

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 586 406**

51 Int. Cl.:

H04N 19/70 (2014.01)

H04N 13/00 (2006.01)

H04N 19/46 (2014.01)

H04N 19/61 (2014.01)

H04N 19/597 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **11.04.2008 E 12159060 (8)**

97 Fecha y número de publicación de la concesión europea: **01.06.2016 EP 2512136**

54 Título: **Enlosamiento en codificación y decodificación de vídeo**

30 Prioridad:

12.04.2007 US 923014 P
20.04.2007 US 925400 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
14.10.2016

73 Titular/es:

THOMSON LICENSING (100.0%)
1-5, rue Jeanne d'Arc
92130 Issy-les-Moulineaux, FR

72 Inventor/es:

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

74 Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

ES 2 586 406 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

Enlosamiento en codificación y decodificación de vídeo

Referencia cruzada a solicitudes relacionadas

5 Esta solicitud reivindica el beneficio de cada una de (1) Solicitud Provisional de EE.UU. N° de Serie 60/923.014, presentada el 12 de abril de 2007 y titulada "Multiview Information" (Expediente del Agente PU070078) y (2) Solicitud Provisional de EE.UU. N° de Serie 60/925.400, presentada el 20 de abril de 2007 y titulada "View Tiling in MVC Coding" (Expediente del Agente PU070103).

Campo técnico

Los presentes principios se refieren de manera general a codificación y/o decodificación de vídeo.

10 Antecedentes

Los fabricantes de pantallas de vídeo pueden usar una estructura de disposición o enlosamiento de diferentes vistas en un único cuadro. Las vistas se pueden extraer entonces de sus respectivas ubicaciones y reproducir.

El documento de patente DE19619598 y titulado "Verfahren zur Speicherung oder Obertragung von stereoskopischen Videosignalen" describe un proceso para el almacenamiento o la transmisión de señales de vídeo estereoscópicas.

15 El documento de patente EP1581003 describe un sistema de monitorización.

Compendio

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

Según otro aspecto general, se genera información que indica cómo 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.

25 Según otro aspecto 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 también incluye una representación codificada de las múltiples imágenes combinadas.

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

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

40 Los detalles de una o más implementaciones se exponen en los dibujos anexos y la descripción de más adelante. Incluso si se describe de una manera particular, debería quedar claro que las implementaciones se pueden configurar o incorporar de diversas maneras. Por ejemplo, una implementación se puede realizar como un método o incorporar como un aparato configurado para realizar un conjunto de operaciones o incorporar como un aparato que almacena instrucciones para realizar un conjunto de operaciones o incorporar en una señal. Otros aspectos y rasgos llegarán a ser evidentes a partir de la siguiente descripción detallada considerada en conjunto con los dibujos anexos y las reivindicaciones.

45 Breve descripción de los dibujos

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

La FIG. 2 es un diagrama que muestra un ejemplo de cuatro vistas volteadas y enlosadas en un único cuadro;

50 La FIG. 3 muestra un diagrama de bloques de un codificador de vídeo al cual se pueden aplicar los presentes principios, de acuerdo con una realización de los presentes principios;

