrada 301. Además, una entrada del insertador de información de mejora complementaria (SEI) 330 también está disponible como una entrada del codificador 300, para recibir metadatos. Una salida de la memoria intermedia de salida 335 está disponible como una salida del codificador 300, para proporcionar un flujo de bits.

Haciendo referencia a la FIG. 4, un descodificador de vídeo que puede realizar una descodificación de vídeo según la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 400.

El descodificador de vídeo 400 incluye una memoria intermedia de entrada 410 que presenta una salida conectada en comunicación de señales a una primera entrada del descodificador de entropía 445. Una primera salida del descodificador de entropía 445 está conectada en comunicación de señales a una primera entrada de un transformador inverso y cuantificador inverso 450. Una salida del transformador inverso y cuantificador inverso 450 está conectada en comunicación de señales a una segunda entrada sin inversión de un combinador 425. Una salida del combinador 425 está conectada en comunicación de señales a una segunda entrada de un filtro de desbloqueo 465 y a una primera entrada de un módulo de intrapredicción 460. Una segunda salida del filtro de desbloqueo 465 está conectada en comunicación de señales a una primera entrada de una memoria intermedia de imágenes de referencia 480. Una salida de la memoria intermedia de imágenes de referencia 480 está conectada en comunicación de señales a una segunda entrada de un compensador de movimiento 470.

Una segunda salida del descodificador de entropía 445 está conectada en comunicación de señales a una tercera entrada del compensador de movimiento 470 y a una primera entrada del filtro de desbloqueo 465. Una tercera salida del descodificador de entropía 445 está conectada en comunicación de señales a una entrada de un controlador de descodificador 405. Una primera salida del controlador de descodificador 405 está conectada en comunicación de señales a una segunda entrada del descodificador de entropía 445. Una segunda salida del controlador de descodificador 405 está conectada en comunicación de señales a una segunda entrada del transformador inverso y cuantificador inverso 450. Una tercera salida del controlador de descodificador 405 está conectada en comunicación de señales a una tercera entrada del filtro de desbloqueo 465. Una cuarta salida del controlador de descodificador 405 está conectada en comunicación de señales a una segunda entrada del módulo de intrapredicción 460, a una primera entrada del compensador de movimiento 470 y a una segunda entrada de la memoria intermedia de imágenes de referencia 480.

Una salida del compensador de movimiento 470 está conectada en comunicación de señales a una primera entrada de un conmutador 497. Una salida del módulo de intrapredicción 460 está conectada en comunicación de señales a una segunda entrada del conmutador 497. Una salida del conmutador 497 está conectada en comunicación de señales a una primera entrada sin inversión del combinador 425.

Una entrada de la memoria intermedia de entrada 410 está disponible como una entrada del descodificador 400, para recibir un flujo de bits de entrada. Una primera salida del filtro de desbloqueo 465 está disponible como una salida del descodificador 400, para proporcionar una imagen de salida.

Haciendo referencia a la FIG. 5, un procedimiento a modo de ejemplo para codificar imágenes para una pluralidad de vistas usando la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 500.

El procedimiento 500 incluye un bloque de inicio 502 que pasa el control a un bloque funcional 504. El bloque funcional 504 gestiona cada vista en una instancia de tiempo particular como una subimagen en formato de mosaico y pasa el control a un bloque funcional 506. El bloque funcional 506 fija un elemento sintáctico `num_coded_views_minus1` y pasa el control a un bloque funcional 508. El bloque funcional 508 fija elementos sintácticos `org_pic_width_in_mbs_minus1` y `org_pic_height_in_mbs_minus1`, y pasa el control a un bloque funcional 510. El bloque funcional 510 fija 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 inferior al número de vistas. Si es así, entonces el control se pasa a un bloque funcional 514. Si no, el control se pasa a un bloque funcional 524.

El bloque funcional 514 fija un elemento sintáctico `view_id[i]` y pasa el control a un bloque funcional 516. El bloque funcional 516 fija un elemento sintáctico `num_parts[view_id[i]]` y pasa el control a un bloque funcional 518. El bloque

funcional 518 fija 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 inferior al valor actual del elemento sintáctico num_parts[view_id[i]]. Si es así, entonces el control se pasa a un bloque funcional 522. Si no, el control se pasa a un bloque funcional 528.

5 El bloque funcional 522 fija los siguientes elementos sintácticos, incrementa la variable 'j' y después devuelve el 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 frame_crop_bottom_offset[view_id[i]][j].

10 El bloque funcional 528 fija un elemento sintáctico 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 sintáctico upsample_view_flag[view_id[i]] es igual a uno. Si es así, entonces el control se pasa a un bloque funcional 532. Si no, el control se pasa a un bloque de decisión 534.

15 El bloque funcional 532 fija un elemento sintáctico 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 sintáctico upsample_filter[view_id[i]] es igual a tres. Si es así, entonces el control se pasa a un bloque funcional 536. Si no, el control se pasa a un bloque funcional 20 540.

El bloque funcional 536 fija los siguientes elementos sintácticos y pasa el control a un bloque funcional 538: vert_dim[view_id[i]]; hor_dim[view_id[i]]; y quantizer[view_id[i]].

25 El bloque funcional 538 fija los coeficientes de filtro para cada componente YUV y pasa el control al bloque funcional 540.

El bloque funcional 540 incrementa la variable 'i' y devuelve el control al bloque de decisión 512.

30 El bloque funcional 524 escribe estos elementos sintácticos en al menos uno de un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), un mensaje de información de mejora complementaria (SEI), una cabecera de unidad de capa de abstracción de red (NAL) y una cabecera de segmento, y pasa el control a un bloque funcional 526. El bloque funcional 526 codifica cada imagen usando la norma MPEG-4 AVC u otro códec de vista única, y pasa el control a un bloque final 599.

35 Haciendo referencia a la FIG. 6, un procedimiento a modo de ejemplo para descodificar imágenes para una pluralidad de vistas usando la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 600.

40 El procedimiento 600 incluye un bloque de inicio 602 que pasa el control a un bloque funcional 604. El bloque funcional 604 analiza sintácticamente los siguientes elementos sintácticos de al menos uno de un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), un mensaje de información de mejora complementaria (SEI), una cabecera de unidad de capa de abstracción de red (NAL) y una cabecera de segmento, y pasa el control a un bloque funcional 606. El bloque funcional 606 analiza sintácticamente un elemento sintáctico

45 num_coded_views_minus1 y pasa el control a un bloque funcional 608. El bloque funcional 608 analiza sintácticamente elementos sintácticos org_pic_width_in_mbs_minus1 y org_pic_height_in_mbs_minus1, y pasa el control a un bloque funcional 610. El bloque funcional 610 fija 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 inferior al número de vistas. Si es así, entonces el control se pasa a un bloque funcional 614. Si no, el control se pasa a un bloque funcional 624.

50 El bloque funcional 614 analiza sintácticamente un elemento sintáctico view_id[i] y pasa el control a un bloque funcional 616. El bloque funcional 616 analiza sintácticamente un elemento sintáctico num_parts_minus1[view_id[i]] y pasa el control a un bloque funcional 618. El bloque funcional 618 fija una variable j igual a cero y pasa el control a un bloque de decisión 620. El bloque de decisión 620 determina si el valor actual de la variable j es inferior al valor actual del elemento sintáctico num_parts[view_id[i]]. Si es así, entonces el control se pasa a un bloque funcional 622. Si no, el control se pasa a un bloque funcional 628.

El bloque funcional 622 analiza sintácticamente los siguientes elementos sintácticos, incrementa la variable 'j' y después devuelve el control al bloque de decisión 620: depth_flag[view_id[i]][j]; flip_dir[view_id[i]][j]; loc_left_offset[view_id[i]][j]; loc_top_offset[view_id[i]][j]; frame_crop_left_offset[view_id[i]][j]; frame_crop_right_offset[view_id[i]][j]; frame_crop_top_offset[view_id[i]][j]; y frame_crop_bottom_offset[view_id[i]][j].

60 El bloque funcional 628 analiza sintácticamente un elemento sintáctico 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 sintáctico upsample_view_flag[view_id[i]] es igual a uno. Si es así, entonces el control se pasa a un bloque funcional 632. Si no, el control se pasa a un bloque de decisión 634.

- El bloque funcional 632 analiza sintácticamente un elemento sintáctico `upsample_filter[view_id[i]]` y pasa el control al bloque de decisión 634.
- 5 El bloque de decisión 634 determina si el valor actual del elemento sintáctico `upsample_filter[view_id[i]]` es igual a tres. Si es así, entonces el control se pasa a un bloque funcional 636. Si no, el control se pasa a un bloque funcional 640.
- 10 El bloque funcional 636 analiza sintácticamente los siguientes elementos sintácticos y pasa el control a un bloque funcional 638: `vert_dim[view_id[i]]`; `hor_dim[view_id[i]]`; y `quantizer[view_id[i]]`.
- El bloque funcional 638 analiza sintácticamente los coeficientes de filtro para cada componente YUV y pasa el control al bloque funcional 640.
- 15 El bloque funcional 640 incrementa la variable 'i' y devuelve el control al bloque de decisión 612.
- El bloque funcional 624 descodifica cada imagen usando la norma MPEG-4 AVC u otro códec de vista única, y pasa el control a un bloque funcional 626. El bloque funcional 626 separa cada vista de la imagen usando la sintaxis de alto nivel y pasa el control a un bloque final 699.
- 20 Haciendo referencia a la FIG. 7, un procedimiento a modo de ejemplo para codificar imágenes para una pluralidad de vistas y profundidades usando la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 700.
- 25 El procedimiento 700 incluye un bloque de inicio 702 que pasa el control a un bloque funcional 704. El bloque funcional 704 dispone cada vista y profundidad correspondiente en una instancia de tiempo particular como una subimagen en formato de mosaico, y pasa el control a un bloque funcional 706. El bloque funcional 706 fija un elemento sintáctico `num_coded_views_minus1` y pasa el control a un bloque funcional 708. El bloque funcional 708 fija elementos sintácticos `org_pic_width_in_mbs_minus1` y `org_pic_height_in_mbs_minus1`, y pasa el control a un
- 30 bloque funcional 710. El bloque funcional 710 fija 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 inferior al número de vistas. Si es así, entonces el control se pasa a un bloque funcional 714. Si no, el control se pasa a un bloque funcional 724.
- 35 El bloque funcional 714 fija un elemento sintáctico `view_id[i]` y pasa el control a un bloque funcional 716. El bloque funcional 716 fija un elemento sintáctico `num_parts[view_id[i]]` y pasa el control a un bloque funcional 718. El bloque funcional 718 fija 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 inferior al valor actual del elemento sintáctico `num_parts[view_id[i]]`. Si es así, entonces el control se pasa a un bloque funcional 722. Si no, el control se pasa a un bloque funcional 728.
- 40 El bloque funcional 722 fija los siguientes elementos sintácticos, incrementa la variable 'j' y después devuelve el 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 `frame_crop_bottom_offset[view_id[i]][j]`.
- 45 El bloque funcional 728 fija un elemento sintáctico `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 sintáctico `upsample_view_flag[view_id[i]]` es igual a uno. Si es así, entonces el control se pasa a un bloque funcional 732. Si no, el control se pasa a un bloque de decisión 734.
- 50 El bloque funcional 732 fija un elemento sintáctico `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 sintáctico `upsample_filter[view_id[i]]` es igual a tres. Si es así, entonces el control se pasa a un bloque funcional 736. Si no, el control se pasa a un bloque funcional
- 55 740.
- El bloque funcional 736 fija los siguientes elementos sintácticos y pasa el control a un bloque funcional 738: `vert_dim[view_id[i]]`; `hor_dim[view_id[i]]`; y `quantizer[view_id[i]]`.
- 60 El bloque funcional 738 fija los coeficientes de filtro para cada componente YUV y pasa el control al bloque funcional 740.
- El bloque funcional 740 incrementa la variable 'i' y devuelve el control al bloque de decisión 712.
- 65 El bloque funcional 724 escribe estos elementos sintácticos en al menos uno de un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), un mensaje de información de mejora

complementaria (SEI), una cabecera de unidad de capa de abstracción de red (NAL) y una cabecera de segmento, y pasa el control a un bloque funcional 726. El bloque funcional 726 codifica cada imagen usando la norma MPEG-4 AVC u otro códec de vista única, y pasa el control a un bloque final 799.

5 Haciendo referencia a la FIG. 8, un procedimiento a modo de ejemplo para descodificar imágenes para una pluralidad de vistas y profundidades usando la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 800.

10 El procedimiento 800 incluye un bloque de inicio 802 que pasa el control a un bloque funcional 804. El bloque funcional 804 analiza sintácticamente los siguientes elementos sintácticos de al menos uno de un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), un mensaje de información de mejora complementaria (SEI), una cabecera de unidad de capa de abstracción de red (NAL), y una cabecera de segmento, y pasa el control a un bloque funcional 806. El bloque funcional 806 analiza sintácticamente un elemento sintáctico num_coded_views_minus1 y pasa el control a un bloque funcional 808. El bloque funcional 808 analiza sintácticamente elementos sintácticos org_pic_width_in_mbs_minus1 y org_pic_height_in_mbs_minus1, y pasa el control a un bloque funcional 810. El bloque funcional 810 fija 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 inferior al número de vistas. Si es así, entonces el control se pasa a un bloque funcional 814. Si no, el control se pasa a un bloque funcional 824.

20 El bloque funcional 814 analiza sintácticamente un elemento sintáctico view_id[i] y pasa el control a un bloque funcional 816. El bloque funcional 816 analiza sintácticamente un elemento sintáctico num_parts_minus1[view_id[i]] y pasa el control a un bloque funcional 818. El bloque funcional 818 fija 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 inferior al valor actual del elemento sintáctico num_parts[view_id[i]]. Si es así, entonces el control se pasa a un bloque funcional 822. Si no, el control se pasa a un bloque funcional 828.

30 El bloque funcional 822 analiza sintácticamente los siguientes elementos sintácticos, incrementa la variable 'j' y después devuelve el control al bloque de decisión 820: depth_flag[view_id[i]][j]; flip_dir[view_id[i]][j]; loc_left_offset[view_id[i]][j]; loc_top_offset[view_id[i]][j]; frame_crop_left_offset[view_id[i]][j]; frame_crop_right_offset[view_id[i]][j]; frame_crop_top_offset[view_id[i]][j]; y frame_crop_bottom_offset[view_id[i]][j].

35 El bloque funcional 828 analiza sintácticamente un elemento sintáctico 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 sintáctico upsample_view_flag[view_id[i]] es igual a uno. Si es así, entonces el control se pasa a un bloque funcional 832. Si no, el control se pasa a un bloque de decisión 834.

El bloque funcional 832 analiza sintácticamente un elemento sintáctico upsample_filter[view_id[i]] y pasa el control al bloque de decisión 834.

40 El bloque de decisión 834 determina si el valor actual del elemento sintáctico upsample_filter[view_id[i]] es igual a tres. Si es así, entonces el control se pasa a un bloque funcional 836. Si no, el control se pasa a un bloque funcional 840.

45 El bloque funcional 836 analiza sintácticamente los siguientes elementos sintácticos y pasa el control a un bloque funcional 838: vert_dim[view_id[i]]; hor_dim[view_id[i]]; y quantizer[view_id[i]].

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

50 El bloque funcional 840 incrementa la variable 'i' y devuelve el control al bloque de decisión 812.

55 El bloque funcional 824 descodifica cada imagen usando la norma MPEG-4 AVC u otro códec de vista única, y pasa el control a un bloque funcional 826. El bloque funcional 826 separa cada vista y profundidad correspondiente de la imagen usando la sintaxis de alto nivel y pasa el control a un bloque funcional 827. El bloque funcional 827 lleva a cabo potencialmente una síntesis de vistas usando la vista extraída y las señales de profundidad y pasa el control a un bloque final 899.

60 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 la profundidad se proporciona como un valor de píxel para cada ubicación correspondiente de una imagen (no mostrada). Además, la FIG. 10 muestra un ejemplo de dos señales de profundidad incluidas en un mosaico 1000. La parte superior derecha del mosaico 1000 es una señal de profundidad que tiene valores de profundidad correspondientes a la imagen en la parte superior izquierda del mosaico 1000. La parte inferior derecha del mosaico 1000 es una señal de profundidad que tiene valores de profundidad correspondientes a la imagen en la parte inferior izquierda del mosaico 1000.

65

Haciendo referencia a la FIG. 11, un ejemplo de 5 vistas presentadas en mosaico en un único cuadro se indica de manera genérica mediante el número de referencia 1100. Las cuatro vistas superiores están en una orientación normal. La quinta vista también está en una orientación normal, pero está dividida en dos partes a lo largo de la parte inferior del mosaico 1100. Una parte izquierda de la quinta vista muestra la "parte superior" de la quinta vista, y una parte derecha de la quinta vista muestra la "parte inferior" de la quinta vista.

Codificación/descodificación usando una norma/recomendación de codificación/descodificación de vídeo de múltiples vistas

Haciendo referencia a la FIG. 12, un codificador de codificación de vídeo de múltiples vistas (MVC) a modo de ejemplo se indica de manera genérica mediante el número de referencia 1200. El codificador 1200 incluye un combinador 1205 que presenta una salida conectada en comunicación de señales a una entrada de un transformador 1210. Una salida del transformador 1210 está conectada en comunicación de señales a una entrada de un cuantificador 1215. Una salida del cuantificador 1215 está conectada en comunicación de señales a una entrada de un codificador de entropía 1220 y a una entrada de un cuantificador inverso 1225. Una salida del cuantificador inverso 1225 está conectada en comunicación de señales a una entrada de un transformador inverso 1230. Una salida del transformador inverso 1230 está conectada en comunicación de señales a una primera entrada sin inversión de un combinador 1235. Una salida del combinador 1235 está conectada en comunicación de señales a una entrada de una unidad de intrapredicción 1245 y a una entrada de un filtro de desbloqueo 1250. Una salida del filtro de desbloqueo 1250 está conectada en comunicación de señales a una entrada de un almacenamiento de imágenes de referencia 1255 (para la vista i). Una salida del almacenamiento de imágenes de referencia 1255 está conectada en comunicación de señales a una primera entrada de un compensador de movimiento 1275 y a una primera entrada de un estimador de movimiento 1280. Una salida del estimador de movimiento 1280 está conectada en comunicación de señales a una segunda entrada del compensador de movimiento 1275.

Una salida de un almacenamiento de imágenes de referencia 1260 (para otras vistas) está conectada en comunicación de señales a una primera entrada de un estimador de disparidad 1270 y a una primera entrada de un compensador de disparidad 1265. Una salida del estimador de disparidad 1270 está conectada en comunicación de señales a una segunda entrada del compensador de disparidad 1265.

Una salida del descodificador de entropía 1220 está disponible como una salida del codificador 1200. Una entrada sin inversión del combinador 1205 está disponible como una entrada del codificador 1200, y está conectada en comunicación de señales a una segunda entrada del estimador de disparidad 1270 y a una segunda entrada del estimador de movimiento 1280. Una salida de un conmutador 1285 está conectada en comunicación de señales a una segunda entrada sin inversión del combinador 1235 y a una entrada con inversión del combinador 1205. El conmutador 1285 incluye una primera entrada conectada en comunicación de señales a una salida del compensador de movimiento 1275, una segunda entrada conectada en comunicación de señales a una salida del compensador de disparidad 1265, y una tercera entrada conectada en comunicación de señales a una salida de la unidad de intrapredicción 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.

Haciendo referencia a la FIG. 13, un descodificador de codificación de vídeo de múltiples vistas (MVC) a modo de ejemplo se indica de manera genérica mediante el número de referencia 1300. El descodificador 1300 incluye un descodificador de entropía 1305 que presenta una salida conectada en comunicación de señales a una entrada de un cuantificador inverso 1310. Una salida del cuantificador inverso está conectada en comunicación de señales a una entrada de un transformador inverso 1315. Una salida del transformador inverso 1315 está conectada en comunicación de señales a una primera entrada sin inversión de un combinador 1320. Una salida del combinador 1320 está conectada en comunicación de señales a una entrada de un filtro de desbloqueo 1325 y a una entrada de una unidad de intrapredicción 1330. Una salida del filtro de desbloqueo 1325 está conectada en comunicación de señales a una entrada de un almacenamiento de imágenes de referencia 1340 (para la vista i). Una salida del almacenamiento de imágenes de referencia 1340 está conectada en comunicación de señales a una primera entrada de un compensador de movimiento 1335.

Una salida de un almacenamiento de imágenes de referencia 1345 (para otras vistas) está conectada en comunicación de señales a una primera entrada de un compensador de disparidad 1350.

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

- Una salida de un conmutador 1355 está conectada en comunicación de señales a una segunda entrada sin inversión del combinador 1320. Una primera entrada del conmutador 1355 está conectada en comunicación de señales a una salida del compensador de disparidad 1350. Una segunda entrada del conmutador 1355 está conectada en comunicación de señales a una salida del compensador de movimiento 1335. Una tercera entrada del conmutador 1355 está conectada en comunicación de señales a una salida de la unidad de intrapredicción 1330. Una salida del módulo de modo 1360 está conectada en comunicación de señales al conmutador 1355 para controlar qué entrada es seleccionada por el conmutador 1355. Una salida del filtro de desbloqueo 1325 está disponible como una salida del descodificador 1300.
- 5
- 10 Haciendo referencia a la FIG. 14, un procedimiento a modo de ejemplo para procesar imágenes para una pluralidad de vistas como preparación a la codificación de las imágenes usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 1400.
- 15 El procedimiento 1400 incluye un bloque de inicio 1405 que pasa el control a un bloque funcional 1410. El bloque funcional 1410 dispone cada N vistas, entre un total de M vistas, en una instancia de tiempo particular como una superimagen en formato de mosaico, y pasa el control a un bloque funcional 1415. El bloque funcional 1415 fija un elemento sintáctico `num_coded_views_minus1` y pasa el control a un bloque funcional 1420. El bloque funcional 1420 fija un elemento sintáctico `view_id[i]` para todas las $(\text{num_coded_views_minus1} + 1)$ vistas y pasa el control a un bloque funcional 1425. El bloque funcional 1425 fija la información de dependencia de referencia entre vistas para imágenes de anclaje y pasa el control a un bloque funcional 1430. El bloque funcional 1430 fija la información de dependencia de referencia entre vistas para imágenes de no anclaje y pasa el control a un bloque funcional 1435. El bloque funcional 1435 fija un elemento sintáctico `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 sintáctico `pseudo_view_present_flag` es igual a verdadero. Si es así, entonces el control se pasa a un bloque funcional 1445. Si no, el control se pasa a un bloque final 1499.
- 20
- 25 El bloque funcional 1445 fija los siguientes elementos sintácticos y pasa el control a un bloque funcional 1450: `tiling_mode`; `org_pic_width_in_mbs_minus1`; y `org_pic_height_in_mbs_minus1`. El bloque funcional 1450 llama a un elemento sintáctico `pseudo_view_info(view_id)` para cada vista codificada y pasa el control al bloque final 1499.
- 30
- Haciendo referencia a la FIG. 15, un procedimiento a modo de ejemplo para codificar imágenes para una pluralidad de vistas usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 1500.
- 35
- El procedimiento 1500 incluye un bloque de inicio 1502 que tiene un parámetro de entrada `pseudo_view_id` y pasa el control a un bloque funcional 1504. El bloque funcional 1504 fija un elemento sintáctico `num_sub_views_minus1` y pasa el control a un bloque funcional 1506. El bloque funcional 1506 fija 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 inferior al número de subvistas. Si es así, entonces el control se pasa a un bloque funcional 1510. Si no, el control se pasa a un bloque funcional 1520.
- 40
- El bloque funcional 1510 fija un elemento sintáctico `sub_view_id[i]` y pasa el control a un bloque funcional 1512. El bloque funcional 1512 fija un elemento sintáctico `num_parts_minus1[sub_view_id[i]]` y pasa el control a un bloque funcional 1514. El bloque funcional 1514 fija 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 inferior al elemento sintáctico `num_parts_minus1[sub_view_id[i]]`. Si es así, entonces el control se pasa a un bloque funcional 1518. Si no, el control se pasa a un bloque de decisión 1522.
- 45
- 50 El bloque funcional 1518 fija los siguientes elementos sintácticos, 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]`.
- 55
- El bloque funcional 1520 codifica la imagen actual para la vista actual usando una codificación de vídeo de múltiples vistas (MVC) y pasa el control a un bloque final 1599.
- El bloque de decisión 1522 determina si un elemento sintáctico `tiling_mode` es igual a cero. Si es así, entonces el control se pasa a un bloque funcional 1524. Si no, el control se pasa a un bloque funcional 1538.
- 60
- El bloque funcional 1524 fija un elemento sintáctico `flip_dir[sub_view_id[i]]` y un elemento sintáctico `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 sintáctico `upsample_view_flag[sub_view_id[i]]` es igual a uno. Si es así, entonces el control se pasa a un bloque funcional 1528. Si no, el control se pasa a un bloque de decisión 1530.
- 65

El bloque funcional 1528 fija un elemento sintáctico `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 sintáctico `upsample_filter[sub_view_id[i]]` es igual a tres. Si es así, el control se pasa a un bloque funcional 1532. Si no, el control se pasa a un bloque funcional 1536.