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

Descripción detallada

- 5 Diversas implementaciones se dirigen a métodos y aparatos para enlosamiento de vista en codificación y decodificación de vídeo. Se apreciará, de esta manera, que los expertos en la técnica serán capaces de idear diversas disposiciones que, aunque no se describan o muestren explícitamente en la presente memoria, incorporan los presentes principios y se incluyen dentro de su espíritu y alcance.
- Todos los ejemplos y el lenguaje condicional expuesto en la presente memoria están destinados a propósitos pedagógicos para ayudar al lector en la comprensión de los presentes principios y los conceptos aportados por el(los) inventor(es) para fomentar la técnica y se tienen que interpretar como que son sin limitación a tales ejemplos y condiciones expuestos específicamente.
- 10 Además, todas las declaraciones en la presente memoria que exponen principios, aspectos y realizaciones de los presentes principios, así como ejemplos específicos de los mismos, se pretende que abarquen tanto equivalentes estructurales como funcionales de los mismos. Adicionalmente, se pretende que tales equivalentes incluyan tanto equivalentes conocidos actualmente así como equivalentes desarrollados en el futuro, es decir, cualquier elemento desarrollado que realice la misma función, con independencia de la estructura.
- 15 De esta manera, por ejemplo, se apreciará por los expertos en la técnica que los diagramas de bloques presentados en la presente memoria representan vistas conceptuales de circuitería ilustrativa que incorporan los presentes principios. Del mismo modo, se apreciará que cualquier gráfico de flujo, diagramas de flujo, diagramas de transición de estados, pseudocódigo y similares representan diversos procesos que se pueden representar sustancialmente en medios legibles por ordenador y de esta manera ejecutar por un ordenador o procesador, sea o no mostrado explícitamente tal ordenador o procesador.
- 20 Las funciones de los diversos elementos mostrados en las figuras se pueden proporcionar a través del uso de hardware dedicado así como de hardware capaz de ejecutar software en asociación con software adecuado. Cuando se proporciona por un procesador, las funciones se pueden proporcionar por un único procesador dedicado, por un único procesador compartido o por una pluralidad de procesadores individuales, algunos de los cuales pueden ser compartidos. Por otra parte, el uso explícito del término “procesador” o “controlador” no se debería interpretar para referirse exclusivamente a hardware capaz de ejecutar software y puede incluir implícitamente, sin limitación, hardware de procesador de señal digital (“DSP”), memoria de sólo lectura (“ROM”) para almacenar software, memoria de acceso aleatorio (“RAM”) y almacenamiento no volátil.
- 25 También se puede incluir otro hardware, convencional y/o personalizado. Del mismo modo, cualquier conmutador mostrado en las figuras es solamente conceptual. Su función se puede llevar a cabo a través de la operación de lógica de programa, a través de lógica dedicada, a través de la interacción del control del programa y de lógica dedicada o incluso manualmente, la técnica particular que es seleccionable por el implementador como se entiende más específicamente a partir del contexto.
- 30 En las reivindicaciones de la misma, cualquier elemento expresado como un medio para realizar una función especificada está destinado a abarcar cualquier forma de realización de esa función incluyendo, por ejemplo, a) una combinación de elementos de circuito que realizan esa función o b) software en cualquier forma, incluyendo, por lo tanto, microprogramas, microcódigo o similares, combinados con circuitería adecuada para ejecutar ese software para realizar la función. Los presentes principios como se definen por tales reivindicaciones residen en el hecho de que las funcionalidades proporcionadas por los diversos medios expuestos se combinan y se unen de la manera que exigen las reivindicaciones. De esta manera se considera que cualquier medio que pueda proporcionar esas funcionalidades es equivalente a los mostrados en la presente memoria.
- 35 La referencia en la especificación a “una realización” (o “una implementación”) de los presentes principios significa que un rasgo, estructura, característica y así sucesivamente particular descrito en conexión con la realización se incluye en al menos una realización de los presentes principios. De esta manera, las apariciones de la frase “en una realización” que aparecen en diversos lugares a lo largo de la especificación no están refiriéndose necesariamente todas a la misma realización.
- 40 Se tiene que apreciar que el uso de los términos “y/o” y “al menos uno de”, por ejemplo, en los casos de “A y/o B” y “al menos uno de A y B”, se pretende que abarque la selección de la primera opción enumerada (A) solamente o la selección de la segunda opción enumerada (B) solamente o la selección de ambas opciones (A y B). Como ejemplo adicional, en los casos de “A, B y/o C” y “al menos uno de A, B y C”, tal expresión se pretende que abarque la selección de la primera opción enumerada (A) solamente o la selección de la segunda opción enumerada (B) solamente o la selección de la tercera opción enumerada (C) solamente o la selección de la primera y la segunda de las opciones enumeradas (A y B) solamente o la selección de la primera y tercera opciones enumeradas (A y C) solamente o la selección de la segunda y tercera opciones enumeradas (B y C) solamente o la selección de todas de las tres opciones (A y B y C). Esto se puede extender, como es fácilmente evidente para un experto ordinario en ésta y técnicas relacionadas, para tantos elementos enumerados.
- 45 Por otra parte, se tiene que apreciar que mientras que una o más realizaciones de los presentes principios se describen en la presente memoria con respecto al estándar de AVC del MPEG-4, los presentes principios no están

limitados únicamente a este estándar y, de esta manera, se pueden utilizar con respecto a otros estándares, recomendaciones y extensiones de los mismos, particularmente a estándares de codificación de vídeo, recomendaciones y extensiones de los mismos, incluyendo extensiones del estándar de AVC del MPEG-4, mientras se mantiene el espíritu de los presentes principios.