5 El bloque funcional 1532 fija los siguientes elementos sintácticos y pasa el control a un bloque funcional 1534: `vert_dim[sub_view_id[i]]`; `hor_dim[sub_view_id[i]]`; y `quantizer[sub_view_id[i]]`. El bloque funcional 1534 fija los coeficientes de filtro para cada componente YUV y pasa el control al bloque funcional 1536.

10 El bloque funcional 1536 incrementa la variable 'i' y devuelve el control al bloque de decisión 1508.

El bloque funcional 1538 fija un elemento sintáctico `pixel_dist_x[sub_view_id[i]]` y un elemento sintáctico `flip_dist_y[sub_view_id[i]]` y pasa el control a un bloque funcional 1540. El bloque funcional 1540 fija 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 inferior al valor actual del elemento sintáctico `num_parts[sub_view_id[i]]`. Si es así, entonces el control se pasa a un bloque funcional 1544. Si no, el control se pasa al bloque funcional 1536.

15 El bloque funcional 1544 fija un elemento sintáctico `num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]]` y pasa el control a un bloque funcional 1546. El bloque funcional 1546 fija los coeficientes para todos los filtros de presentación en mosaico de píxeles y pasa el control al bloque funcional 1536.

20 Haciendo referencia a la FIG. 16, un procedimiento a modo de ejemplo para procesar imágenes para una pluralidad de vistas como preparación a la descodificación de las imágenes usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 1600.

25 El procedimiento 1600 incluye un bloque de inicio 1605 que pasa el control a un bloque funcional 1615. El bloque funcional 1615 analiza sintácticamente un elemento sintáctico `num_coded_views_minus1` y pasa el control a un bloque funcional 1620. El bloque funcional 1620 analiza sintácticamente un elemento sintáctico `view_id[i]` para todas las (`num_coded_views_minus1 + 1`) vistas y pasa el control a un bloque funcional 1625. El bloque funcional 1625 analiza sintácticamente la información de dependencia de referencia entre vistas para imágenes de anclaje y pasa el control a un bloque funcional 1630. El bloque funcional 1630 analiza sintácticamente la información de dependencia de referencia entre vistas para imágenes de no anclaje y pasa el control a un bloque funcional 1635. El bloque funcional 1635 analiza sintácticamente un elemento sintáctico `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 sintáctico `pseudo_view_present_flag` es igual a verdadero. Si es así, entonces el control se pasa a un bloque funcional 1645. Si no, el control se pasa a un bloque final 1699.

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

35 Haciendo referencia a la FIG. 17, un procedimiento a modo de ejemplo para descodificar imágenes para una pluralidad de vistas usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 1700.

40 El procedimiento 1700 incluye un bloque de inicio 1702 que empieza con un parámetro de entrada `pseudo_view_id` y pasa el control a un bloque funcional 1704. El bloque funcional 1704 analiza sintácticamente un elemento sintáctico `num_sub_views_minus1` y pasa el control a un bloque funcional 1706. El bloque funcional 1706 fija 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 inferior al número de subvistas. Si es así, entonces el control se pasa a un bloque funcional 1710. Si no, el control se pasa a un bloque funcional 1720.

45 El bloque funcional 1710 analiza sintácticamente un elemento sintáctico `sub_view_id[i]` y pasa el control a un bloque funcional 1712. El bloque funcional 1712 analiza sintácticamente un elemento sintáctico `num_parts_minus1[sub_view_id[i]]` y pasa el control a un bloque funcional 1714. El bloque funcional 1714 fija 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 inferior al elemento sintáctico `num_parts_minus1[sub_view_id[i]]`. Si es así, entonces el control se pasa a un bloque funcional 1718. Si no, el control se pasa a un bloque de decisión 1722.

50 El bloque funcional 1718 fija los siguientes elementos sintácticos, 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]`.

65

El bloque funcional 1720 descodifica la imagen actual para la vista actual usando una codificación de vídeo de múltiples vistas (MVC) y pasa el control a un bloque funcional 1721. El bloque funcional 1721 separa cada vista de la imagen usando la sintaxis de alto nivel y pasa el control a un bloque final 1799.

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

10 El bloque de decisión 1722 determina si un el elemento sintáctico `tiling_mode` es igual a cero. Si es así, entonces el control se pasa a un bloque funcional 1724. Si no, el control se pasa a un bloque funcional 1738.

15 El bloque funcional 1724 analiza sintácticamente un elemento sintáctico `flip_dir[sub_view_id[i]]` y un elemento sintáctico `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 sintáctico `upsample_view_flag[sub_view_id[i]]` es igual a uno. Si es así, entonces el control se pasa a un bloque funcional 1728. Si no, el control se pasa a un bloque de decisión 1730.

20 El bloque funcional 1728 analiza sintácticamente un elemento sintáctico `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 sintáctico `upsample_filter[sub_view_id[i]]` es igual a tres. Si es así, el control se pasa a un bloque funcional 1732. Si no, el control se pasa a un bloque funcional 1736.

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

El bloque funcional 1736 incrementa la variable 'i' y devuelve el control al bloque de decisión 1708.

30 El bloque funcional 1738 analiza sintácticamente un elemento sintáctico `pixel_dist_x[sub_view_id[i]]` y el elemento sintáctico `flip_dist_y[sub_view_id[i]]`, y pasa el control a un bloque funcional 1740. El bloque funcional 1740 fija 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 inferior al valor actual del elemento sintáctico `num_parts[sub_view_id[i]]`. Si es así, entonces el control se pasa a un bloque funcional 1744. Si no, el control se pasa al bloque funcional 1736.

35 El bloque funcional 1744 analiza sintácticamente un elemento sintáctico `num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]]` y pasa el control a un bloque funcional 1746. El bloque funcional 1746 analiza sintácticamente los coeficientes para todos los filtros de presentación en mosaico de píxeles y pasa el control al bloque funcional 1736.

40 Haciendo referencia a la FIG. 18, un procedimiento a modo de ejemplo para procesar imágenes para una pluralidad de vistas y profundidades como preparación a la codificación de las imágenes usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 1800.

45 El procedimiento 1800 incluye un bloque de inicio 1805 que pasa el control a un bloque funcional 1810. El bloque funcional 1810 dispone cada N vistas y mapas de profundidad, entre un total de M vistas y mapas de profundidad, en una instancia de tiempo particular como una superimagen en formato de mosaico, y pasa el control a un bloque funcional 1815. El bloque funcional 1815 fija un elemento sintáctico `num_coded_views_minus1` y pasa el control a un bloque funcional 1820. El bloque funcional 1820 fija un elemento sintáctico `view_id[i]` para todas las
 50 (`num_coded_views_minus1 + 1`) profundidades correspondientes a `view_id[i]` y pasa el control a un bloque funcional 1825. El bloque funcional 1825 fija la información de dependencia de referencia entre vistas para imágenes de profundidad de anclaje y pasa el control a un bloque funcional 1830. El bloque funcional 1830 fija la información de dependencia de referencia entre vistas para imágenes de profundidad de no anclaje y pasa el control a un bloque funcional 1835. El bloque funcional 1835 fija un elemento sintáctico `pseudo_view_present_flag` y pasa el control a un
 55 bloque de decisión 1840. El bloque de decisión 1840 determina si el valor actual del elemento sintáctico `pseudo_view_present_flag` es igual a verdadero. Si es así, entonces el control se pasa a un bloque funcional 1845. Si no, el control se pasa a un bloque final 1899.

60 El bloque funcional 1845 fija los siguientes elementos sintácticos y pasa el control a un bloque funcional 1850: `tiling_mode`; `org_pic_width_in_mbs_minus1`; y `org_pic_height_in_mbs_minus1`. El bloque funcional 1850 llama a un elemento sintáctico `pseudo_view_info(view_id)` para cada vista codificada y pasa el control al bloque final 1899.

65 Haciendo referencia a la FIG. 19, un procedimiento a modo de ejemplo para codificar imágenes para una pluralidad de vistas y profundidades usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 1900.

El procedimiento 1900 incluye un bloque de inicio 1902 que pasa el control a un bloque funcional 1904. El bloque funcional 1904 fija un elemento sintáctico `num_sub_views_minus1` y pasa el control a un bloque funcional 1906. El bloque funcional 1906 fija 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 inferior al número de subvistas. Si es así, entonces el control se pasa a un bloque funcional 1910. Si no, el control se pasa a un bloque funcional 1920.

El bloque funcional 1910 fija un elemento sintáctico `sub_view_id[i]` y pasa el control a un bloque funcional 1912. El bloque funcional 1912 fija un elemento sintáctico `num_parts_minus1[sub_view_id[i]]` y pasa el control a un bloque funcional 1914. El bloque funcional 1914 fija 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 inferior al elemento sintáctico `num_parts_minus1[sub_view_id[i]]`. Si es así, entonces el control se pasa a un bloque funcional 1918. Si no, el control se pasa a un bloque de decisión 1922.

El bloque funcional 1918 fija los siguientes elementos sintácticos, 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]`.

El bloque funcional 1920 codifica la profundidad actual para la vista actual usando una codificación de vídeo de múltiples vistas (MVC) y pasa el control a un bloque final 1999. La señal de profundidad puede codificarse de manera similar al modo en que se codifica su señal de vídeo correspondiente. Por ejemplo, la señal de profundidad para una vista puede incluirse en un mosaico que incluye solamente otras señales de profundidad, o solamente señales de vídeo, o señales tanto de profundidad como de vídeo. El mosaico (seudovista) se trata después como una única vista para MVC, y también hay presumiblemente otros mosaicos que se tratan como otras vistas para MVC.

El bloque de decisión 1922 determina si un el elemento sintáctico `tiling_mode` es igual a cero. Si es así, entonces el control se pasa a un bloque funcional 1924. Si no, el control se pasa a un bloque funcional 1938.

El bloque funcional 1924 fija un elemento sintáctico `flip_dir[sub_view_id[i]]` y un elemento sintáctico `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 sintáctico `upsample_view_flag[sub_view_id[i]]` es igual a uno. Si es así, entonces el control se pasa a un bloque funcional 1928. Si no, el control se pasa a un bloque de decisión 1930.

El bloque funcional 1928 fija un elemento sintáctico `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 sintáctico `upsample_filter[sub_view_id[i]]` es igual a tres. Si es así, el control se pasa a un bloque funcional 1932. Si no, el control se pasa a un bloque funcional 1936.

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

El bloque funcional 1936 incrementa la variable '`i`' y devuelve el control al bloque de decisión 1908.

El bloque funcional 1938 fija un elemento sintáctico `pixel_dist_x[sub_view_id[i]]` y el elemento sintáctico `flip_dist_y[sub_view_id[i]]` y pasa el control a un bloque funcional 1940. El bloque funcional 1940 fija 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 inferior al valor actual del elemento sintáctico `num_parts[sub_view_id[i]]`. Si es así, entonces el control se pasa a un bloque funcional 1944. Si no, el control se pasa al bloque funcional 1936.

El bloque funcional 1944 fija un elemento sintáctico `num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]]` y pasa el control a un bloque funcional 1946. El bloque funcional 1946 fija los coeficientes para todos los filtros de presentación en mosaico de píxeles y pasa el control al bloque funcional 1936.

Haciendo referencia a la FIG. 20, un procedimiento a modo de ejemplo para procesar imágenes para una pluralidad de vistas y profundidades como preparación a la descodificación de las imágenes usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 2000.

El procedimiento 2000 incluye un bloque de inicio 2005 que pasa el control a un bloque funcional 2015. El bloque funcional 2015 analiza sintácticamente un elemento sintáctico `num_coded_views_minus1` y pasa el control a un bloque funcional 2020. El bloque funcional 2020 analiza sintácticamente un elemento sintáctico `view_id[i]` para todas las (`num_coded_views_minus1 + 1`) profundidades correspondientes a `view_id[i]` y pasa el control a un bloque funcional 2025. El bloque funcional 2025 analiza sintácticamente la información de dependencia de referencia entre vistas para imágenes de profundidad de anclaje y pasa el control a un bloque funcional 2030. El bloque funcional

2030 analiza sintácticamente la información de dependencia de referencia entre vistas para imágenes de profundidad de no anclaje y pasa el control a un bloque funcional 2035. El bloque funcional 2035 analiza sintácticamente un elemento sintáctico 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 sintáctico pseudo_view_present_flag es igual a verdadero. Si es así, entonces el control se pasa a un bloque funcional 2045. Si no, el control se pasa a un bloque final 2099.

El bloque funcional 2045 analiza sintácticamente los siguientes elementos sintácticos y pasa el control a un bloque funcional 2050: tiling_mode; org_pic_width_in_mbs_minus1; y org_pic_height_in_mbs_minus1. El bloque funcional 2050 llama a un elemento sintáctico pseudo_view_info(view_id) para cada vista codificada y pasa el control al bloque final 2099.

Haciendo referencia a la FIG. 21, un procedimiento a modo de ejemplo para descodificar imágenes para una pluralidad de vistas y profundidades usando la extensión de codificación de vídeo de múltiples vistas (MVC) de la norma MPEG-4 AVC se indica de manera genérica mediante el número de referencia 2100.

El procedimiento 2100 incluye un bloque de inicio 2102 que empieza con un parámetro de entrada pseudo_view_id y pasa el control a un bloque funcional 2104. El bloque funcional 2104 analiza sintácticamente un elemento sintáctico num_sub_views_minus1 y pasa el control a un bloque funcional 2106. El bloque funcional 2106 fija una 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 inferior al número de subvistas. Si es así, entonces el control se pasa a un bloque funcional 2110. Si no, el control se pasa a un bloque funcional 2120.

El bloque funcional 2110 analiza sintácticamente un elemento sintáctico sub_view_id[i] y pasa el control a un bloque funcional 2112. El bloque funcional 2112 analiza sintácticamente un elemento sintáctico num_parts_minus1[sub_view_id[i]] y pasa el control a un bloque funcional 2114. El bloque funcional 2114 fija 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 inferior al elemento sintáctico num_parts_minus1[sub_view_id[i]]. Si es así, entonces el control se pasa a un bloque funcional 2118. Si no, el control se pasa a un bloque de decisión 2122.

El bloque funcional 2118 fija los siguientes elementos sintácticos, 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 funcional 2120 descodifica la imagen actual usando una codificación de vídeo de múltiples vistas (MVC) y pasa el control a un bloque funcional 2121. El bloque funcional 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 una sintaxis de alto nivel es como se ha descrito anteriormente.

El bloque de decisión 2122 determina si un el elemento sintáctico tiling_mode es igual a cero. Si es así, entonces el control se pasa a un bloque funcional 2124. Si no, el control se pasa a un bloque funcional 2138.

El bloque funcional 2124 analiza sintácticamente un elemento sintáctico flip_dir[sub_view_id[i]] y un elemento sintáctico 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 sintáctico upsample_view_flag[sub_view_id[i]] es igual a uno. Si es así, entonces el control se pasa a un bloque funcional 2128. Si no, el control se pasa a un bloque de decisión 2130.

El bloque funcional 2128 analiza sintácticamente un elemento sintáctico 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 sintáctico upsample_filter[sub_view_id[i]] es igual a tres. Si es así, el control se pasa a un bloque funcional 2132. Si no, el control se pasa a un bloque funcional 2136.

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

El bloque funcional 2136 incrementa la variable 'i' y devuelve el control al bloque de decisión 2108.

El bloque funcional 2138 analiza sintácticamente un elemento sintáctico pixel_dist_x[sub_view_id[i]] y el elemento sintáctico flip_dist_y[sub_view_id[i]] y pasa el control a un bloque funcional 2140. El bloque funcional 2140 fija 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 inferior al valor actual del elemento sintáctico num_parts[sub_view_id[i]]. Si es así, entonces el control se pasa a un bloque funcional 2144. Si no, el control se pasa al bloque funcional 2136.

El bloque funcional 2144 analiza sintácticamente un elemento sintáctico num_pixel_tiling_filter_co coeffs_minus1[sub_view_id[i]] y pasa el control a un bloque funcional 2146. El bloque funcional 2146 analiza sintácticamente los coeficientes para todos los filtros de presentación en mosaico de píxeles y pasa el control al bloque funcional 2136.

5 Haciendo referencia a la FIG. 22, ejemplos de presentación en mosaico a nivel de píxel se indican de manera genérica mediante el número de referencia 2200. La FIG. 22 se describe posteriormente en detalle.

10 Presentación de vistas en mosaico usando MPEG-4 AVC o MVC

Una aplicación de codificación de vídeo de múltiples vistas es una TV de punto de vista libre (o FTV). Esta aplicación requiere que el usuario pueda moverse libremente entre dos o más vistas. Para conseguir esto, las vistas "virtuales" entre dos vistas tienen que interpolarse o sintetizarse. Existen varios procedimientos para realizar una interpolación de vistas. Uno de los procedimientos usa profundidad para la interpolación/síntesis de vistas.

15 Cada vista puede tener una señal de profundidad asociada. Por tanto, la profundidad puede considerarse otra forma de señal de vídeo. La FIG. 9 muestra un ejemplo de una señal de profundidad 900. Para permitir aplicaciones tales como la FTV, la señal de profundidad se transmite junto con la señal de vídeo. En el marco propuesto de presentación en mosaico, la señal de profundidad también puede añadirse como uno de los mosaicos. La FIG. 10 muestra un ejemplo de señales de profundidad añadidas como mosaicos. Las señales de profundidad/mosaicos se muestran en el lado derecho de la FIG. 10.

25 Una vez que la profundidad se codifica como un mosaico de todo el cuadro, la sintaxis de alto nivel debería indicar qué mosaico es la señal de profundidad, de manera que el renderizador pueda usar la señal de profundidad de manera apropiada.

30 En caso de que la secuencia de entrada (tal como la mostrada en la FIG. 1) se codifique usando un codificador basado en la norma MPEG-4 AVC (o un codificador correspondiente a una norma y/o una recomendación de codificación de vídeo diferente), la sintaxis de alto nivel propuesta puede estar presente en, por ejemplo, el conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS), una cabecera de segmento y/o un mensaje de información de mejora complementaria (SEI). Una forma de realización del procedimiento propuesto se muestra en la Tabla 1, donde la sintaxis está presente en un mensaje de información de mejora complementaria (SEI).

35 En caso de que las secuencias de entrada de las seudovistas (tal como la mostrada en la FIG. 1) se codifiquen usando la extensión de la codificación de vídeo de múltiples vistas (MVC) del codificador basado en la norma MPEG-4 AVC (o de un codificador correspondiente a una norma de codificación de vídeo de múltiples vistas con respecto a una norma y/o recomendación de codificación de vídeo diferente), la sintaxis de alto nivel propuesta puede estar presente en el SPS, el PPS, una cabecera de segmento, un mensaje de SEI o un perfil especificado. Una forma de realización del procedimiento propuesto se muestra en la Tabla 1. La Tabla 1 muestra elementos sintácticos presentes en la estructura de conjunto de parámetros de secuencia (SPS), incluidos elementos sintácticos propuestos según una forma de realización de los presentes principios.

Tabla 1

seq_parameter_set_mvc_extension() {	C	Descriptor
num_views_minus_1		ue(v)
for(i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1 [i]		ue(v)
for(j = 0; j < num_anchor_refs_l1 [i]; j++)		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)

seq_parameter_set_mvc_extension() {	C	Descriptor
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]		ue(v)
num_non_anchor_refs_l1 [i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]		ue(v)
}		
pseudo_view_present_flag		u(1)
if (pseudo_view_present_flag) {		
tiling_mode		
org_pic_width_in_mbs_minus1		
org_pic_height_in_mbs_minus1		
for(i = 0; i < num_views_minus_1; i++)		
pseudo_view_info(i);		
}		
}		

La Tabla 2 muestra elementos sintácticos para el elemento sintáctico pseudo_view_info de la Tabla 1, según una forma de realización de los presentes principios.

5

Tabla 2

pseudo_view_info (pseudo_view_id) {	C	Descriptor
num_sub_views_minus_1[pseudo_view_id]	5	ue(v)
if (num_sub_views_minus_1 != 0) {		
for (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)
for(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)
}		
if (tiling_mode == 0) {		
flip_dir[sub_view_id[i][j]]	5	u(2)
upsample_view_flag[sub_view_id[i]]	5	u(1)
if(upsample_view_flag[sub_view_id[i]])		
upsample_filter[sub_view_id[i]]	5	u(2)
if(upsample_fiter[sub_view_id[i]] == 3) {		
vert_dim[sub_view_id[i]]	5	ue(v)

pseudo_view_info (pseudo_view_id) {	C	Descriptor
hor_dim[sub_view_id[i]]	5	ue(v)
quantizer[sub_view_id[i]]	5	ue(v)
for (yuv= 0; yuv< 3; yuv++) {		
for (y = 0; y < vert_dim[sub_view_id[i]]-1; y ++) {		
for (x = 0; x < hor_dim[sub_view_id[i]]-1; x ++)		
filter_coefs[sub_view_id[i]] [yuv][y][x]	5	se(v)
}		
}		
}		
} // if(tiling_mode == 0)		
else if (tiling_mode == 1) {		
pixel_dist_x[sub_view_id[i]]		
pixel_dist_y[sub_view_id[i]]		
for(j = 0; j <= num_parts[sub_view_id[i]]; j++) {		
num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]][j]		
for (coeff_idx = 0; coeff_idx <=		
num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]][j]; j++)		
pixel_tiling_filter_coefs[sub_view_id[i]][j][coeff_idx]		
} // for (j = 0; j <= num_parts[sub_view_id[i]]; j++)		
} // else if (tiling_mode == 1)		
} // for (i = 0; i < num_sub_views_minus_1; i++)		
} // if (num_sub_views_minus_1 != 0)		
}		

Semántica de los elementos sintácticos presentados en la Tabla y la Tabla 2

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

5 tiling_mode igual a 0 indica que las subvistas están presentadas en mosaico a nivel de imagen. Un valor de 1 indica que la presentación en mosaico se realiza a nivel de píxel.

10 El nuevo mensaje de SEI puede usar un valor para el tipo datos útiles de SEI que no se ha usado en la norma MPEG-4 AVC o en una extensión de la norma MPEG-4 AVC. El nuevo mensaje de SEI incluye varios elementos sintácticos con la siguiente semántica.

num_coded_views_minus1 + 1 indica el número de vistas codificadas admitidas por el flujo de bits. El valor de num_coded_views_minus1 está dentro del intervalo comprendido entre 0 y 1023, ambos inclusive.

15 org_pic_width_in_mbs_minus1 + 1 especifica la anchura de una imagen en cada vista en unidades de macrobloques.

20 La variable para la anchura de imagen en unidades de macrobloques se obtiene de la siguiente manera:
PicWidthInMbs = org_pic_width_in_mbs_minus1 + 1

La variable para la anchura de imagen para la componente de brillo se obtiene de la siguiente manera:
PicWidthInSamplesL = PicWidthInMbs * 16

25 La variable para la anchura de imagen para la componente de color se obtiene de la siguiente manera:
PicWidthInSamplesC = PicWidthInMbs * MbWidthC

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

La variable para la altura de imagen en unidades de macrobloques se obtiene de la siguiente manera:
 $PicHeightInMbs = org_pic_height_in_mbs_minus1 + 1$

5 La variable para la altura de imagen para la componente de brillo se obtiene de la siguiente manera:
 $PicHeightInSamplesL = PicHeightInMbs * 16$

10 La variable para la altura de imagen para la componente de color se obtiene de la siguiente manera:
 $PicHeightInSamplesC = PicHeightInMbs * MbHeightC$

$num_sub_views_minus1 + 1$ indica el número de subvistas codificadas incluidas en la vista actual. El valor de $num_coded_views_minus1$ está dentro del intervalo comprendido entre 0 y 1023, ambos inclusive.

15 $sub_view_id[i]$ especifica el sub_view_id de la subvista con un orden de descodificación indicado por i .

$num_parts[sub_view_id[i]]$ especifica el número de partes en que está dividida la imagen de $sub_view_id[i]$.

20 $loc_left_offset[sub_view_id[i]][j]$ y $loc_top_offset[sub_view_id[i]][j]$ especifican las ubicaciones en desplazamientos de píxeles izquierdos y superiores, respectivamente, donde la parte actual j está ubicada en la imagen reconstruida final de la vista, donde sub_view_id es igual a $sub_view_id[i]$.

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

25 $frame_crop_left_offset[view_id[i]][j]$, $frame_crop_right_offset[view_id[i]][j]$, $frame_crop_top_offset[view_id[i]][j]$, y $frame_crop_bottom_offset[view_id[i]][j]$ especifican las muestras de las imágenes en la secuencia de vídeo codificada que son parte de $num_part\ j$ y $view_id\ i$, en lo que respecta a una región rectangular especificada en coordenadas de cuadro para la salida.

30 Las variables $CropUnitX$ y $CropUnitY$ se obtienen de la siguiente manera:

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

35 $CropUnitX = 1$
 $CropUnitY = 2 - frame_mbs_only_flag$

- En otro caso ($chroma_format_idc$ es igual a 1, 2 o 3), $CropUnitX$ y $CropUnitY$ se obtienen de la siguiente manera:

40 $CropUnitX = SubWidthC$
 $CropUnitY = SubHeightC * (2 - frame_mbs_only_flag)$

El rectángulo de recorte de cuadro incluye muestras de brillo con coordenadas de cuadro horizontales a partir de lo siguiente:

45 $CropUnitX * frame_crop_left_offset$ hasta $PicWidthInSamplesL - (CropUnitX * frame_crop_right_offset + 1)$ y coordenadas de cuadro verticales desde $CropUnitY * frame_crop_top_offset$ hasta $(16 * FrameHeightInMbs) - (CropUnitY * frame_crop_bottom_offset + 1)$, inclusive. El valor de $frame_crop_left_offset$ estará dentro del intervalo comprendido entre 0 y $(PicWidthInSamplesL / CropUnitX) - (frame_crop_right_offset + 1)$, inclusive; y el valor de $frame_crop_top_offset$ estará dentro del intervalo comprendido entre 0 y $(16 * FrameHeightInMbs / CropUnitY) - (frame_crop_bottom_offset + 1)$, inclusive.

55 Cuando $chroma_format_idc$ no es igual a 0, las muestras especificadas correspondientes de los dos vectores cromáticos son las muestras que tienen coordenadas de cuadro $(x / SubWidthC, y / SubHeightC)$, donde (x, y) son las coordenadas de cuadro de las muestras de brillo especificadas.

En lo que respecta a los campos descodificados, las muestras especificadas del campo descodificado son las muestras que están dentro del rectángulo especificado en coordenadas de cuadro.

60 $num_parts[view_id[i]]$ especifica el número de partes en que está dividida la imagen de $view_id[i]$.

$depth_flag[view_id[i]]$ especifica si la parte actual es, o no, 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 a la vista identificada por $view_id[i]$.

flip_dir[sub_view_id[i]][j] especifica la dirección de giro de la parte actual. flip_dir igual a 0 indica que no hay giro, flip_dir igual a 1 indica giro en una dirección horizontal, flip_dir igual a 2 indica giro en una dirección vertical, y flip_dir igual a 3 indica giro en una dirección tanto horizontal como vertical.

5 flip_dir[view-id[i]][j] especifica la dirección de giro de la parte actual. flip_dir igual a 0 indica que no hay giro, flip_dir igual a 1 indica giro en una dirección horizontal, flip_dir igual a 2 indica giro en una dirección vertical, y flip_dir igual a 3 indica giro en una dirección tanto horizontal como vertical.

10 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 está ubicada en la imagen reconstruida final de la vista, donde view_id es igual a view_id[i].

upsample_view_flag[view_id[i]] indica si la imagen que pertenece a la vista especificada por view_id[i] tiene que muestrearse de manera ascendente. upsample_view_flag[view_id[i]] igual a 0 especifica que la imagen con view_id igual a view_id[i] no se muestreará de manera ascendente. upsample_view_flag[view_id[i]] igual a 1 especifica que la imagen con view_id igual a view_id[i] se muestreará de manera ascendente.

upsample_filter[view_id[i]] indica el tipo de filtro que va a usarse para el muestreo ascendente. upsample_filter[view_id[i]] igual a 0 indica que debe usarse un filtro AVC de 6 etapas, upsample_filter[view_d[i]] igual a 1 indica que debe usarse un filtro SVC de 4 etapas, upsample_filter[view_d[i]] igual a 2 indica que debe usarse un filtro bilineal, upsample_filter[view_id[i]] igual a 3 indica que se transmiten coeficientes de filtro personalizados. Cuando upsample_fiter[view_id[i]] no está presente está fijado a 0. En esta forma de realización se usa un filtro personalizado 2D. Puede extenderse fácilmente a un filtro 1D, y a algún otro filtro no lineal.

25 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 cuantificación para cada coeficiente de filtro.

30 filter_coefs[view_id[i]] [yuv][y][x] especifica los coeficientes de filtro cuantificados. yuv indica 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.

35 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 seudovista reconstruida final entre píxeles vecinos en la vista con sub_view_id igual a sub_view_id[i].

40 num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]][j] + 1 indica el número de coeficientes de filtro cuando el modo de presentación en mosaico se fija igual a 1.

pixel_tiling_filter_coefs[sub_view_id[i]][j] señala los coeficientes de filtro que se requieren para representar un filtro que puede usarse para filtrar la imagen presentada en mosaico.

Ejemplos de presentación en mosaico a nivel de píxel

45 Haciendo referencia a la FIG. 22, dos ejemplos que muestran la composición de una seudovista presentando píxeles en mosaico de cuatro vistas se indican respectivamente mediante los números de referencia 2210 y 2220, respectivamente. Las cuatro vistas están indicadas conjuntamente mediante el número de referencia 2250. Los valores sintácticos del primer ejemplo en la FIG. 22 se proporcionan en la siguiente Tabla 3.

Tabla 3

pseudo_view_info (pseudo_view_id) {	Valor
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

pseudo_view_info (pseudo_view_id) {	Valor
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 sintácticos del segundo ejemplo en la FIG. 22 son todos idénticos excepto los dos elementos sintácticos siguientes: loc_left_offset[3][0] igual a 5 y loc_top_offset[3][0] igual a 3.

5 El desplazamiento indica que los píxeles correspondientes a una vista deberían empezar en una determinada ubicación desplazada. Esto se muestra en la FIG. 22 (2220). Esto puede realizarse, por ejemplo, cuando dos vistas producen imágenes en las que objetos comunes aparecen desplazados de una vista a otra. Por ejemplo, si una primera y una segunda cámara (que representan una primera y una segunda vista) toman imágenes de un objeto, el objeto puede aparecer cinco píxeles desplazado a la derecha en la segunda vista en comparación con la primera vista. Esto significa que el píxel (i-5, j) en la primera vista corresponde al píxel (i, j) en la segunda vista. Si los píxeles de las dos vistas están simplemente presentados en mosaico píxel a píxel, entonces puede no haber una gran correlación entre píxeles vecinos en el mosaico, y las ganancias de codificación espacial pueden ser pequeñas. Por 10 el contrario, desplazando el mosaico de manera que el píxel(i-5, j) de la vista uno se sitúe junto al píxel(i, j) de la vista dos, la correlación espacial puede aumentar y la ganancia de codificación espacial también puede aumentar. Esto se debe a que, por ejemplo, los píxeles correspondientes para el objeto en la primera y en la segunda vista se 15 presentan en mosaico de manera próxima entre sí.

Por tanto, la presencia de loc_left_offset y loc_top_offset puede mejorar la eficacia de codificación. La información de desplazamiento puede obtenerse por medios externos. Por ejemplo, la información de posición de las cámaras o los vectores de disparidad global entre las vistas pueden usarse para determinar tal información de desplazamiento. 20

Como resultado del desplazamiento, algunos píxeles de la seudovista no tienen asignados valores de píxel de ninguna vista. Siguiendo con el ejemplo anterior, cuando se presenta en mosaico el píxel(i-5, j) de la vista uno junto al píxel(i, j) de la vista dos, para los valores de i=0... 4 no hay ningún píxel(i-5, j) de la vista uno que presentar en mosaico, de modo que esos píxeles están vacíos en el mosaico. Para aquellos píxeles en la seudovista (mosaico) 25 que no tengan asignados valores de píxel de ninguna vista, al menos una implementación usa un procedimiento de interpretación similar al procedimiento de interpretación de subpíxeles en la compensación de movimiento en AVC. Es decir, los píxeles de mosaico vacíos pueden interpolarse a partir de píxeles vecinos. Tal interpolación puede dar como resultado una mayor correlación espacial en el mosaico y una mayor ganancia de codificación para el mosaico. 30

En la codificación de vídeo, puede elegirse un tipo de codificación diferente para cada imagen, tal como imágenes I, P y B. En la codificación de vídeo de múltiples vistas, se definen imágenes de anclaje y de no anclaje. En una forma de realización, se propone que la decisión de agrupación pueda tomarse según un tipo de imagen. Esta información de agrupación se señala en la sintaxis de alto nivel. 35

Haciendo referencia a la FIG. 11, un ejemplo de 5 vistas presentadas en mosaico en un único cuadro se indica de manera genérica mediante el número de referencia 1100. En particular, la secuencia de baile de salón se muestra

con 5 vistas presentadas en mosaico en un único cuadro. Además, puede observarse que la quinta vista está dividida en dos partes, de manera que puede disponerse en un cuadro rectangular. En este caso, cada vista tiene un tamaño QVGA, de manera que la dimensión de cuadro total es de 640x600. Puesto que 600 no es un múltiplo de 16, debe aumentarse hasta 608.

5

En este ejemplo, el mensaje de SEI posible puede ser como el mostrado en la Tabla 4.

Tabla 4

multiview_display_info(payloadSize) {	Valor
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
if(upsample_view_flag[view_id[0]]) {	
vert_dim[view_id[0]]	6
hor_dim[view_id[0]]	6
quantizer[view_id[0]]	32
for (yuv= 0; yuv< 3; yuv++) {	
for (y = 0; y < vert_dim[view_id[i]] - 1; y ++) {	
for (x = 0; x < hor_dim[view_id[i]]-1; x ++)	
filter_coefs[view_id[i]][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

ES 2 675 164 T3

multiview_display_info(payloadSize) {	Valor
loc_top_offset[view_id[1]] [0]	0
frame_crop_left_offset[view_id[1]] [0]	320
frame_crop_right_offset[view_id[1]] [0]	640
frame_crop_top_offset[view_id[1]] [0]	0
frame_crop_bottom_offset[view_id[1]] [0]	320
upsample_view_flag[view_id[1]]	1
if(upsample_view_flag[view_id[1]]) {	
vert_dim[view_id[1]]	6
hor_dim[view_id[1]]	6
quantizer[view_id[1]] for (yuv= 0; yuv< 3; yuv++) {	32
for (y = 0; y < vert_dim[view_id[i]] -1; y ++)	
for (x = 0; x < hor_dim[view_id[i]]-1; x ++)	
filter_coefss[view_id[i]] [yuv][y][x]	XX
... (de manera similar para las vistas 2 y 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] frame_crop_top_offset[view_id[4]] [1]	640 480
frame_crop_bottom_offset[view_id[4]] [1]	600

multiview_display_info(payloadSize) {	Valor
upsample_view_flag[view_id[4]]	1
if(upsample_view_flag[view_id[4]]) {	
vert_dim[view_id[4]]	6
hor_dim[view_id[4]]	6
quantizer[view_id[4]]	32
for (yuv= 0; yuv< 3; yuv++) {	
for (y = 0; y < vert_dim[view_id[i]] -1; y ++) {	
for (x = 0; x < hor_dim[view_id[i]]-1; x ++)	
filter_coefs[view_id[i]] [yuv][y][x]	XX

La Tabla 5 muestra la estructura sintáctica general para transmitir información de múltiples vistas para el ejemplo mostrado en la Tabla 4.

5

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)
for(i = 0; i <= num_coded_views_minus1; i++) {		
view_id[i]	5	ue(v)
num_parts[view_id[i]]	5	ue(v)
for(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)
if(upsample_view_flag[view_id[i]])		
upsample_filter[view_id[i]]	5	u(2)
if(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)
for (yuv= 0; yuv< 3; yuv++) {		
for (y = 0; y < vert_dim[view_id[i]]-1; y ++) {		
for (x = 0; x < hor_dim[view_id[i]]-1; x ++)		
filter_coefs[view_id[i]][yuv][y][x]	5	se(v)

multiview_display_info(payloadSize) {	C	Descriptor
}		
}		
}		
}		
}		

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 descodificador u otro dispositivo que reciba vídeo codificado y que proporcione, por ejemplo, vídeo descodificado para mostrarse a un usuario o para almacenarse.
 5 Por tanto, el dispositivo 2300 puede proporcionar su salida a una televisión, un monitor de ordenador, un ordenador u otro dispositivo de procesamiento.

El dispositivo 2300 incluye un descodificador 2310 que recibe una señal de datos 2320. La señal de datos 2320 puede incluir, por ejemplo, un flujo compatible con AVC o con MVC. El descodificador 2310 descodifica toda o parte
 10 de la señal recibida 2320 y proporciona como salida una señal de vídeo descodificada 2330 e información de presentación en mosaico 2340. El vídeo descodificado 2330 y la información de presentación en mosaico 2340 se proporcionan a un selector 2350. El dispositivo 2300 incluye además una interfaz de usuario 2360 que recibe una entrada de usuario 2370. La interfaz de usuario 2360 proporciona al selector 2350 una señal de selección de imagen 2380 basándose en la entrada de usuario 2370. La señal de selección de imagen 2380 y la entrada de usuario 2370
 15 indican cuál de las múltiples imágenes desea ver un usuario. El selector 2350 proporciona la(s) imagen(es) seleccionada(s) como una salida 2390. El selector 2350 usa la información de selección de imagen 2380 para seleccionar la imagen del vídeo descodificado 2330 que va a proporcionarse como la salida 2390. El selector 2350 usa la información de presentación en mosaico 2340 para asignar la(s) imagen(es) seleccionada(s) del vídeo descodificado 2330.

20 En varias implementaciones, el selector 2350 incluye la interfaz de usuario 2360, mientras que en otras implementaciones no se necesita ninguna interfaz de usuario 2360 ya que el selector 2350 recibe la entrada de usuario 2370 directamente sin que se realice ninguna otra función de interfaz. El selector 2350 puede implementarse en software o como un circuito integrado, por ejemplo. El selector 2350 también puede incorporar el descodificador 2310.
 25

En términos más generales, los descodificadores de varias implementaciones descritas en esta solicitud pueden proporcionar una salida descodificada que incluye un mosaico completo. Además, o como alternativa, los descodificadores pueden proporcionar una salida descodificada que incluye solamente una o más imágenes seleccionadas (imágenes o señales de profundidad, por ejemplo) del mosaico.
 30

Como se ha indicado anteriormente, puede usarse una sintaxis de alto nivel para realizar la señalización según una o más formas de realización de los presentes principios. Por ejemplo, puede usarse la sintaxis de alto nivel, pero no está limitada a señalar cualquiera de lo siguiente: el número de vistas codificadas presentes en el cuadro mayor; 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 que se divide el cuadro de una vista; para cada parte de la vista, la dirección de giro (que puede ser, por ejemplo, nula, solamente un giro horizontal, solamente un giro vertical o un giro horizontal y vertical); para cada parte de la vista, la posición izquierda en los píxeles o el número de macrobloques a los que pertenece la parte actual en el cuadro final para la vista; para cada parte de la vista, la posición superior de la parte en píxeles o el número de macrobloques a los que pertenece la parte actual en el cuadro final para la vista; para cada parte de la vista, la posición izquierda, en el gran cuadro descodificado/codificado actual, de la ventana de recorte en píxeles o número de macrobloques; para cada parte de la vista, la posición derecha, en el gran cuadro descodificado/codificado actual, de la ventana de recorte en píxeles o número de macrobloques; para cada parte de la vista, la posición superior, en el gran cuadro descodificado/codificado actual, de la ventana de recorte en píxeles o número de macrobloques; y para cada parte de la vista, la posición inferior, en el gran cuadro descodificado/codificado actual, de la ventana de recorte en píxeles o número de macrobloques; para cada vista codificada, si la vista tiene que muestrearse de manera ascendente antes de proporcionarse (donde si es necesario realizar el muestreo ascendente, puede usarse una sintaxis de alto nivel para indicar el procedimiento de muestreo ascendente (que incluye, pero sin limitarse a, un filtro AVC de 6 etapas, un filtro SVC de 4 etapas, un filtro bilineal o un filtro lineal o no lineal 1D, 2D personalizado)).
 35
 40
 45
 50

Debe observarse que los términos "codificador" y "descodificador" connotan estructuras generales y no están limitados a ninguna función o característica particular. Por ejemplo, un descodificador puede recibir una portadora modulada que transporta un flujo de bits codificado, y desmodular el flujo de bits codificado, así como descodificar el flujo de bits.
 55

Se han descrito varios procedimientos. Muchos de estos procedimientos se han descrito en detalle para ofrecer una divulgación concisa. Sin embargo, debe observarse que se contemplan variaciones que pueden variar una o muchas de las características específicas descritas para estos procedimientos. Además, muchas de las características enumeradas son conocidas en la técnica y, por consiguiente, no se describen en gran detalle.

5 Además, se ha hecho referencia al uso de una sintaxis de alto nivel para enviar cierta información en varias implementaciones. Sin embargo, debe entenderse que otras implementaciones usan una sintaxis de menor nivel o, de hecho, otros mecanismos diferentes (tales como, por ejemplo, enviar información como parte de los datos codificados) para proporcionar la misma información (o variaciones de esa información).

10 Varias implementaciones proporcionan una presentación en mosaico y una señalización apropiada para permitir que múltiples vistas (más en general, imágenes) se presenten en mosaico en una única imagen, se codifiquen como una única imagen y se envíen como una única imagen. La información de señalización puede permitir que un posprocesador fragmente las vistas/imágenes. Además, las múltiples imágenes que están presentadas en mosaico pueden ser vistas, pero al menos una de las imágenes puede ser información de profundidad. Estas implementaciones pueden proporcionar una o más ventajas. Por ejemplo, los usuarios pueden desear mostrar múltiples vistas en mosaico, y estas diversas implementaciones proporcionan una manera eficiente de codificar y transmitir o almacenar tales vistas disponiéndolas en mosaico antes de su codificación y transmisión/almacenamiento en mosaico.

20 Las implementaciones que presentan múltiples vistas en mosaico en el contexto de AVC y/o MVC también ofrecen ventajas adicionales. AVC solo se usa ostensiblemente para una única vista, por lo que no se espera ninguna vista adicional. Sin embargo, tales implementaciones basadas en AVC pueden proporcionar múltiples vistas en un entorno AVC, ya que las vistas presentadas en mosaico pueden disponerse de manera que, por ejemplo, un descodificador sepa que las imágenes presentadas en mosaico pertenecen a diferentes vistas (por ejemplo, la imagen superior izquierda en la seudovista es la vista 1, la imagen superior derecha es la vista 2, etc.).

30 Además, MVC ya incluye múltiples vistas, de manera que no se espera que múltiples vistas estén incluidas en una única seudovista. Además, MVC tiene un límite en lo que respecta al número de vistas que pueden admitirse, y tales implementaciones basadas en MVC aumentan de manera eficaz el número de vistas que pueden admitirse al permitir (como en las implementaciones basadas en AVC) que se presenten en mosaico vistas adicionales. Por ejemplo, cada seudovista puede corresponder a una de las vistas admitidas de MVC, y el descodificador puede saber que cada "vista admitida" incluye realmente cuatro vistas en un orden predispuesto de presentación en mosaico. Por tanto, en esta implementación, el número de posibles vistas es cuatro veces el número de "vistas admitidas".

40 Las implementaciones descritas en el presente documento pueden implementarse en, por ejemplo, un procedimiento o proceso, un aparato o un programa de software. Incluso si solo se describe en el contexto de una única forma de implementación (por ejemplo, solo se describe como un procedimiento), la implementación de las características descritas también puede implementarse de otras maneras (por ejemplo, un aparato o un programa). Un aparato puede implementarse, por ejemplo, en un hardware, software y firmware apropiado. Los procedimientos pueden implementarse, por ejemplo, en un aparato tal como, por ejemplo, un procesador, que hace referencia a dispositivos de procesamiento en general, incluidos, por ejemplo, un ordenador, un microprocesador, un circuito integrado o un dispositivo de lógica programable. Dispositivos de procesamiento incluyen además dispositivos de comunicación, tales como, por ejemplo, ordenadores, teléfonos celulares, asistentes digitales portátiles/personales (PDA), y otros dispositivos que faciliten la comunicación de información entre usuarios finales.