5 Además, se tiene que apreciar que mientras que una o más de otras realizaciones de los presentes principios se describen en la presente memoria con respecto a la extensión de codificación de vídeo multivista del estándar de AVC del MPEG-4, los presentes principios no van a estar limitados a únicamente esta extensión y/o este estándar y, de esta manera, se pueden utilizar con respecto a otros estándares de codificación de vídeo, recomendaciones y extensiones de los mismos con relación a codificación de vídeo multivista, mientras que se mantiene el espíritu de los presentes principios. La codificación de vídeo multivista (MVC) es la estructura de compresión para la codificación de secuencias multivista. Una secuencia de codificación de vídeo multivista (MVC) es un conjunto de dos o más secuencias de vídeo que capturan la misma escena desde un punto de vista diferente.

10 También, se tienen que apreciar que mientras que una o más de otras realizaciones de los presentes principios se describen en la presente memoria que usan información de profundidad con respecto al contenido de vídeo, los presentes principios no se limitan a tales realizaciones y, de esta manera, se pueden implementar otras realizaciones que no usan información de profundidad, mientras se mantiene el espíritu de los presentes principios.

15 Adicionalmente, como se usa en la presente memoria, "sintaxis de alto nivel" se refiere a sintaxis presente en el flujo de bits que reside jerárquicamente por encima de la capa de macrobloque. Por ejemplo, sintaxis de alto nivel, como se usa en la presente memoria, puede referirse a, pero no está limitada a, sintaxis a nivel de cabecera de segmento, nivel de Información de Mejora Suplementaria (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).

20 En la implementación actual de codificación multivideo (MVC) en base al estándar de Codificación de Vídeo Avanzado (AVC) Parte 10 del Grupo de Expertos en Imágenes en Movimiento 4 (MPEG-4) de la Organización internacional de Estandarización/Comisión Electrotécnica Internacional (ISO/IEC)/Recomendación H.264 de la Unión Internacional de Telecomunicaciones, Sector de Telecomunicaciones (UIT-T) (en adelante, el "Estándar de AVC del MPEG-4"), el software de referencia logra predicción multivista codificando cada vista con un único codificador y teniendo en consideración las referencias de vistas cruzadas. Cada vista se codifica como un flujo de bits separado por el codificador en su resolución original y más tarde se combinan todos los flujos de bits para formar un único flujo de bits que entonces se decodifica. Cada vista produce una salida decodificada YUV separada.

25 Otro planteamiento para predicción multivista implica agrupar un conjunto de vistas en pseudovistas. En un ejemplo de este planteamiento, podemos enlazar las imágenes a partir de cada N vistas de entre el total de M vistas (muestreadas al mismo tiempo) en un cuadro más grande o un supercuadro con un posible muestreo descendente u otras operaciones. Volviendo a la FIG. 1, un ejemplo de las cuatro vistas enlazadas en un único cuadro se indica de manera general por el número de referencia 100. Todas de las cuatro vistas están en su orientación normal.

30 Volviendo a la FIG. 2, un ejemplo de cuatro vistas volteadas y enlazadas en un único cuadro se indica de manera general por el número de referencia 200. La vista superior izquierda está en su orientación normal. La vista superior derecha está volteada horizontalmente. La vista inferior izquierda está volteada verticalmente. La vista inferior derecha está volteada tanto horizontal como verticalmente. De esta manera, si hay cuatro vistas, entonces una imagen de cada vista se dispone en un supercuadro como una losa. Esto provoca una única secuencia de entrada no codificada con una resolución grande.