50 Implementaciones de los diversos procesos y características descritos en el presente documento pueden realizarse en varios equipos o aplicaciones diferentes, por ejemplo, de manera particular, en equipos o aplicaciones asociados a la codificación y descodificación de datos. Ejemplos de equipos incluyen codificadores de vídeo, descodificadores de vídeo, códecs de vídeo, servidores web, descodificadores, ordenadores portátiles, ordenadores personales, teléfonos celulares, PDA y otros dispositivos de comunicación. Como resultará evidente, los equipos pueden ser móviles e incluso estar instalados en un vehículo móvil.

55 Además, los procedimientos pueden implementarse mediante instrucciones realizadas por un procesador, y tales instrucciones pueden almacenarse 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 disquete compacto, una memoria de acceso aleatorio (RAM) o una memoria de solo lectura (ROM). Las instrucciones pueden formar un programa de aplicación guardado de manera tangible en un medio legible por procesador. Como resultará evidente, un procesador puede incluir un medio legible por procesador que tiene, por ejemplo, instrucciones para llevar a cabo un proceso. Tales programas de aplicación pueden cargarse en y ejecutarse por una máquina que comprende cualquier arquitectura adecuada. Preferentemente, la máquina está implementada en una plataforma informática que cuenta con 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 el presente documento pueden ser parte del código de microinstrucciones o parte del programa de aplicación, o cualquier

combinación de los mismos, que pueda ejecutarse por una CPU. Además, otras diversas unidades periféricas pueden estar conectadas a la plataforma informática, tal como una unidad de almacenamiento de datos adicional y una unidad de impresión.

- 5 Como le resultará evidente a un experto en la técnica, las implementaciones también pueden producir una señal formateada para transportar información que, por ejemplo, puede almacenarse o transmitirse. La información puede incluir, por ejemplo, instrucciones para realizar un procedimiento, o datos producidos por una de las implementaciones descritas. Una señal de este tipo puede formatearse, por ejemplo, como una onda electromagnética (por ejemplo, usando una parte de radiofrecuencia del espectro) o como una señal de banda base.
- 10 El formateo puede incluir, por ejemplo, codificar un flujo de datos, producir una sintaxis y modular una portadora con el flujo de datos codificado y la sintaxis. La información que la señal transporta puede ser, por ejemplo, información analógica o digital. La señal puede transmitirse a través de diversos enlaces alámbricos o inalámbricos diferentes, como se sabe.
- 15 Debe entenderse además que, puesto que algunos de los componentes de sistema constituyentes y de los procedimientos descritos en los dibujos adjuntos se implementan preferentemente en software, las conexiones reales entre los componentes de sistema o los bloques funcionales de proceso pueden diferir dependiendo de la manera en que estén programados los presentes principios. Dadas las enseñanzas del presente documento, un experto en la técnica podrá contemplar estas y otras implementaciones o configuraciones similares de los presentes
- 20 principios.

Se han descrito diversas implementaciones. Sin embargo, debe entenderse que pueden realizarse varias modificaciones. Por ejemplo, elementos de diferentes implementaciones pueden combinarse, complementarse, modificarse o eliminarse para producir otras implementaciones. Además, un experto en la técnica entenderá que

25 otras estructuras y procesos pueden sustituir a los descritos, y que las implementaciones resultantes llevarán a cabo, al menos sustancialmente, la(s) misma(s) función(es), en al menos sustancialmente la(s) misma(s) manera(s), para conseguir, al menos sustancialmente, el/los mismo(s) resultado(s) que las implementaciones dadas a conocer. En particular, aunque en el presente documento se han descrito formas de realización ilustrativas haciendo referencia a los dibujos adjuntos, debe entenderse que los presentes principios no están limitados a esas formas de

30 realización precisas, y que un experto en la técnica puede realizar varios cambios y modificaciones en los mismos sin apartarse del alcance de los presentes principios definidos por las reivindicaciones. Por consiguiente, estas y otras implementaciones son contempladas por esta solicitud y están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento, caracterizado por que comprende:

- 5 - recibir un flujo de vídeo;
- acceder a una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen (826), donde las múltiples imágenes incluyen una primera imagen de una primera vista de un vídeo de múltiples vistas y una segunda imagen de una segunda vista del vídeo de múltiples vistas, donde la imagen de vídeo forma parte del flujo de vídeo recibido;
- 10 - acceder a información que indica cómo las múltiples imágenes de la imagen de vídeo accedida están combinadas, en el que la información accedida indica que al menos una de las múltiples imágenes está girada de manera individual en una o más de una dirección horizontal o una dirección vertical,
- y en el que la información accedida está comprendida en al menos uno de entre una cabecera de segmento, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, una cabecera de unidad de
- 15 capa de abstracción de red y un mensaje de información de mejora complementaria,
- donde la primera imagen de la primera vista no está girada y la segunda imagen de la segunda vista está girada, la primera imagen de la primera vista y la segunda imagen de la segunda vista están dispuestas lado a lado o una encima de otra; y
- 20 - descodificar la imagen de vídeo para proporcionar una representación descodificada de las múltiples imágenes combinadas (824, 826),

en el que la información accedida (824, 826) puede usarse para posprocesar la representación descodificada para girar de manera individual la al menos una de las múltiples imágenes.

25 2. El procedimiento según la reivindicación 1, en el que el acceso a la imagen de vídeo, el acceso a la información y la descodificación de la imagen de vídeo se realizan en un descodificador (1650).

3. Un aparato, caracterizado por que comprende:

- 30 - medios para recibir un flujo de vídeo;
- medios para acceder a una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen (826), donde las múltiples imágenes incluyen una primera imagen de una primera vista de un vídeo de múltiples vistas y una segunda imagen de una segunda vista del vídeo de múltiples vistas, donde la imagen de vídeo forma parte del flujo de vídeo recibido;
- 35 - medios para acceder a información que indica cómo las múltiples imágenes de la imagen de vídeo accedida están combinadas, en el que la información accedida indica que al menos una de las múltiples imágenes está girada de manera individual en una o más de una dirección horizontal o una dirección vertical,
- y en el que la información accedida está comprendida en al menos uno de entre una cabecera de segmento, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, una cabecera de unidad de
- 40 capa de abstracción de red y un mensaje de información de mejora complementaria,
- donde la primera imagen de la primera vista no está girada y la segunda imagen de la segunda vista está girada, la primera imagen de la primera vista y la segunda imagen de la segunda vista están dispuestas lado a lado o una encima de otra; y
- 45 - medios para descodificar la imagen de vídeo para proporcionar una representación descodificada de las múltiples imágenes combinadas (824, 826),

en el que la información accedida (824, 826) puede usarse para posprocesar la representación descodificada para girar de manera individual la al menos una de las múltiples imágenes.

50 4. El aparato según la reivindicación 3, en el que los medios para acceder a la imagen de vídeo, acceder a la información y descodificar la imagen de vídeo están incluidos en un descodificador (1650).

5. Un medio legible por ordenador que presenta un código de programa legible por ordenador grabado en el mismo, donde el código de programa legible por ordenador comprende:

- 55 - código de programa para acceder a una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen, donde la imagen forma parte de un flujo de vídeo recibido (826), las múltiples imágenes

incluyen una primera imagen de una primera vista de un vídeo de múltiples vistas y una segunda imagen de una segunda vista del vídeo de múltiples vistas;

- código de programa para acceder a información que indica cómo las múltiples imágenes de la imagen de vídeo accedida están combinadas, en el que la información accedida indica que al menos una de las múltiples imágenes está girada de manera individual en una o más de una dirección horizontal o una dirección vertical,

y en el que la información accedida está comprendida en al menos uno de entre una cabecera de segmento, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, una cabecera de unidad de capa de abstracción de red y un mensaje de información de mejora complementaria;

donde la primera imagen de la primera vista no está girada y la segunda imagen de la segunda vista está girada, la primera imagen de la primera vista y la segunda imagen de la segunda vista están dispuestas lado a lado o una encima de otra; y

- código de programa para descodificar la imagen de vídeo para proporcionar una representación descodificada de las múltiples imágenes combinadas (824, 826),

en el que la información accedida (824, 826) puede usarse para posprocesar la representación descodificada para girar de manera individual la al menos una de las múltiples imágenes.

6. El medio legible por ordenador según la reivindicación 5, en el que el acceso a la imagen de vídeo, el acceso a la información y la descodificación de la imagen de vídeo se realizan en un descodificador (1650).

7. Un procedimiento, caracterizado por que comprende:

- generar una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen, donde las múltiples imágenes incluyen una primera imagen de una primera vista de un vídeo de múltiples vistas y una segunda imagen de una segunda vista del vídeo de múltiples vistas;

- generar información que indica cómo las múltiples imágenes de la imagen de vídeo generada están combinadas, en el que la información generada indica que al menos una de las múltiples imágenes está girada de manera individual en una o más de una dirección horizontal o una dirección vertical,

y en el que la información generada está comprendida en al menos uno de entre una cabecera de segmento, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, una cabecera de unidad de capa de abstracción de red y un mensaje de información de mejora complementaria,

donde la primera imagen de la primera vista no está girada y la segunda imagen de la segunda vista está girada, la primera imagen de la primera vista y la segunda imagen de la segunda vista están dispuestas lado a lado o una encima de otra;

- codificar la imagen de vídeo generada y la información generada; y

- proporcionar un flujo de vídeo que incluye la imagen de vídeo codificada y la información codificada.

8. El procedimiento según la reivindicación 7, en el que la generación de la imagen de vídeo, la generación de la información, la codificación y la provisión del flujo de vídeo se realizan en un codificador.

9. Un aparato, caracterizado por que comprende:

- medios para generar una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen, donde las múltiples imágenes incluyen una primera imagen de una primera vista de un vídeo de múltiples vistas y una segunda imagen de una segunda vista del vídeo de múltiples vistas;

- medios para generar información que indica cómo las múltiples imágenes de la imagen de vídeo generada están combinadas, en el que la información generada indica que al menos una de las múltiples imágenes está girada de manera individual en una o más de una dirección horizontal o una dirección vertical,

y en el que la información generada está comprendida en al menos uno de entre una cabecera de segmento, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, una cabecera de unidad de capa de abstracción de red y un mensaje de información de mejora complementaria,

donde la primera imagen de la primera vista no está girada y la segunda imagen de la segunda vista está girada, la primera imagen de la primera vista y la segunda imagen de la segunda vista están dispuestas lado a lado o una encima de otra;

- medios para codificar la imagen de vídeo generada y la información generada; y

- medios para proporcionar un flujo de vídeo que incluye la imagen de vídeo codificada y la información codificada.

5 10. El aparato según la reivindicación 9, en el que los medios para generar la imagen de vídeo, generar la información, codificar y proporcionar el flujo de vídeo están incluidos en un codificador.

11. Un medio legible por ordenador que presenta un código de programa legible por ordenador grabado en el mismo, donde el código de programa legible por ordenador comprende:

10 - código de programa para generar una imagen de vídeo que incluye múltiples imágenes combinadas en una única imagen, donde las múltiples imágenes incluyen una primera imagen de una primera vista de un vídeo de múltiples vistas y una segunda imagen de una segunda vista del vídeo de múltiples vistas;

15 - código de programa para generar información que indica cómo las múltiples imágenes de la imagen de vídeo generada están combinadas, en el que la información generada indica que al menos una de las múltiples imágenes está girada de manera individual en una o más de una dirección horizontal o una dirección vertical,
y en el que la información generada está comprendida en al menos uno de entre una cabecera de segmento, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, una cabecera de unidad de capa de abstracción de red y un mensaje de información de mejora complementaria,

20 donde la primera imagen de la primera vista no está girada y la segunda imagen de la segunda vista está girada, la primera imagen de la primera vista y la segunda imagen de la segunda vista están dispuestas lado a lado o una encima de otra;

- código de programa para codificar la imagen de vídeo generada y la información generada; y

25 - código de programa para proporcionar un flujo de vídeo que incluye la imagen de vídeo codificada y la información codificada.

12. El medio legible por ordenador según la reivindicación 11, en el que la generación de la imagen de vídeo, la generación de la información, la codificación y la provisión del flujo de vídeo se realizan en un codificador.