35 Alternativamente, se puede muestrear descendentemente la imagen para producir una resolución menor. De esta manera, creamos múltiples secuencias que incluyen cada una diferentes vistas que se enlazan juntas. Cada una de tales secuencias forma entonces una pseudovista, donde cada pseudovista incluye N vistas enlazadas diferentes. La FIG. 1 muestra una pseudovista y la FIG. 2 muestra otra pseudovista. Estas pseudovistas entonces se pueden codificar usando estándares de codificación de vídeo existentes tales como el Estándar MPEG-2 y Estándar de AVC del MPEG-4 de la ISO/IEC.

40 Aún otro planteamiento para predicción multivista simplemente implica codificar las diferentes vistas de manera independiente usando un nuevo estándar y, después de decodificar, enlazar las vistas como se requiera por el reproductor.

45 Además, en otro planteamiento, las vistas también se pueden enlazar de una forma en modo de píxeles. Por ejemplo, en una supervista que se compone de cuatro vistas, un píxel (x, y) puede ser de la vista 0, mientras que un píxel (x+1, y) puede ser de la vista 1, un píxel (x, y+1) puede ser de la vista 2 y un píxel (x+1, y+1) puede ser de la vista 3.

50 Muchos fabricantes de pantallas usan tal estructura de disposición o enlazar de diferentes vistas en un único cuadro y luego extraer las vistas desde sus respectivas ubicaciones y representarlas. En tales casos, no hay ninguna forma estándar para determinar si el flujo de bits tiene tal propiedad. De esta manera, si un sistema usa el

método de enlosamiento de imágenes de diferentes vistas en un cuadro grande, entonces el método de extracción de las diferentes vistas es propietario.

No obstante, no hay forma estándar de determinar si el flujo de bits tiene tal propiedad. Proponemos una sintaxis de alto nivel a fin de facilitar al presentador o reproductor extraer tal información a fin de ayudar en la visualización u otro procesamiento posterior. También es posible que las subimágenes tengan diferentes resoluciones y se puede necesitar algún muestreo ascendente para representar eventualmente la vista. El usuario puede querer tener el método de muestreo ascendente indicado en la sintaxis de alto nivel también. Adicionalmente, también se pueden transmitir parámetros para cambiar el foco de profundidad.

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

Por otra parte, en una realización, proponemos un método de señalización de cómo disponer vistas antes de que se envíen al codificador y/o decodificador de vídeo multivista. Ventajosamente, la realización puede conducir a una implementación simplificada de la codificación multivista y puede beneficiar a la eficacia de la codificación. Ciertas vistas se pueden poner juntas y formar una pseudovista o supervista y luego la supervista enlosada se trata como una vista normal por un codificador y/o decodificador de vídeo multivista común, por ejemplo, según la implementación de codificación de vídeo multivista basada en el Estándar de AVC del MPEG-4. Se propone una nueva bandera en la extensión del Conjunto de Parámetros de Secuencia (SPS) de codificación de vídeo multivista para señalar el uso de la técnica de pseudovistas. La realización está destinada a la visualización fácil y cómoda de flujos de vídeo multivista en monitores 3D que pueden usar tal estructura.

Codificación/decodificación usando un estándar/recomendación de codificación/decodificación de vídeo de única vista

En la implementación actual de codificación multivideo (MVC) en base al estándar de Codificación de Vídeo Avanzada (AVC) Parte 10 del Grupo de Expertos en Imágenes en Movimiento 4 (MPEG-4) de la Organización internacional de Estandarización/Comisión Electrotécnica Internacional (ISO/IEC)/Recomendación H.264 de la Unión Internacional de Telecomunicaciones, Sector de Telecomunicaciones (UIT-T) (en adelante, el "Estándar de AVC del MPEG-4"), el software de referencia logra predicción multivista codificando cada vista con un único codificador y teniendo en consideración las referencias de vistas cruzadas. Cada vista se codifica como un flujo de bits separado por el codificador en su resolución original y más tarde se combinan todos los flujos de bits para formar un único flujo de bits que entonces se decodifica. Cada vista produce una salida decodificada YUV separada.