100



FIG. 1

200

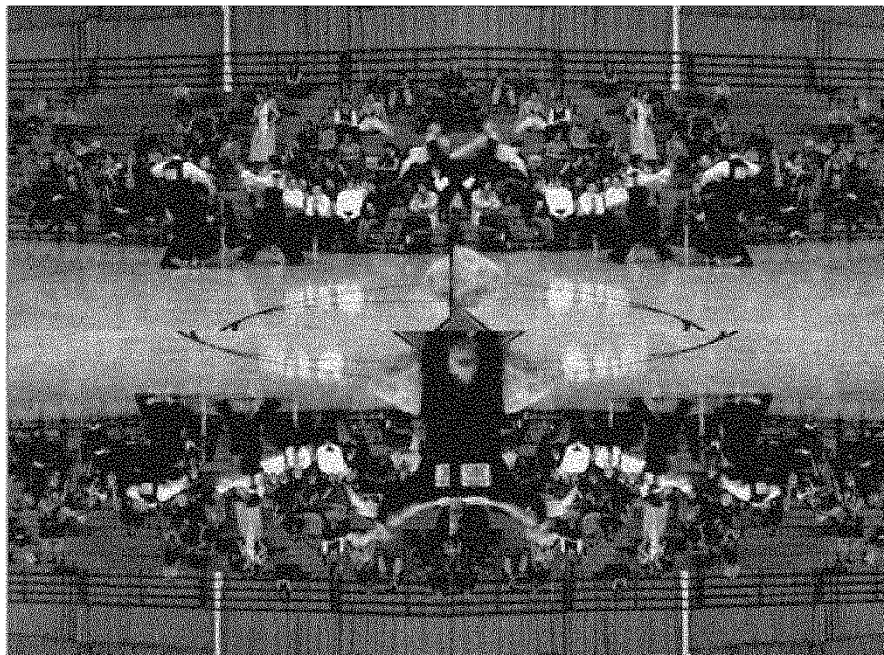


FIG. 2

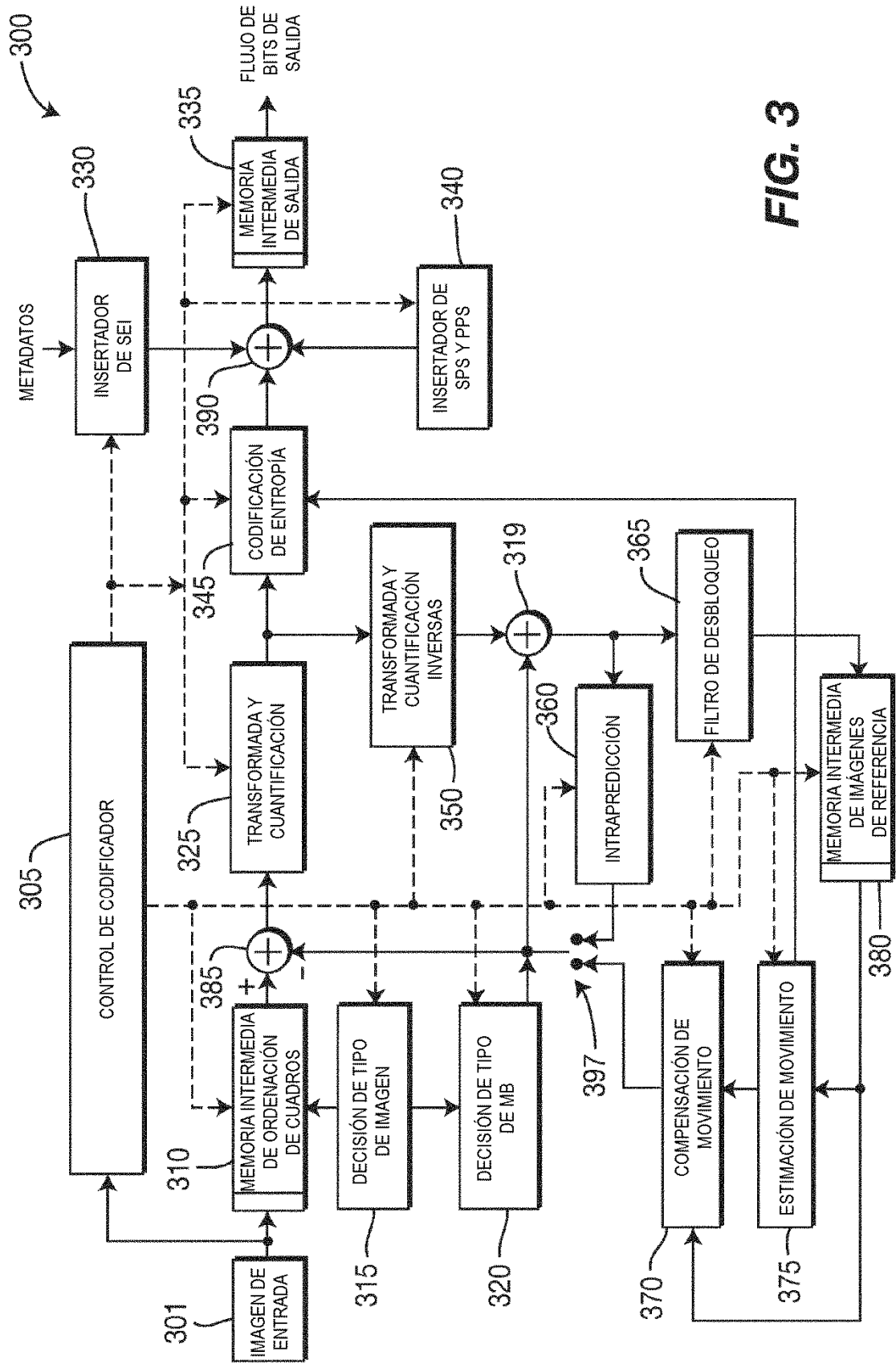


FIG. 3

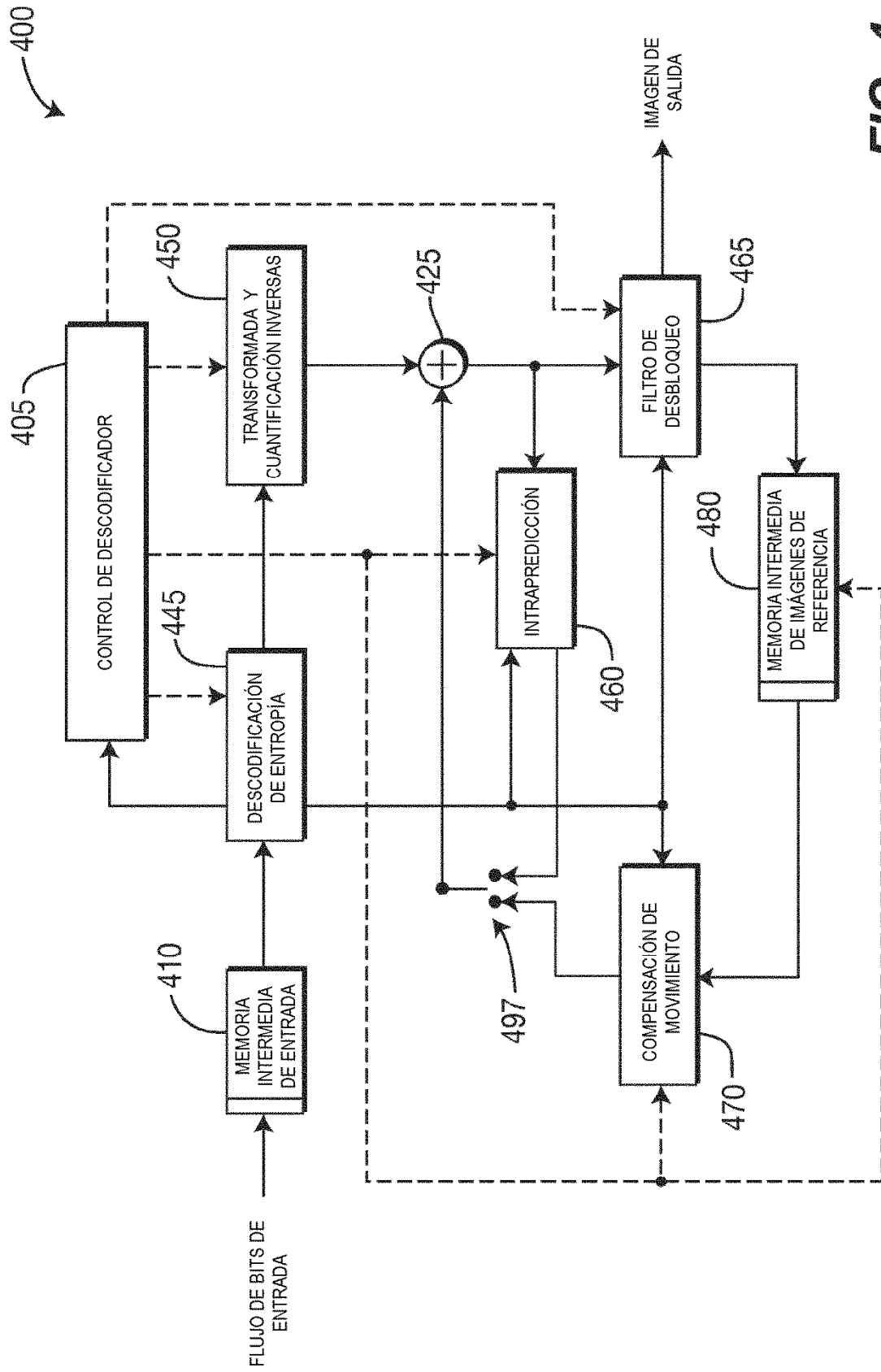


FIG. 4

FIG. 5A

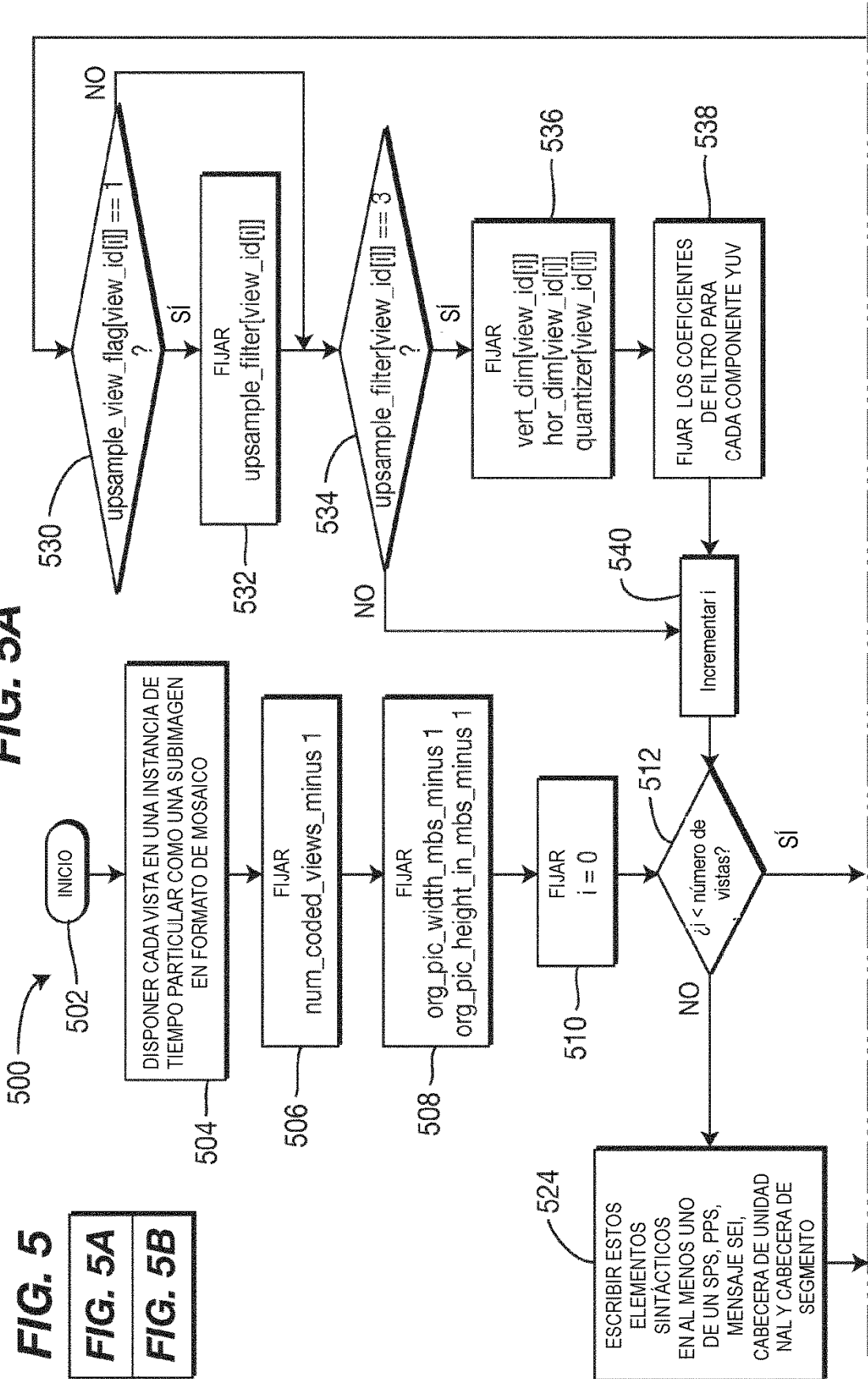


FIG. 5

FIG. 5A

FIG. 5B

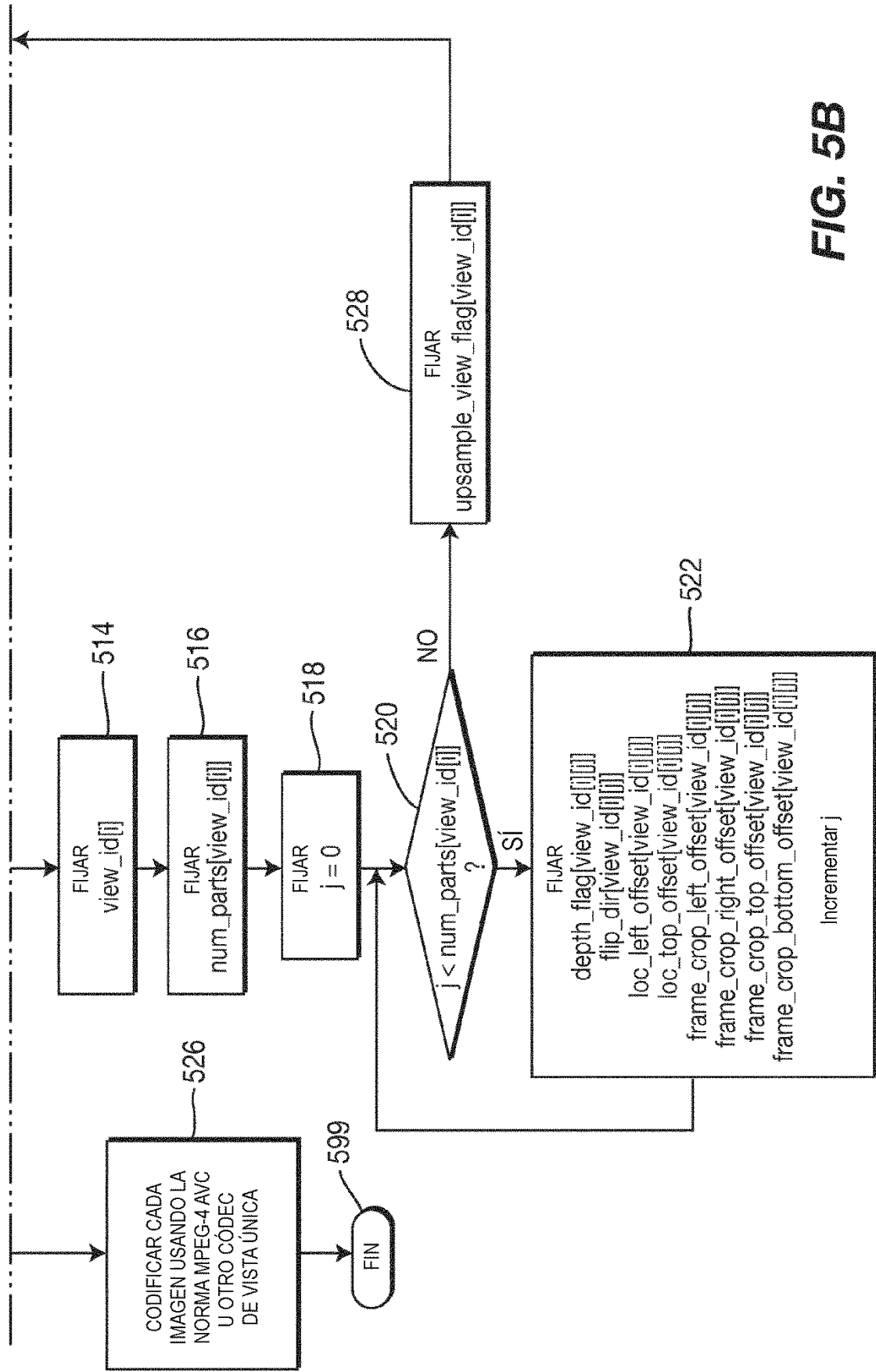
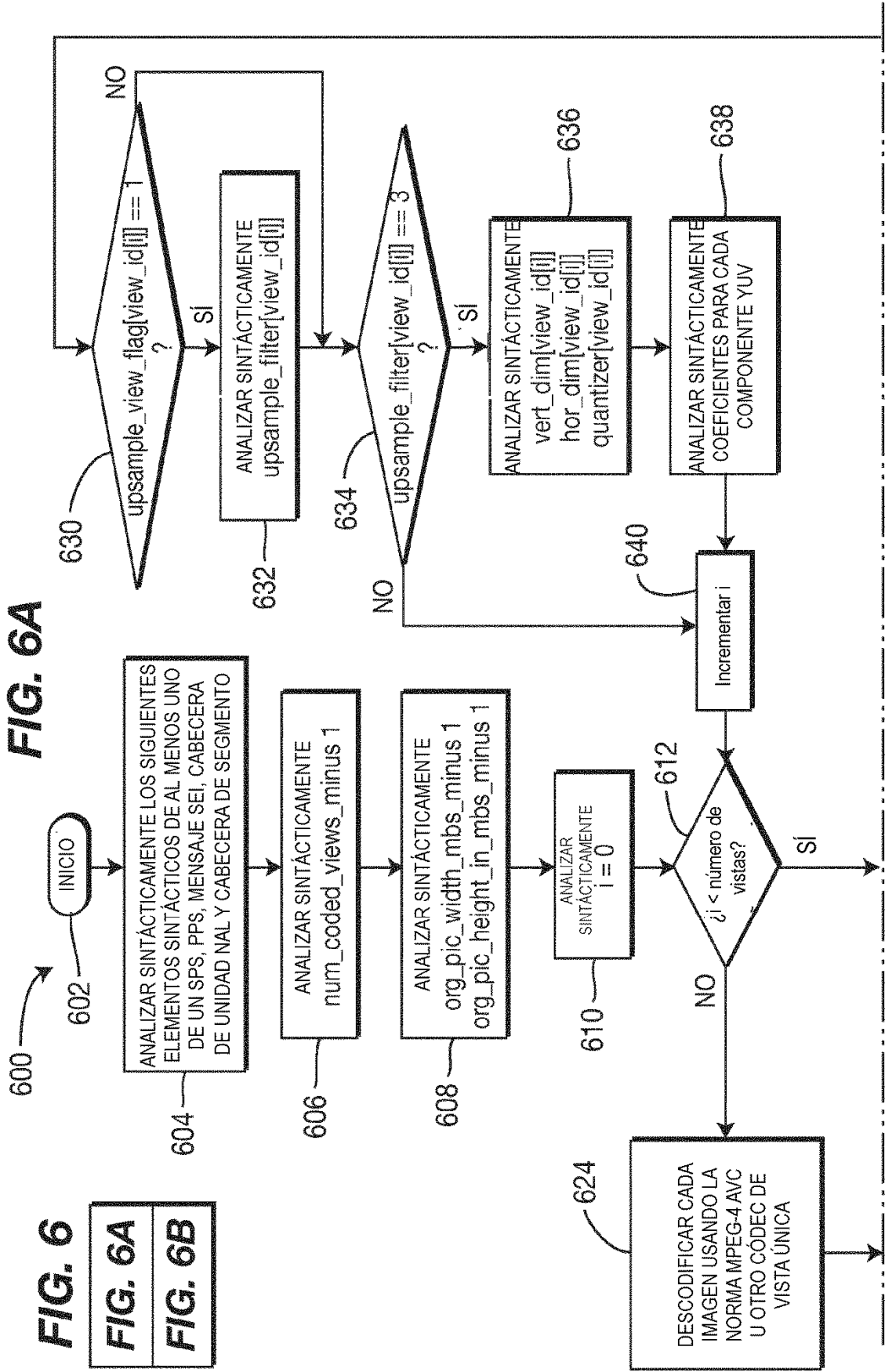


FIG. 5B



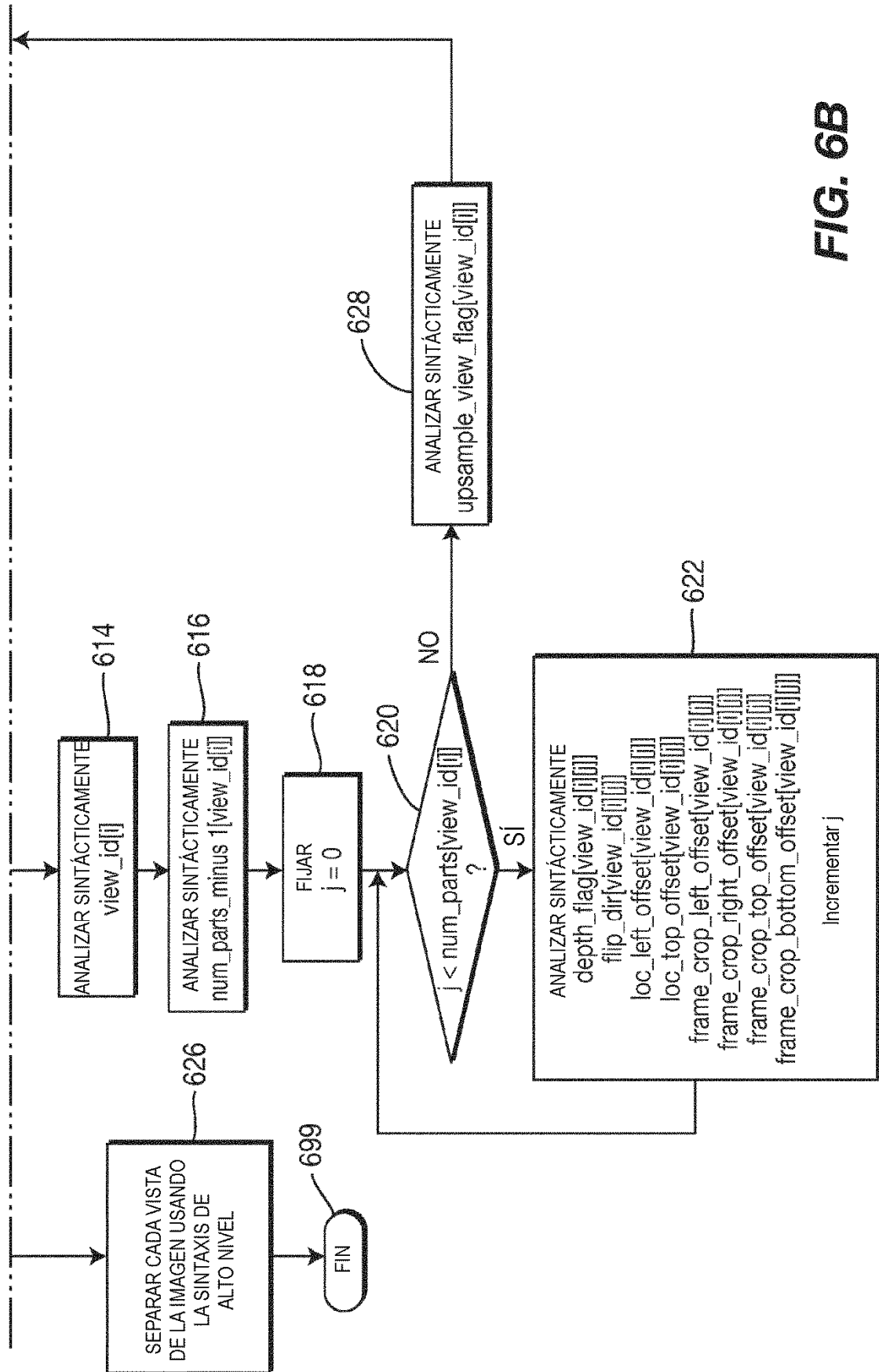


FIG. 6B

FIG. 7A

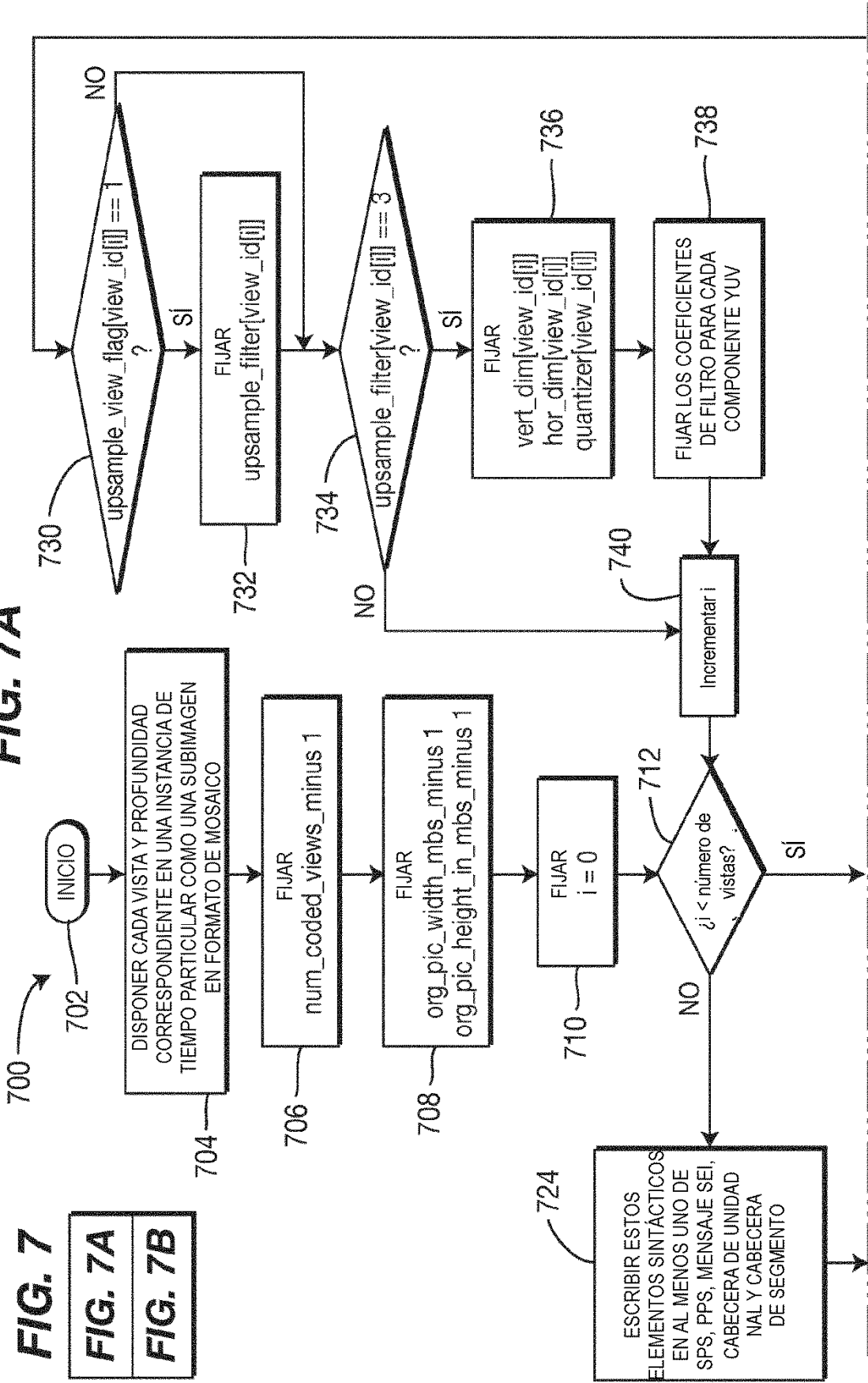


FIG. 7

FIG. 7A

FIG. 7B

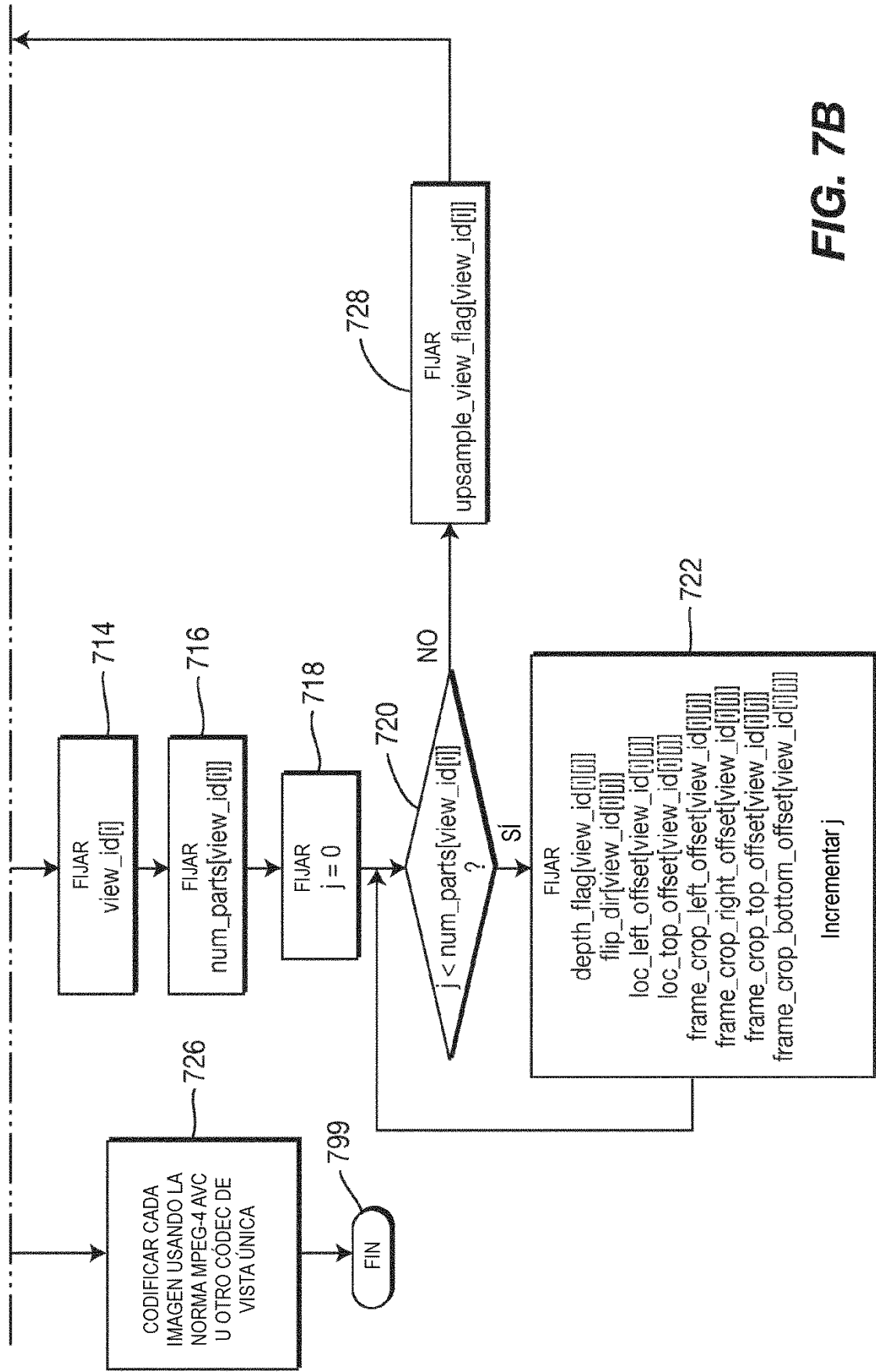
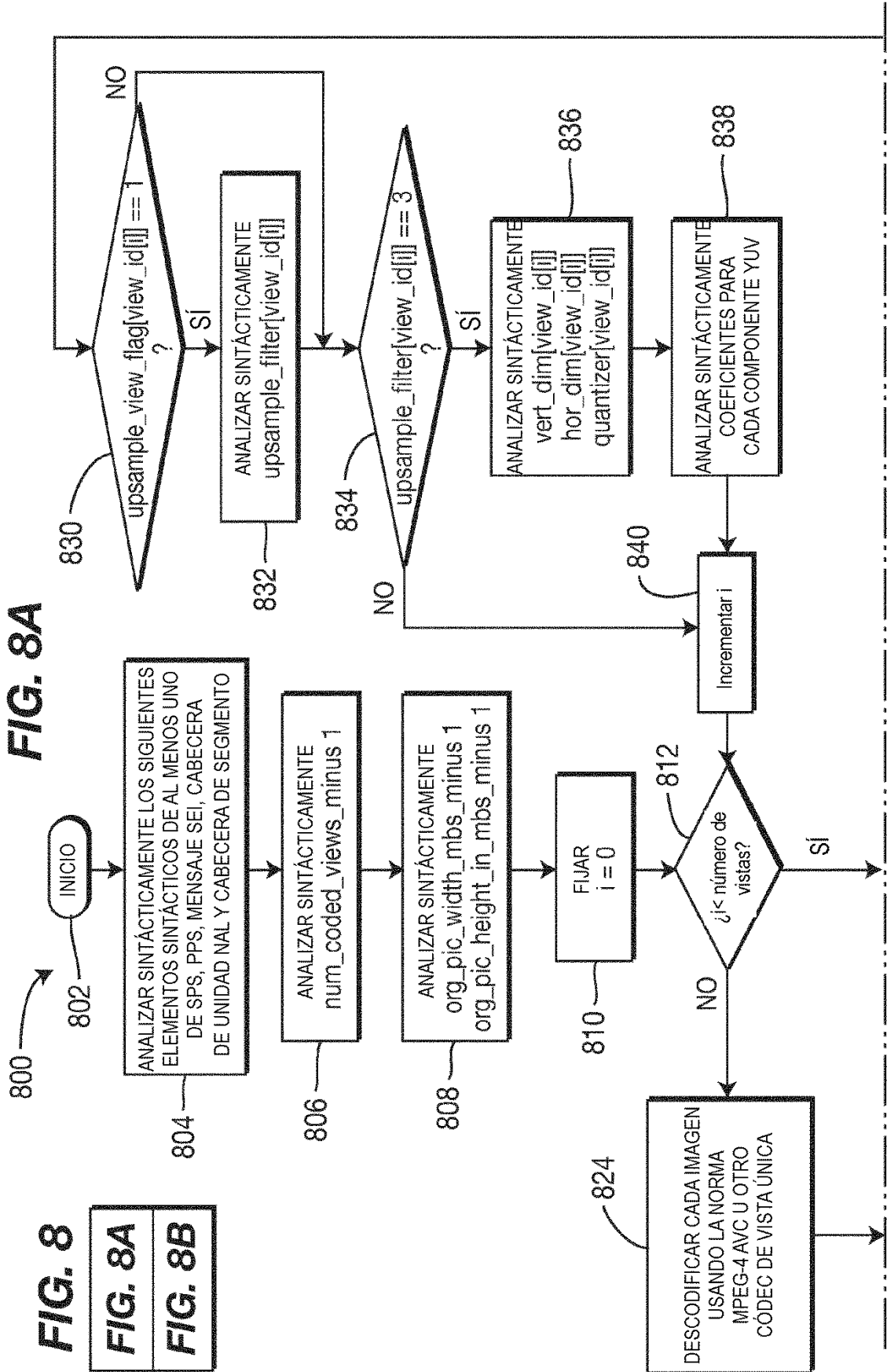