Otro planteamiento para predicción multivista implica el enlosamiento de las imágenes de cada vista (muestreada al mismo tiempo) en un cuadro más grande o un supercuadro con una posible operación de muestreo descendente. Volviendo a la FIG. 1, un ejemplo de cuatro vistas enlosadas en un único cuadro se indica de manera general por el número de referencia 100. Volviendo a la FIG. 2, un ejemplo de cuatro vistas volteadas y enlosadas en único cuadro se indica de manera general por el número de referencia 200. De esta manera, si hay cuatro vistas, entonces, una imagen de cada vista se dispone en un supercuadro como una losa. Esto implica una única secuencia de entrada no codificada con una resolución grande. Esta señal entonces se puede codificar usando estándares de codificación de vídeo existentes tales como el Estándar MPEG-2 y el Estándar de AVC del MPEG-4 de la ISO/IEC.

Aún otro planteamiento para predicción multivista simplemente implica codificar las diferentes vistas independientemente usando un nuevo estándar y, después de la decodificación, enlosar las vistas como se requiera por el reproductor.

Muchos fabricantes de pantallas usan tal estructura de disposición o enlosamiento de diferentes vistas en un único cuadro y luego extraer las vistas desde sus respectivas ubicaciones y representarlas. En tales casos, no hay ninguna forma estándar para determinar si el flujo de bits tiene tal propiedad. De esta manera, si un sistema usa el método de enlosamiento de imágenes de diferentes vistas en un cuadro grande, entonces el método de extracción de las diferentes vistas es propietario.

Volviendo a la FIG. 3, un codificador de vídeo capaz de realizar codificación de vídeo de acuerdo con el estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 300.

El codificador de vídeo 300 incluye un almacenador temporal de ordenación de cuadro 310 que tiene una salida en comunicación de señal con una entrada no inversora de un combinador 385. Una salida del combinador 385 está conectada en comunicación de señal con una primera entrada de un transformador y cuantificador 325. Una salida del transformador y cuantificador 325 está conectada en comunicación de señal con una primera entrada de un codificador de entropía 345 y una primera entrada de un transformador inverso y cuantificador inverso 350. Una salida del codificador de entropía 345 está conectada en comunicación de señal con una primera entrada no

inversora de un combinador 390. Una salida del combinador 390 está conectada en comunicación de señal con una primera entrada de un almacenador temporal de salida 335.

5 Una primera salida de un controlador del codificador 305 está conectada en comunicación de señal con una segunda entrada del almacenador temporal de ordenación de cuadro 310, una segunda entrada del transformador inverso y cuantificador inverso 350, una entrada de un módulo de decisión de tipo imagen 315, una entrada de un módulo de decisión de tipo macrobloque (tipo MB) 320, una segunda entrada de un módulo de intrapredicción 360, una segunda entrada de un filtro de desbloqueo 365, una primera entrada de un compensador de movimiento 370, una primera entrada de un estimador de movimiento 375 y una segunda entrada de un almacenador temporal de imagen de referencia 380.

10 Una segunda salida del controlador del codificador 305 está conectada en comunicación de señal con una primera entrada de un insertador de Información de Mejora Suplementaria (SEI) 330, una segunda entrada del transformador y cuantificador 325, una segunda entrada del codificador de entropía 345, una segunda entrada del almacenador temporal de salida 335 y una entrada del insertador de Conjunto de Parámetros de Secuencia (SPS) y Conjunto de Parámetros de Imagen (PPS) 340.

15 Una primera salida del módulo de decisión de tipo imagen 315 está conectada en comunicación de señal con una tercera entrada de un almacenador temporal de ordenación de cuadro 310. Una segunda salida del módulo de decisión de tipo imagen 315 está conectada en comunicación de señal con una segunda entrada de un módulo de decisión de tipo macrobloque 320.

20 Una salida del insertador de Conjunto de Parámetros de Secuencia (SPS) y Conjunto de Parámetros de Imagen (PPS) 340 está conectada en comunicación de señal con una tercera entrada no inversora del combinador 390. Una salida del insertador SEI 330 está conectada en comunicación de señal con una segunda entrada no inversora del combinador 390.