FIG. 7B



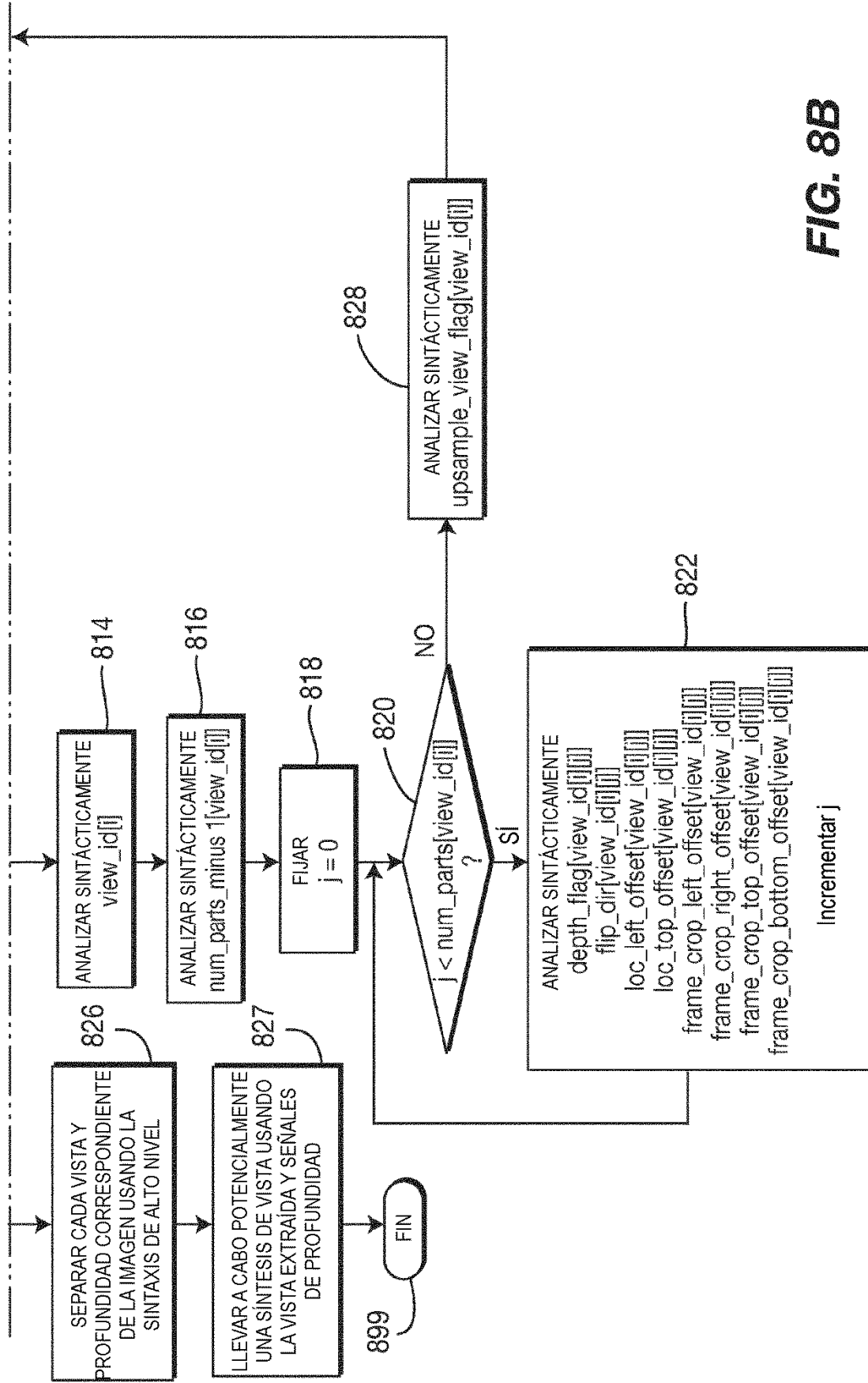


FIG. 8B

900

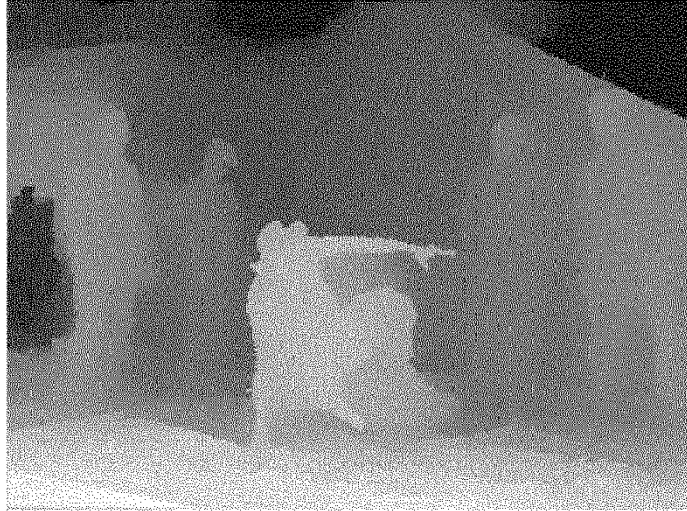


FIG. 9

1000



FIG. 10

1100

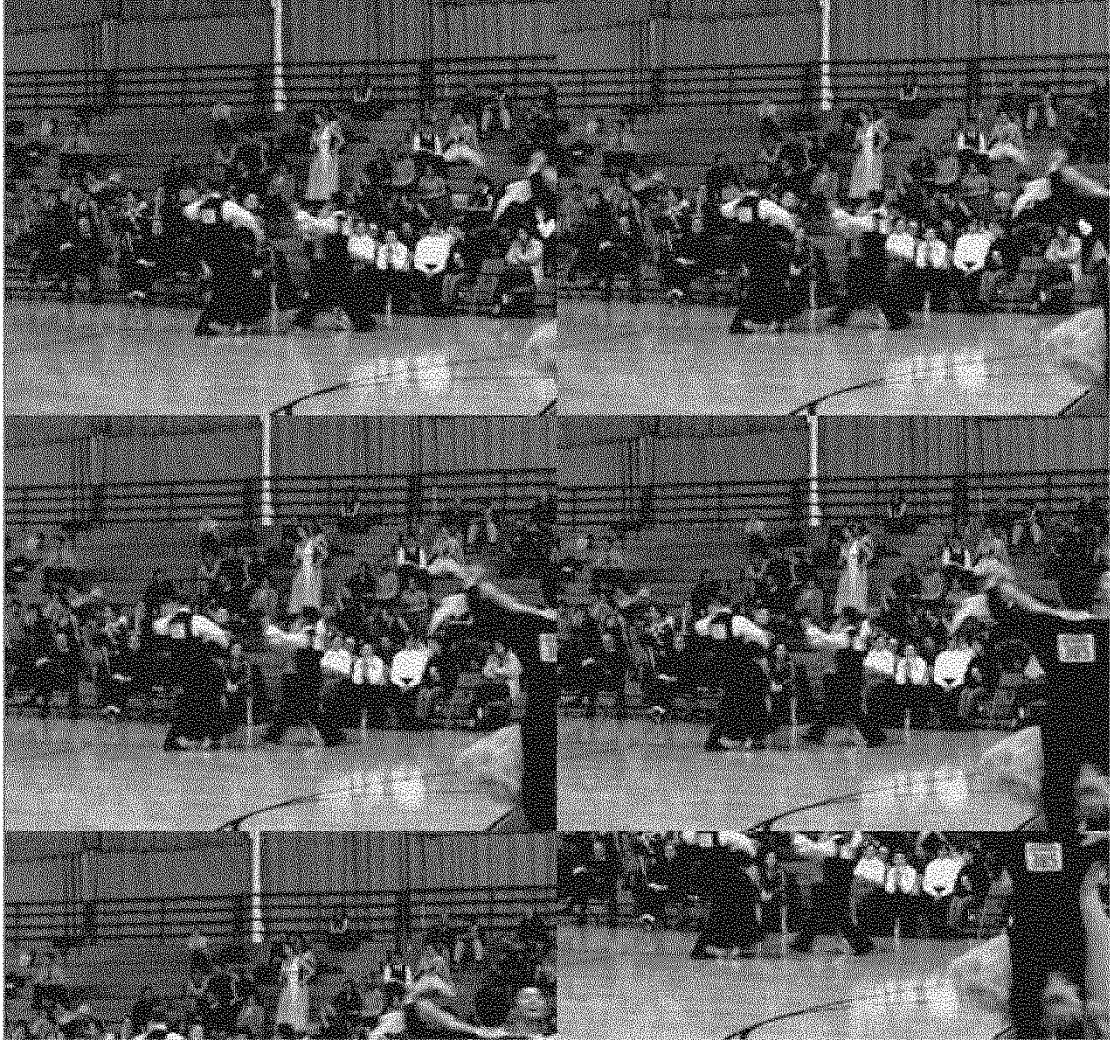


FIG. 11

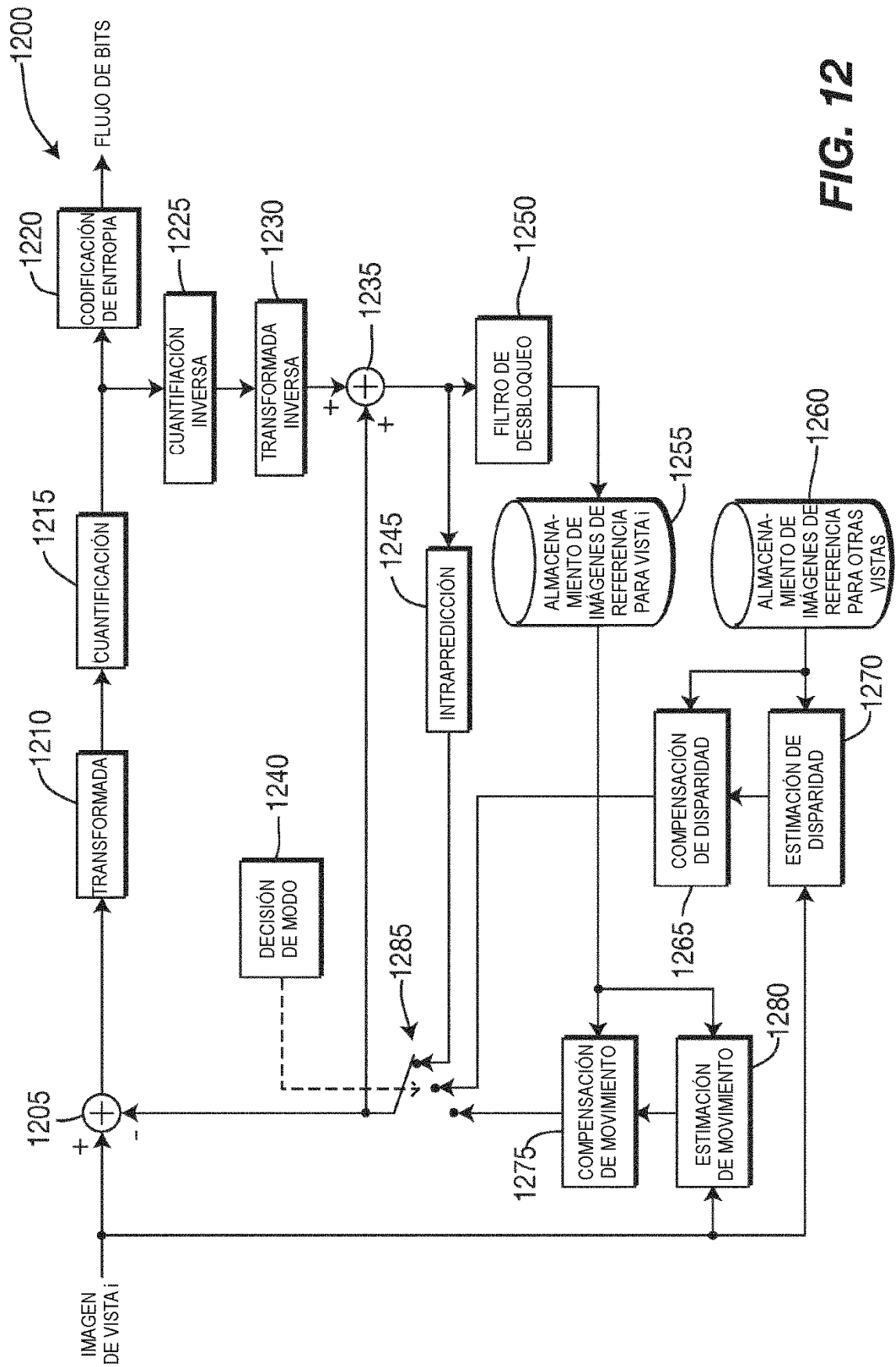


FIG. 12

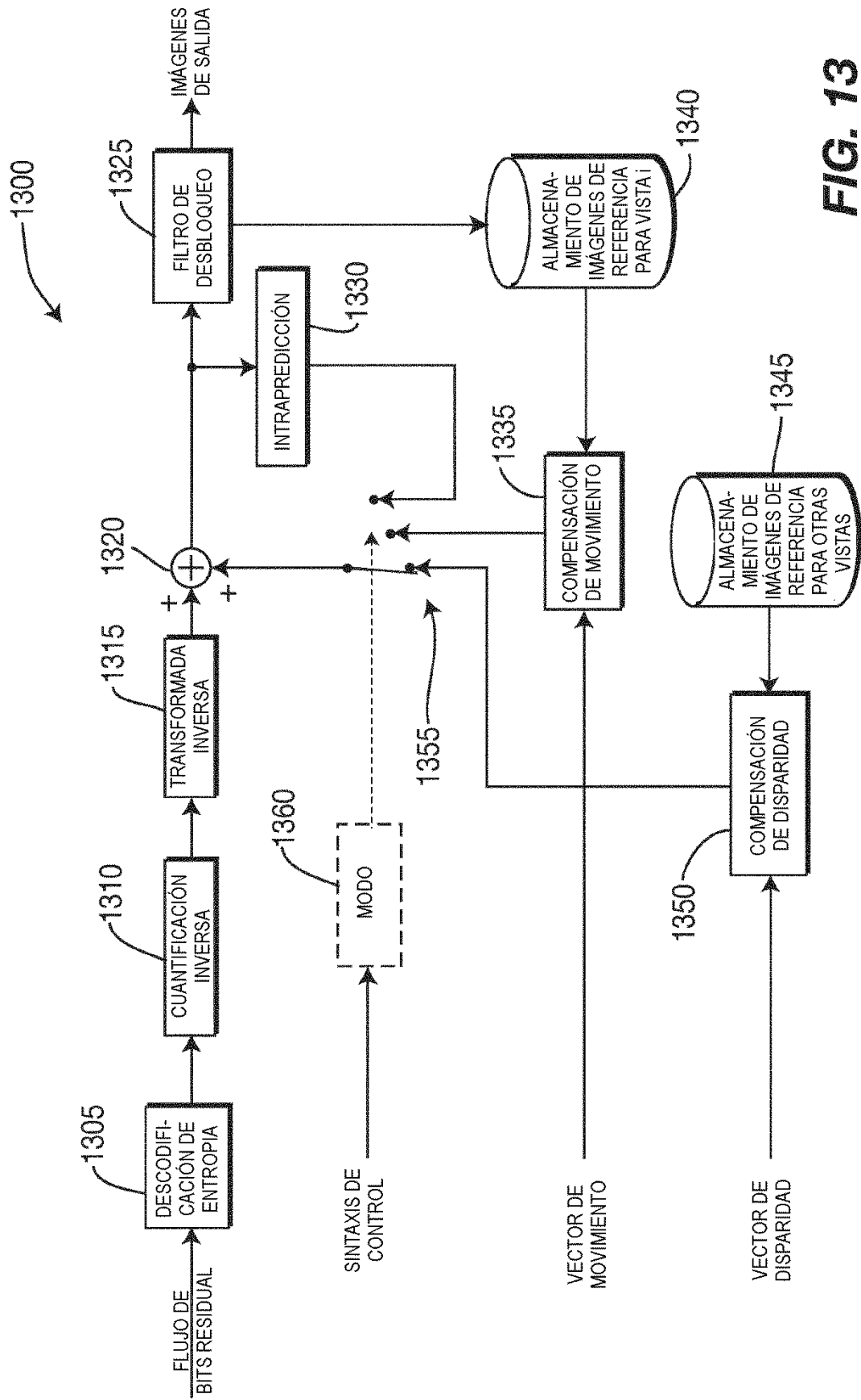


FIG. 13

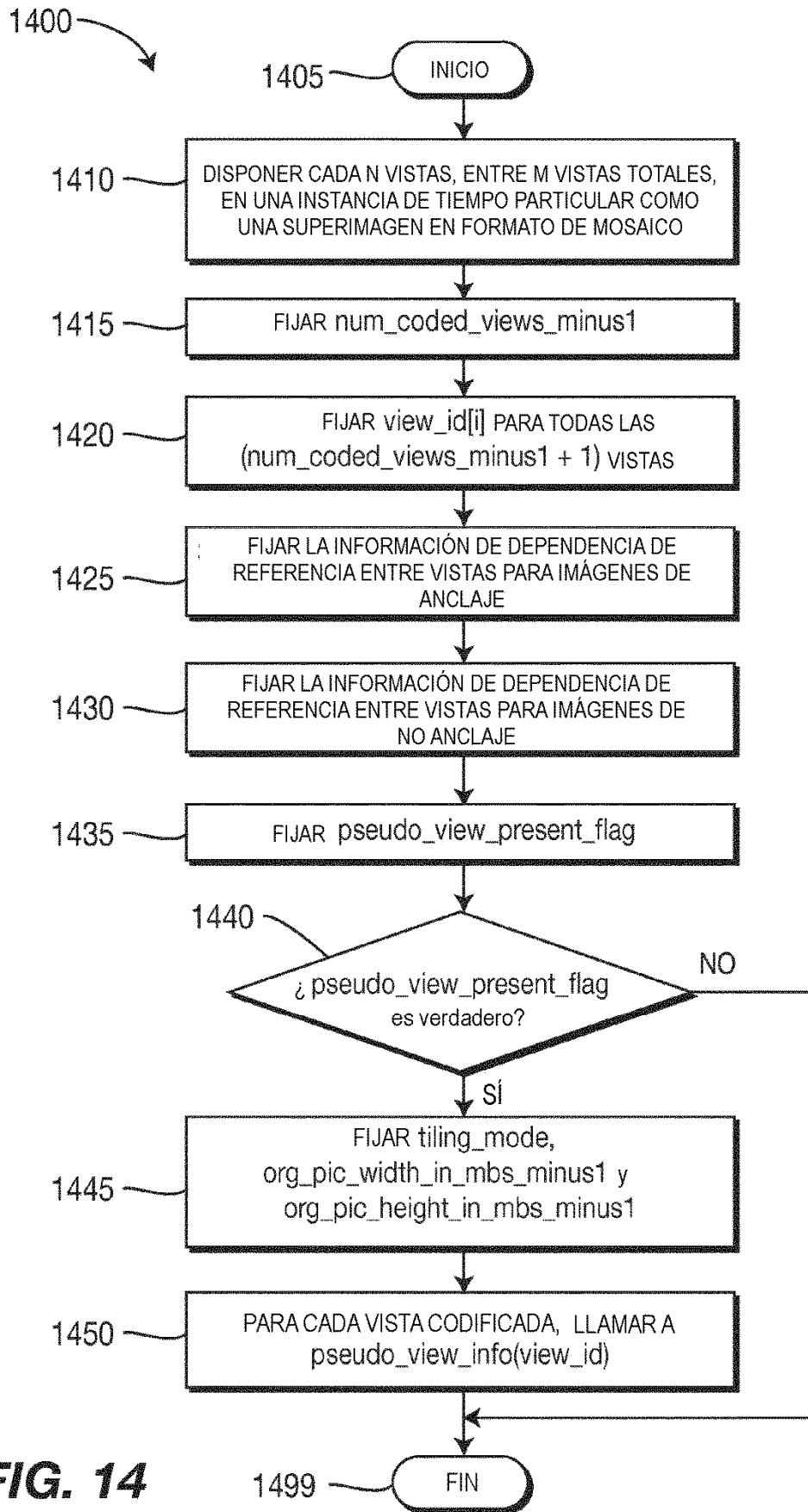
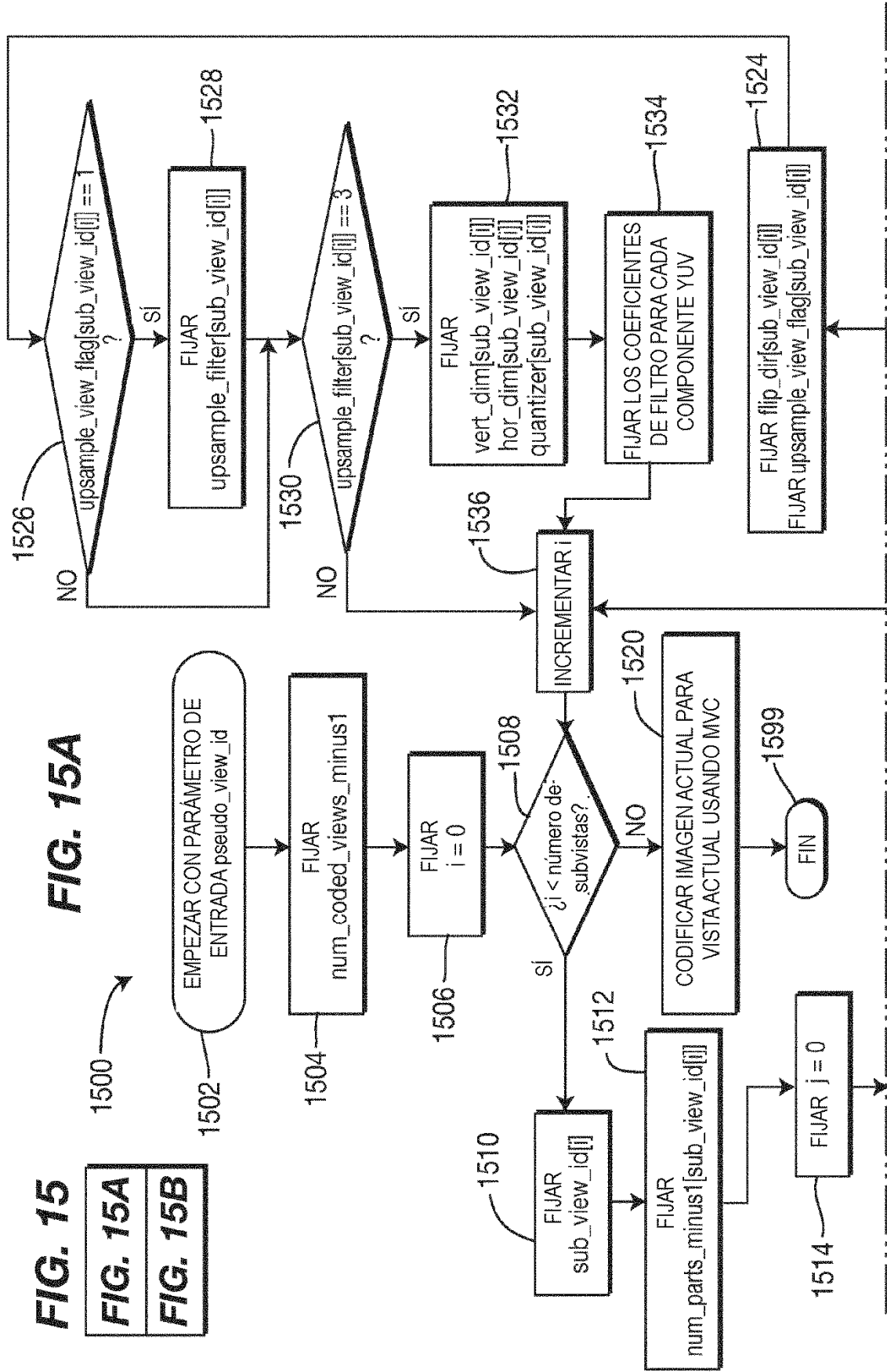