25 Una salida del cuantificador inverso y transformador inverso 350 está conectada en comunicación de señal con una primera entrada no inversora de un combinador 319. Una salida del combinador 319 está conectada en comunicación de señal con una primera entrada del módulo de intrapredicción 360 y una primera entrada del filtro de desbloqueo 365. Una salida del filtro de desbloqueo 365 está conectada en comunicación de señal con una primera entrada de un almacenador temporal de imagen de referencia 380. Una salida del almacenador temporal de imagen de referencia 380 está conectada en comunicación de señal con una segunda entrada del estimador de movimiento 375 y con una primera entrada de un compensador de movimiento 370. Una primera salida del estimador de movimiento 375 está conectada en comunicación de señal con una segunda entrada del compensador de movimiento 370. Una segunda salida del estimador de movimiento 375 está conectada en comunicación de señal con una tercera entrada del codificador de entropía 345.

35 Una salida del compensador de movimiento 370 está conectada en comunicación de señal con una primera entrada de un conmutador 397. Una salida del módulo de intrapredicción 360 está conectada en comunicación de señal con una segunda entrada del conmutador 397. Una salida del módulo de decisión de tipo macrobloque 320 está conectada en comunicación de señal con una tercera entrada del conmutador 397 a fin de 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 entrada de control, es decir, la tercera entrada) va a ser proporcionada o no por el compensador de movimiento 370 o el módulo de intrapredicción 360. La salida del conmutador 397 está conectada en comunicación de señal con una segunda entrada no inversora del combinador 319 y con una entrada inversora del combinador 385.

45 Las entradas del almacenador temporal de ordenación de cuadro 310 y el controlador del codificador 105 están disponibles como entrada del codificador 300, para recibir una imagen de entrada 301. Por otra parte, una entrada del insertador de información de Mejora Suplementaria (SEI) 330 está disponible como una entrada del codificador 300, para recibir metadatos. Una salida del almacenador temporal de salida 335 está disponible como una salida del codificador 300, para sacar un flujo de bits.

Volviendo a la FIG. 4, un decodificador de vídeo capaz de realizar decodificación de vídeo de acuerdo con el estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 400.

50 El decodificador de vídeo 400 incluye un almacenador temporal de entrada 410 que tiene una salida conectada en comunicación de señal con una primera entrada del decodificador de entropía 445. Una primera salida del decodificador de entropía 445 está conectada en comunicación de señal con una primera entrada de un transformador inverso y cuantificador inverso 450. Una salida del transformador de inverso y cuantificador inverso 450 está conectada en comunicación de señal con una segunda entrada no inversora de un combinador 425. Una salida del combinador 425 está conectada en comunicación de señal con una segunda entrada de un filtro de desbloqueo 465 y una primera entrada de un módulo de intrapredicción 460. Una segunda salida del filtro de desbloqueo 465 está conectada en comunicación de señal con una primera entrada de un almacenador temporal de imágenes de referencia 480. Una salida del almacenador temporal de imágenes de referencia 480 está conectada en comunicación de señal con una segunda entrada de un compensador de movimiento 470.

Una segunda salida del decodificador de entropía 445 está conectada en comunicación de señal con una tercera entrada del compensador de movimiento 470 y una primera entrada del filtro de desbloqueo 465. Una tercera salida del decodificador de entropía 445 está conectada en comunicación de señal con una entrada de un controlador de decodificador 405. Una primera salida del controlador del decodificador 405 está conectada en comunicación de
 5 señal con una segunda entrada del decodificador de entropía 445. Una segunda salida del controlador del decodificador 405 está conectada en comunicación de señal con una segunda de entrada del transformador inverso y cuantificador inverso 450. Una tercera salida del controlador del decodificador 405 está conectada en comunicación de señal con una tercera entrada del filtro de desbloqueo 465. Una cuarta salida del controlador del decodificador 405 está conectada en comunicación de señal con una segunda entrada del módulo de intrapredicción
 10 460, con una primera entrada del compensador de movimiento 470 y con una segunda entrada del almacenador temporal de imagen de referencia 480.