FIG. 14

FIG. 15

FIG. 15A

FIG. 15B



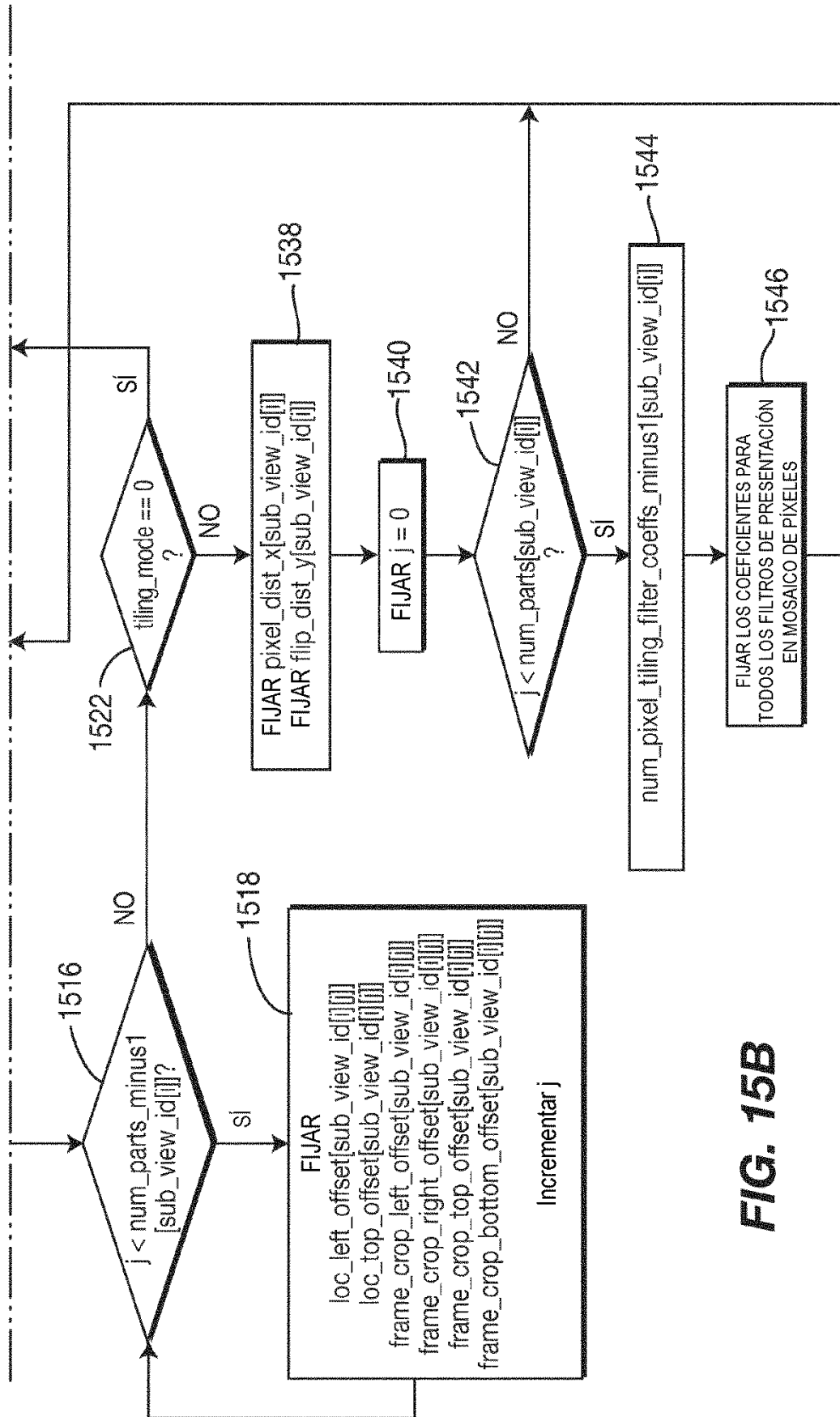


FIG. 15B

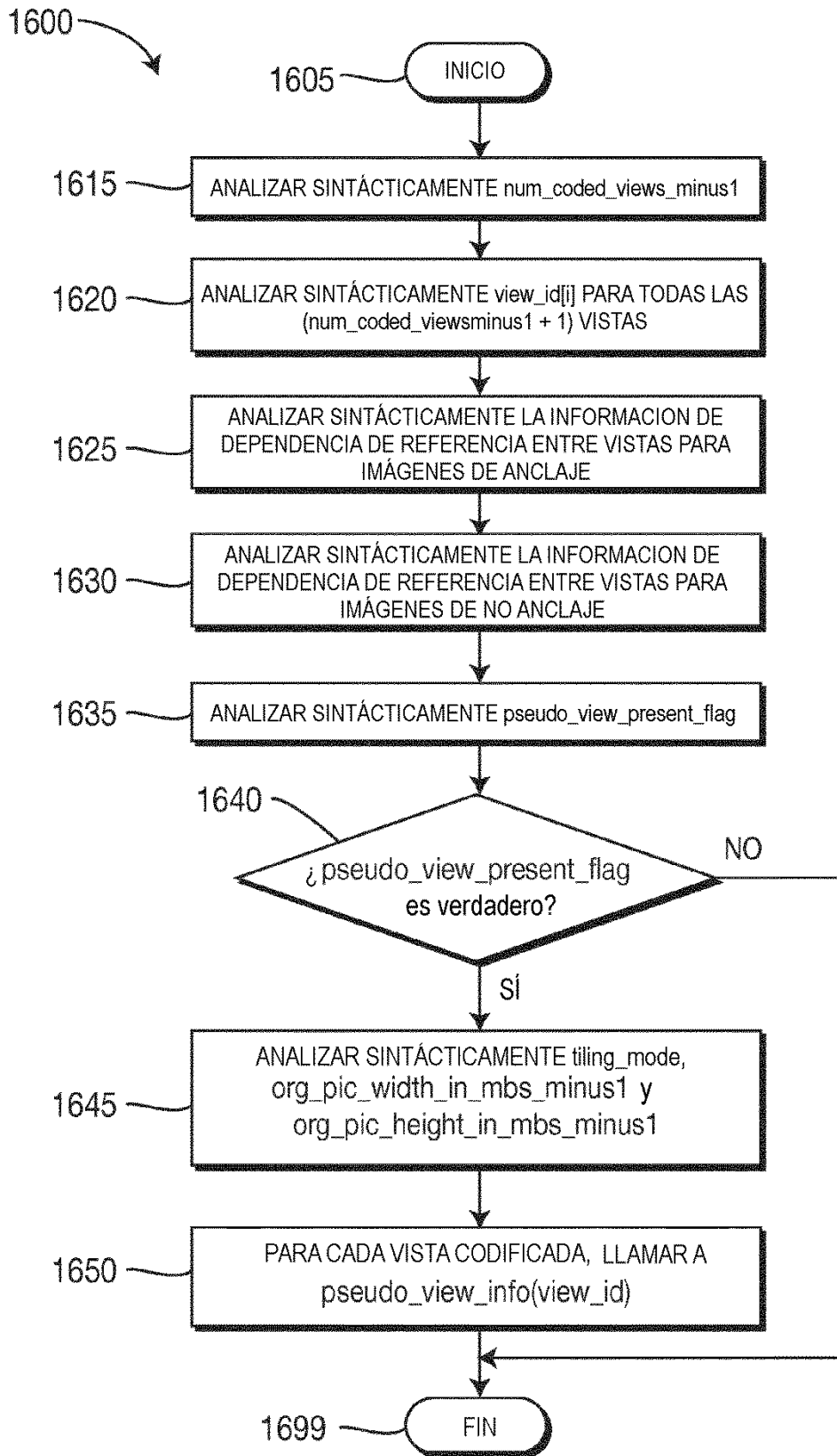


FIG. 16

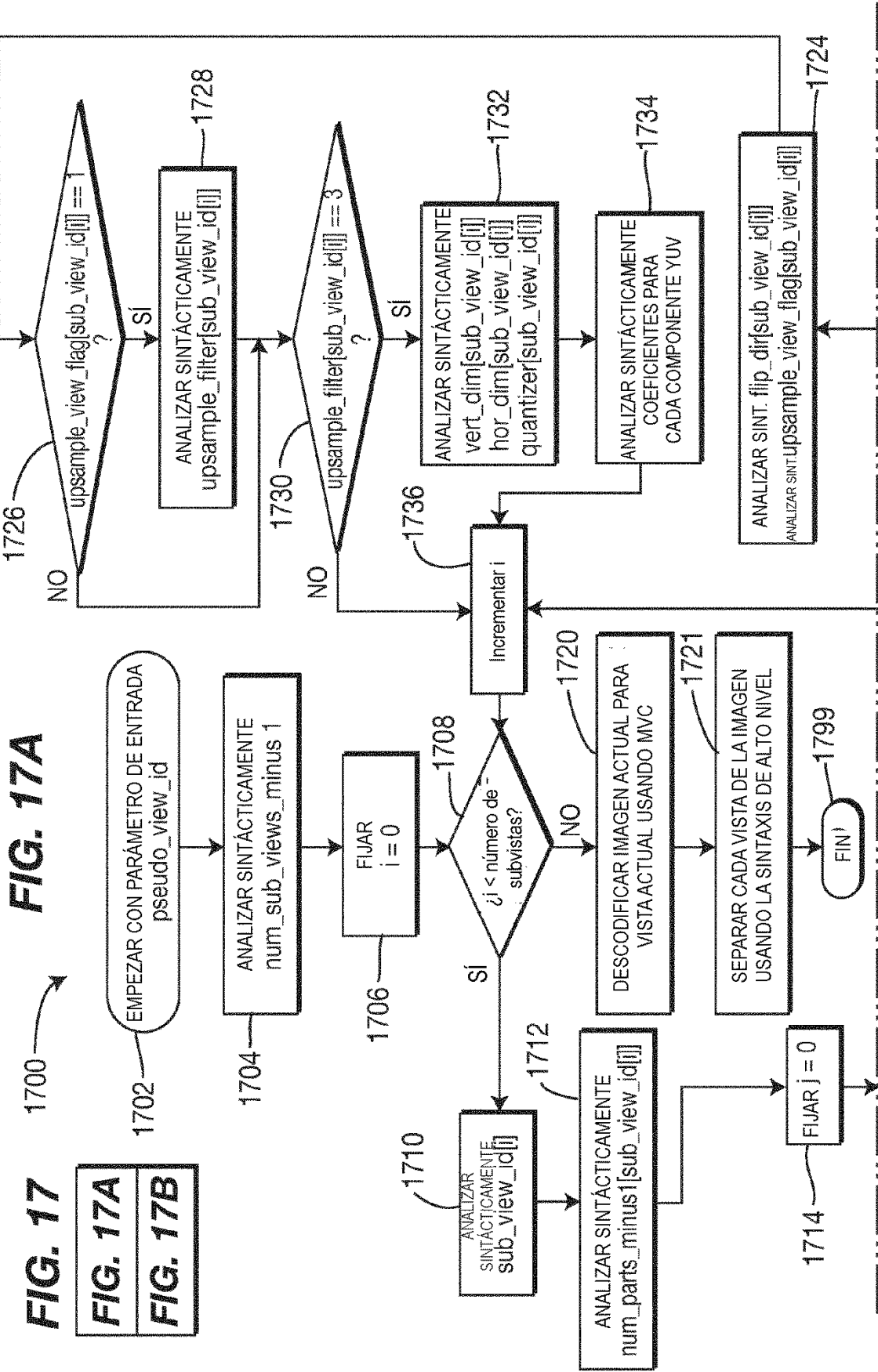


FIG. 17A

FIG. 17

FIG. 17A

FIG. 17B

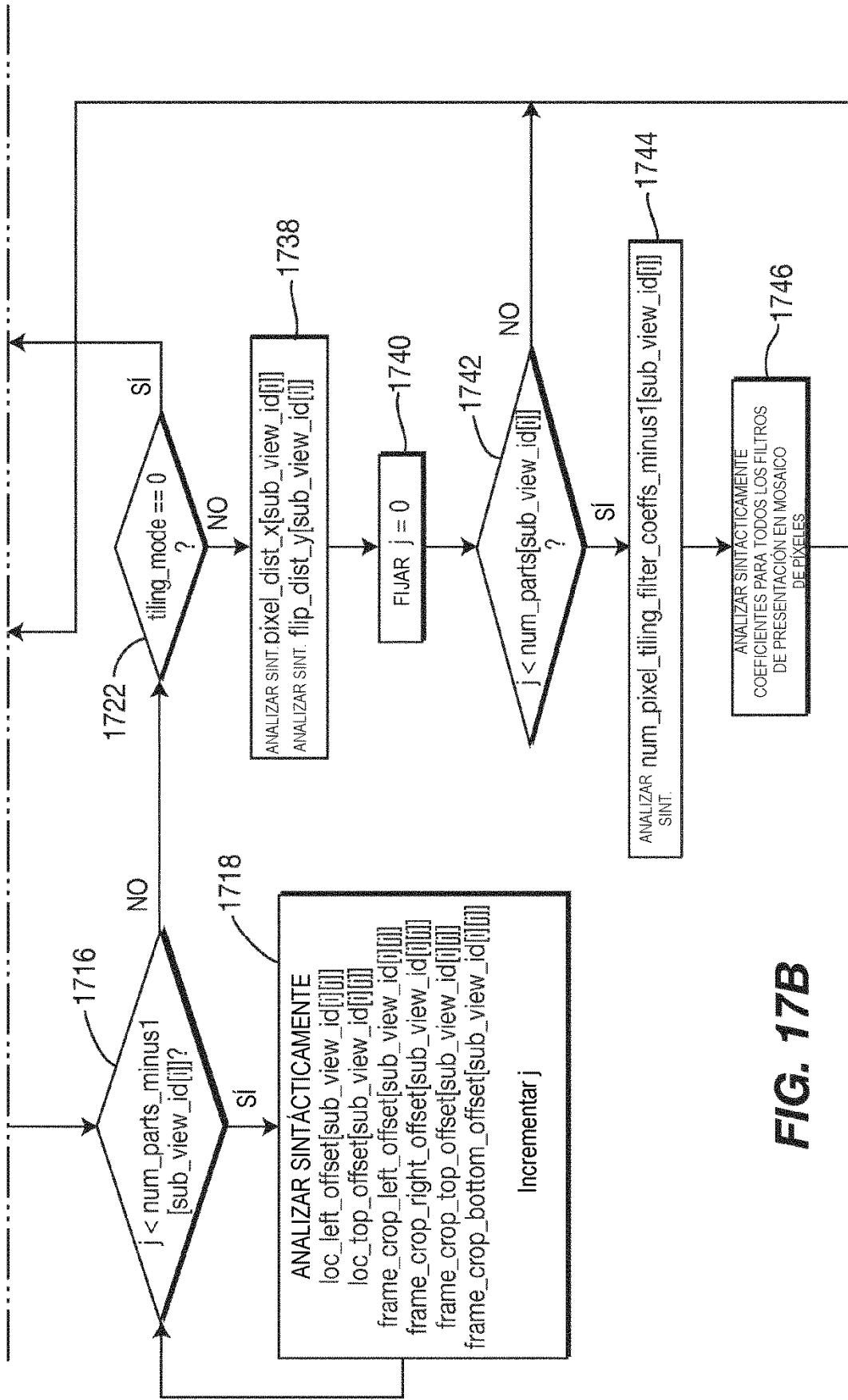


FIG. 17B

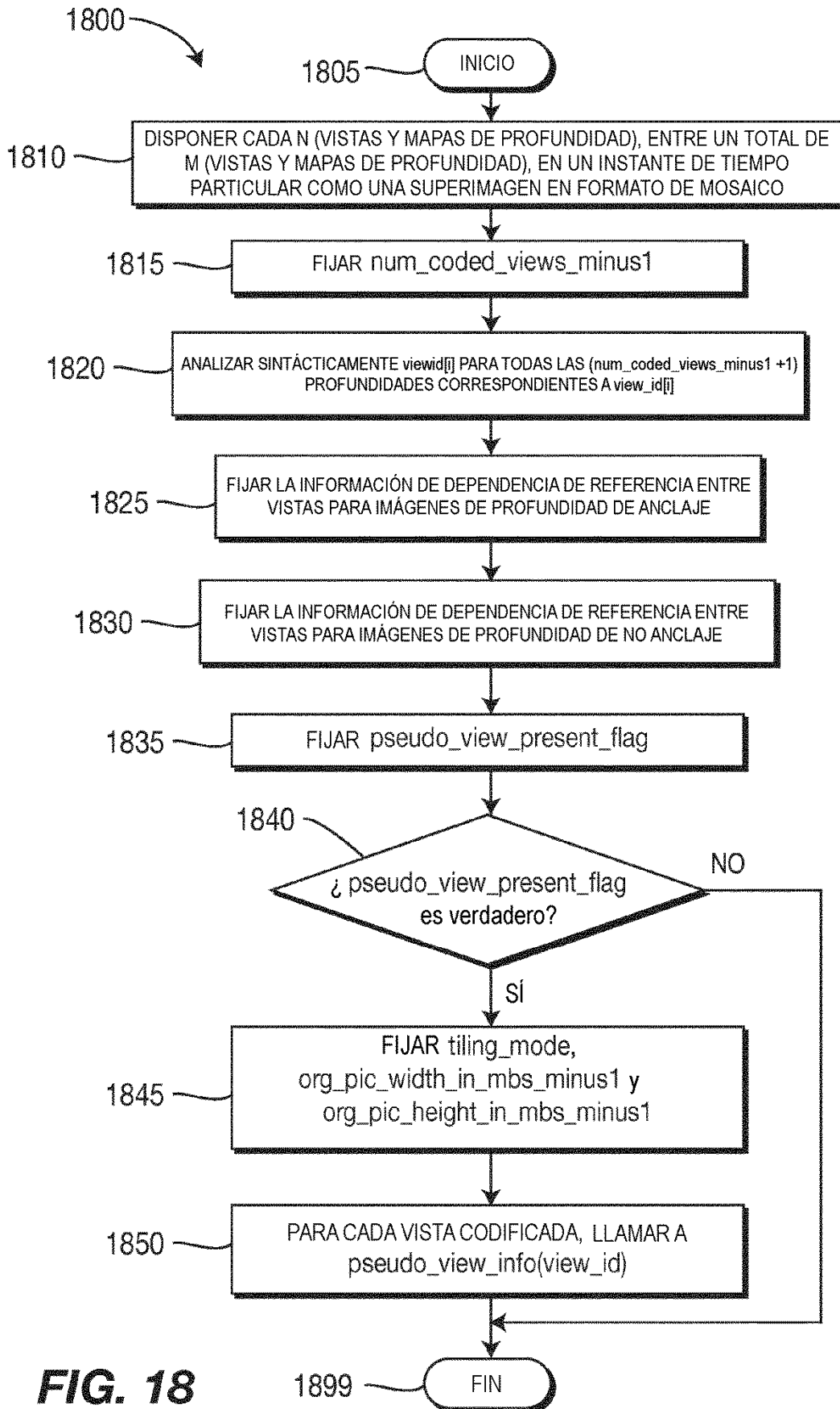
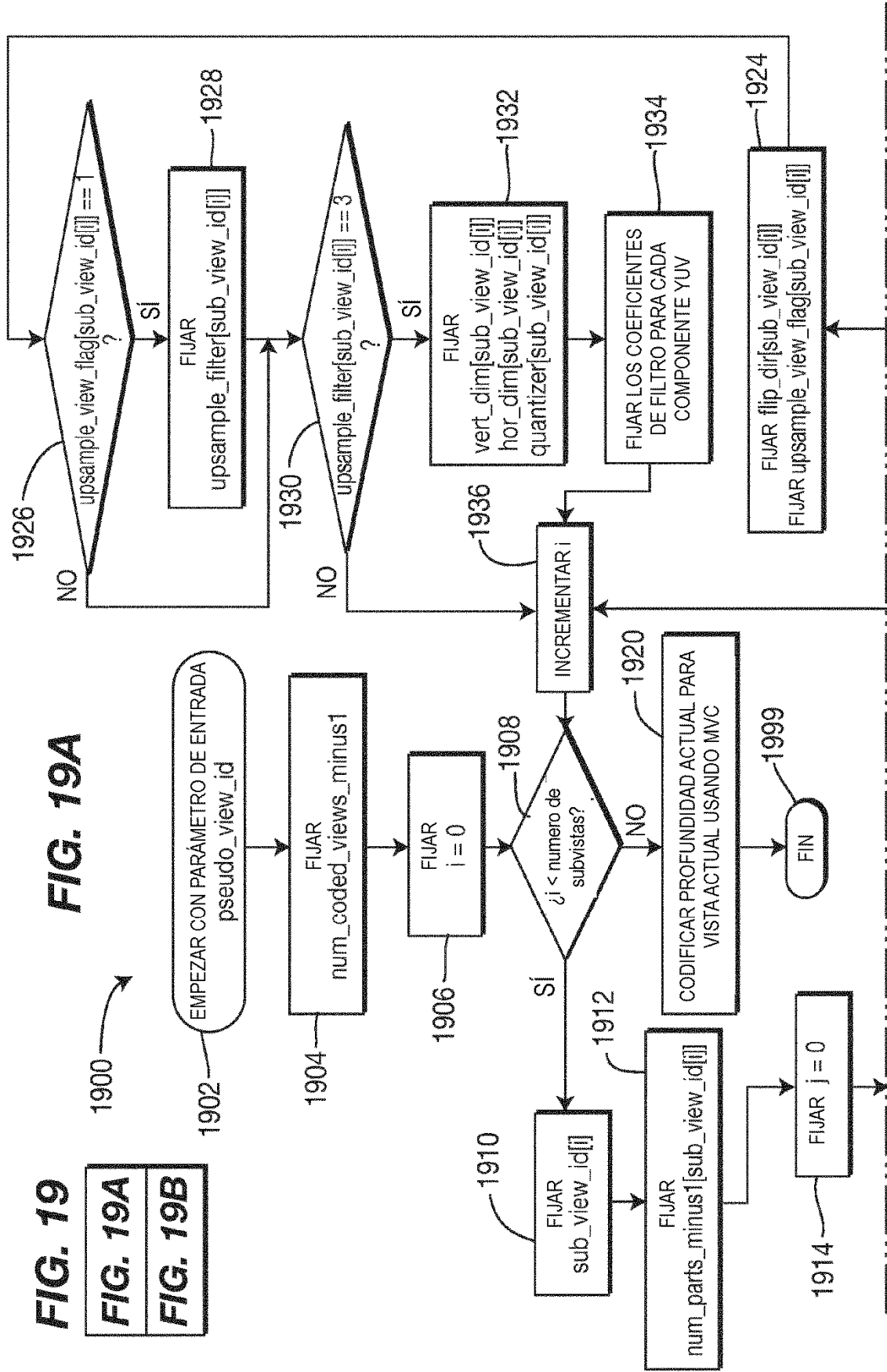


FIG. 18

FIG. 19

FIG. 19A

FIG. 19B



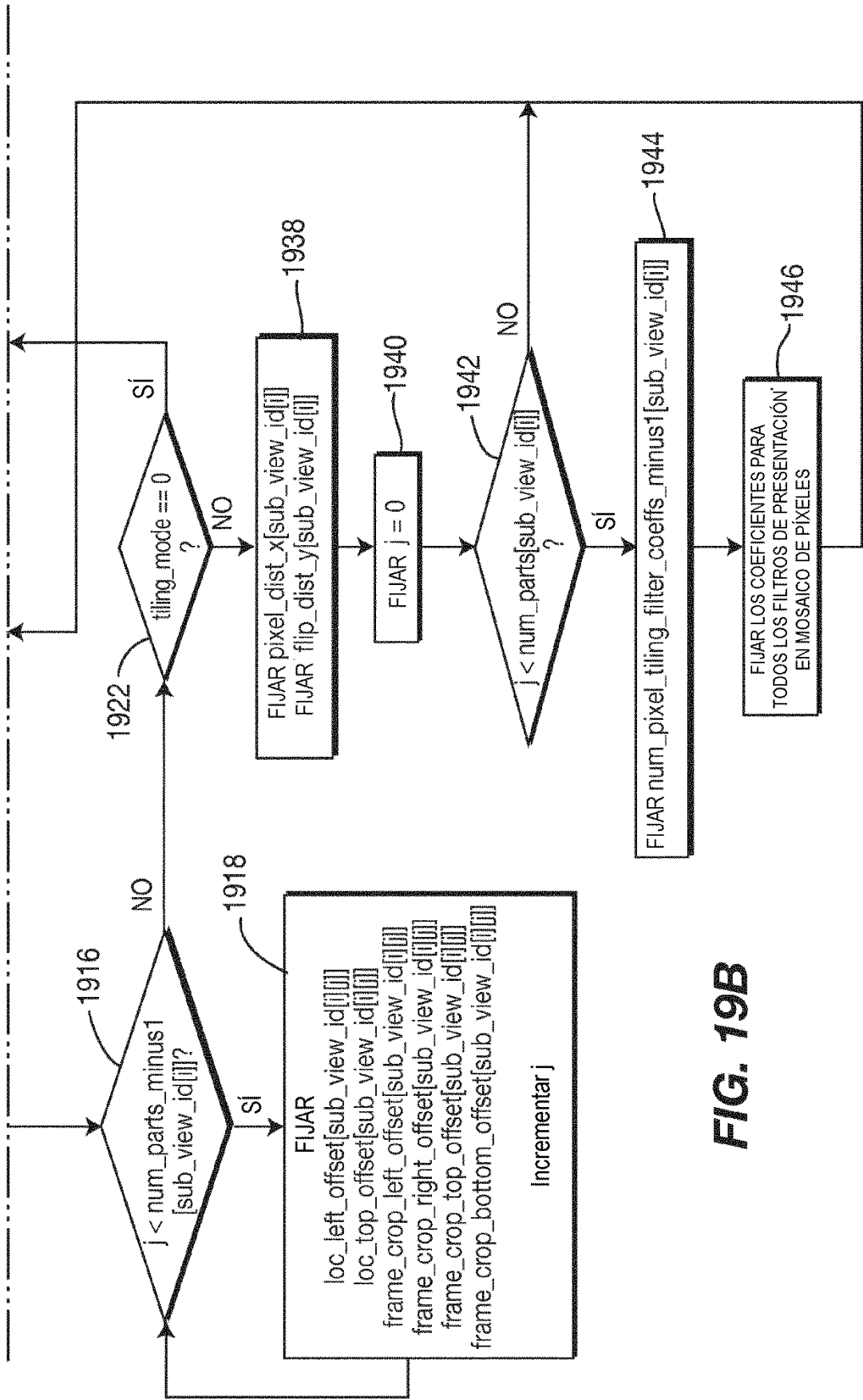


FIG. 19B

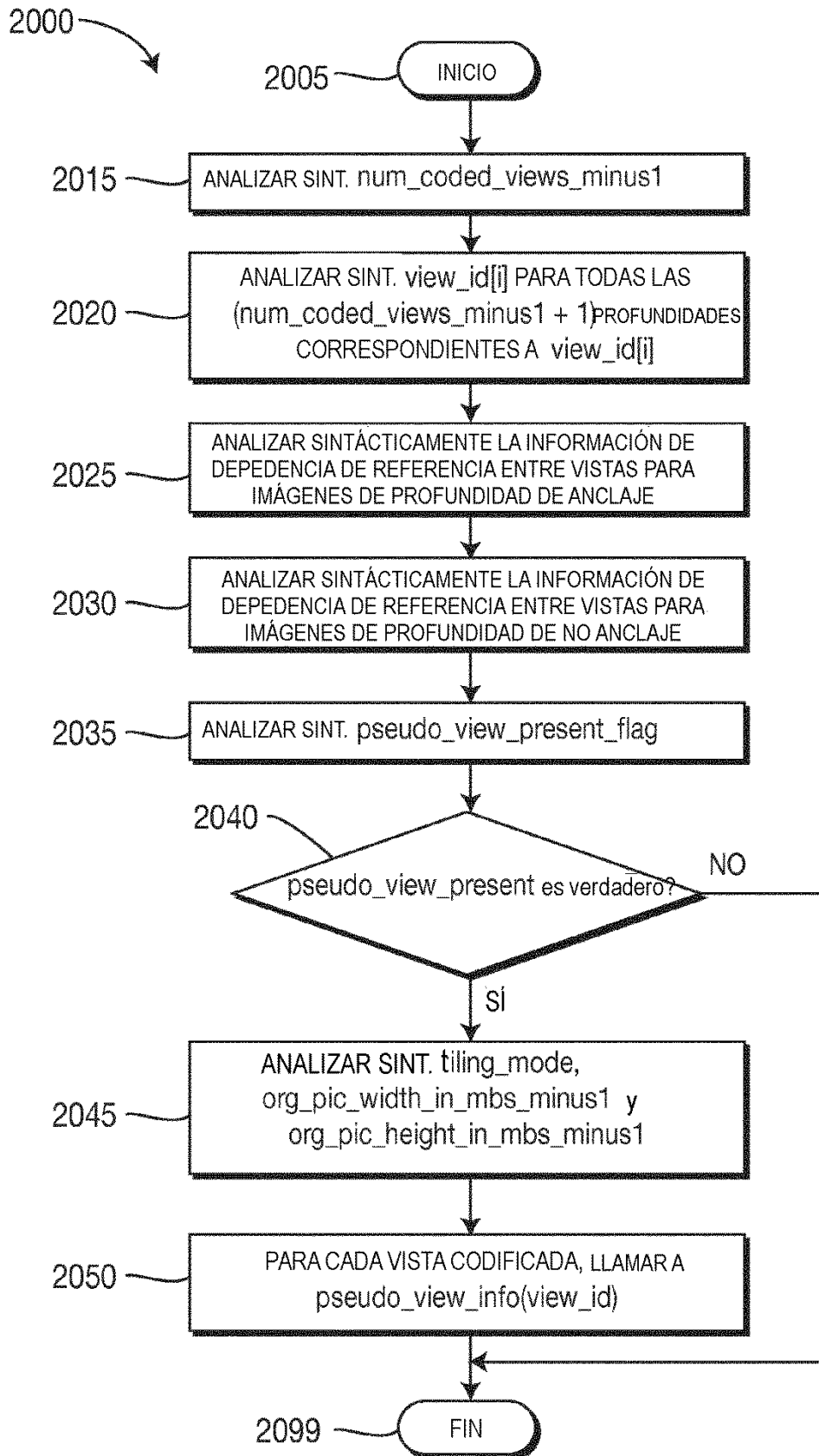


FIG. 20

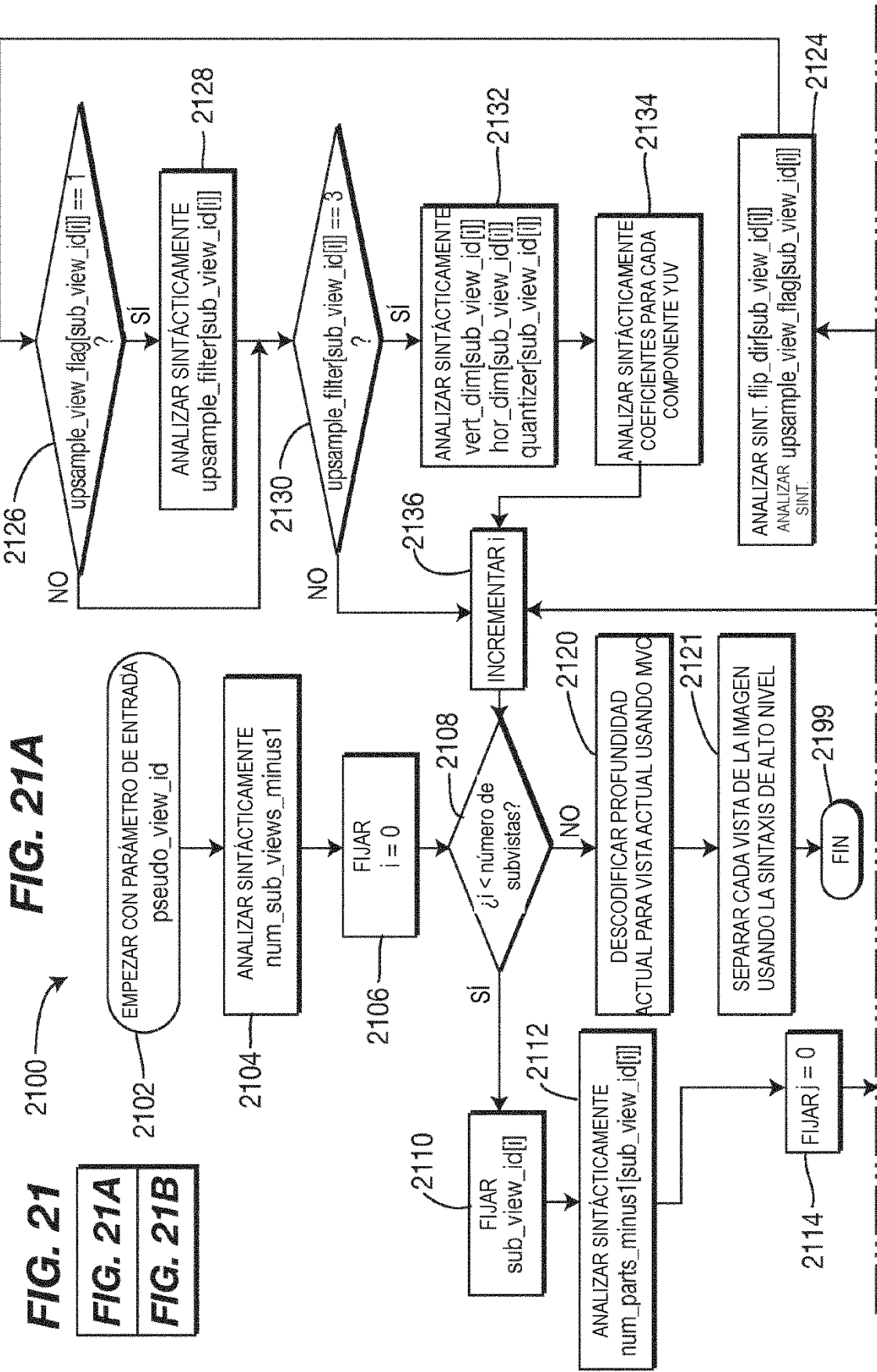


FIG. 21A

FIG. 21

FIG. 21A

FIG. 21B

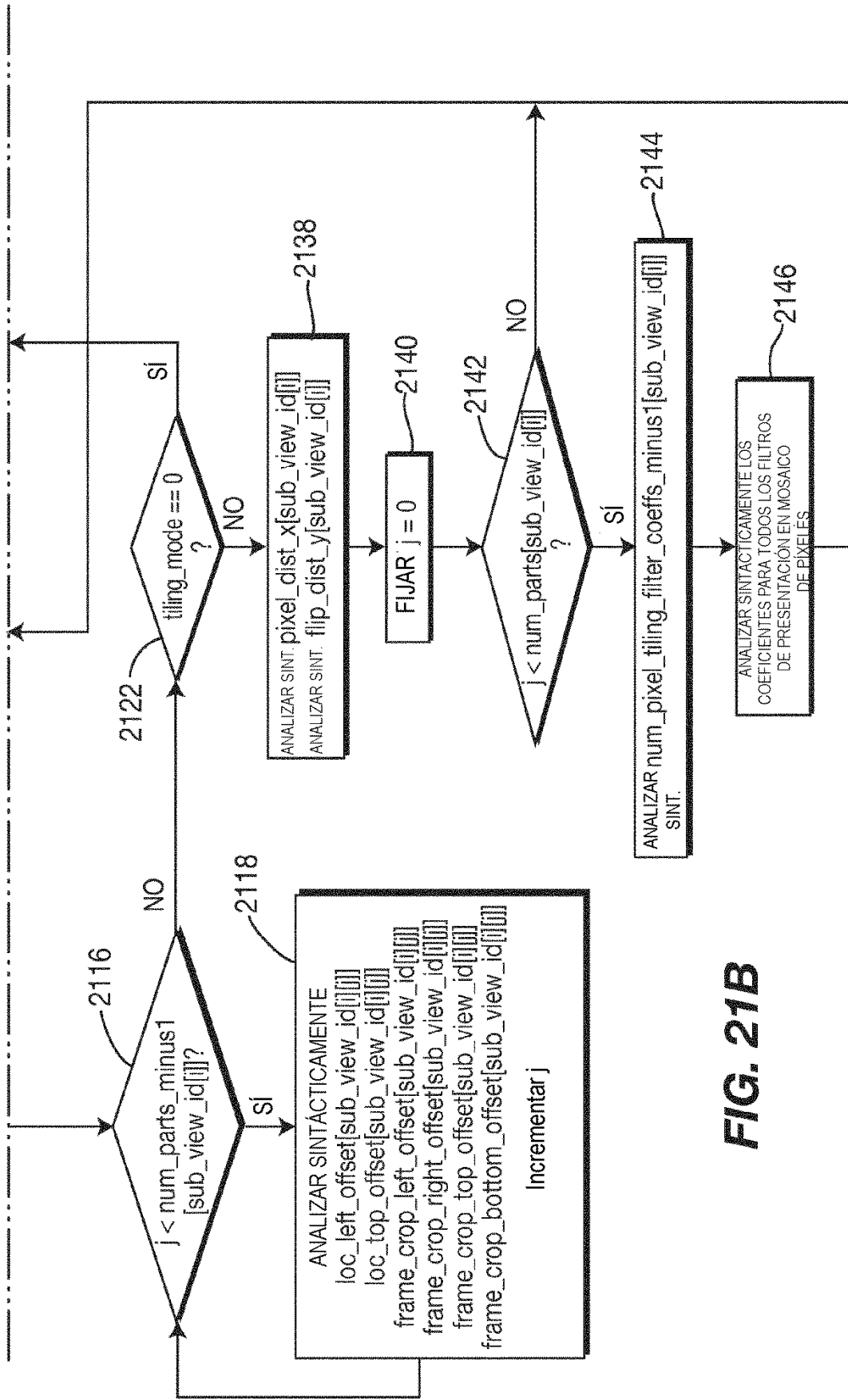


FIG. 21B

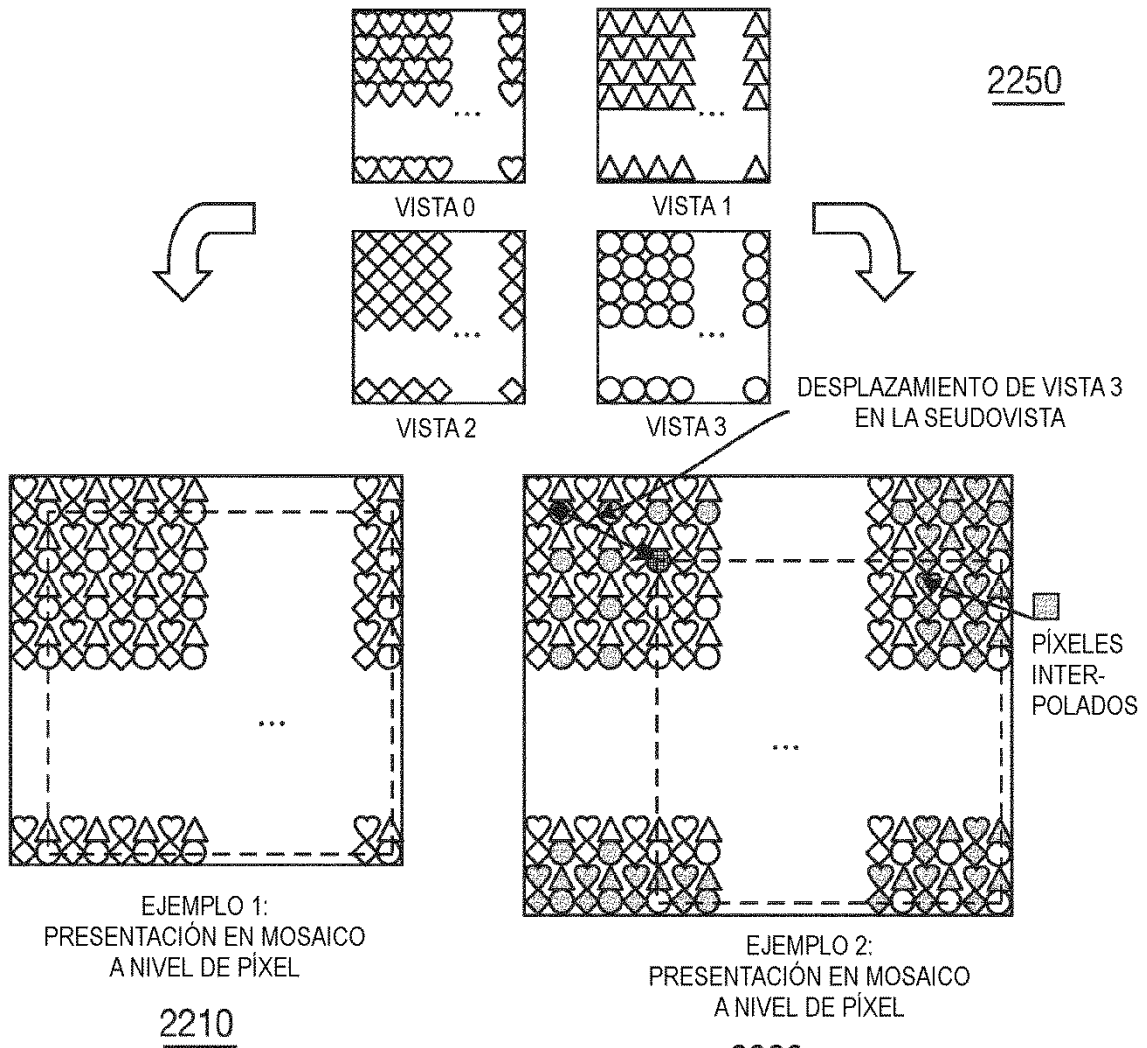


FIG. 22

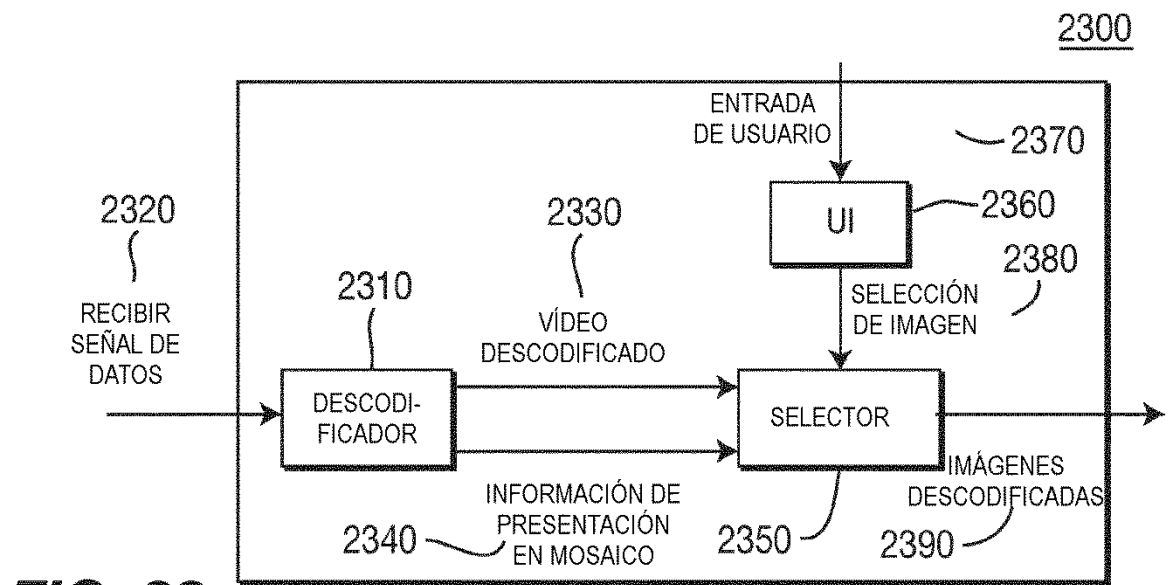


FIG. 23