Una salida del compensador de movimiento 470 está conectada en comunicación de señal con una primera entrada de un conmutador 497. Una salida del módulo de intrapredicción 460 está conectada en comunicación de señal con una segunda entrada del conmutador 497. Una salida del conmutador 497 está conectada en comunicación de señal con una primera entrada no inversora del combinador 425.
 15

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

Volviendo a la FIG. 5, un método ejemplar para codificación de imágenes para una pluralidad de vistas usando el estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 500.
 20

El método 500 incluye un bloque de inicio 502 que pasa el control a un bloque de función 504. El bloque de función 504 dispone cada vista en un instante de tiempo particular como una subimagen en formato de losa y pasa el control a un bloque de función 506. El bloque de función 506 fija un elemento de sintaxis num_coded_views_minus1 y pasa el control a un bloque de función 508. El bloque de función 508 fija los elementos de sintaxis
 25 org_pic_width_in_mbs_minus1 y org_pic_height_in_mbs_minus1 y pasa el control a un bloque de función 510. El bloque de función 510 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 menor o no que el número de vistas. Si es así, entonces se pasa el control a un bloque de función 514. De otro modo, el control se pasa a un bloque de función 524.

El bloque de función 514 fija un elemento de sintaxis view_id[i] y pasa el control a un bloque de función 516. El bloque de función 516 fija un elemento de sintaxis num_parts [view_id[i]] y pasa el control a un bloque de función 518. El bloque de función 518 fija de una variable j igual a cero y pasa el control a un bloque de decisión 520. El bloque de decisión 520 determina si el valor actual de la variable j es menor o no que el valor actual del elemento de sintaxis num_parts[view_id[i]]. Si es así, entonces el control se pasa a un bloque de función 522. De otro modo, el control se pasa a un bloque de función 528.
 30

El bloque de función 522 fija los siguientes elementos de sintaxis, incrementa la variable j y luego 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].
 35

El bloque de función 528 fija un elemento de sintaxis upsample_view_flag[view_id[i]] y pasa el control a un bloque de decisión 530. El bloque de decisión 530 determina si el valor actual del elemento de sintaxis upsample_view_flag[view_id[i]] es igual o no a uno. Si es así, entonces se pasa el control a un bloque de función 532. De otro modo, el control se pasa a un bloque de decisión 534.
 40

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

El bloque de decisión 534 determina si el valor actual del elemento de sintaxis upsample_filter[view_id [i]] es igual o no a tres. Si es así, entonces se pasa el control a un bloque de función 536. De otro modo, se pasa el control a un bloque de función 540.
 45

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

El bloque de función 538 fija los coeficientes de filtro para cada componente YUV y pasa el control al bloque de función 540.
 50

El bloque de función 540 aumenta la variable i y devuelve el control al bloque de decisión 512.

El bloque de función 524 escribe estos elementos de sintaxis para al menos uno del Conjunto de Parámetros de Secuencia (SPS), Conjunto de Parámetros de Imagen (PPS), mensaje de Información de Mejora Suplementaria (SEI), cabecera de unidad de Capa de Abstracción de Red (NAL) y cabecera de segmento y pasa el control a un
 55

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

Volviendo a la FIG. 6, un método ejemplar para decodificación de imágenes para una pluralidad de vistas usando el Estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 600.

5 El método 600 incluye un bloque de inicio 602 que pasa el control a un bloque de función 604. El bloque de función 604 analiza sintácticamente los siguientes elementos de sintaxis de al menos uno del Conjunto de Parámetros de Secuencia (SPS), el Conjunto de Parámetros de Imagen (PPS), mensaje de Información de Mejora Suplementaria (SEI), cabecera de unidad de Capa de Abstracción de Red (NAL) y la cabecera de segmento y pasa el control a un bloque de función 606. El bloque de función 606 analiza sintácticamente un elemento de sintaxis num_coded_views_minus1 y pasa el control a un bloque de función 608. El bloque de función 608 analiza sintácticamente los elementos de sintaxis org_pic_width_in_mbs_minus1 y org_pic_height_in_mbs_minus1 y pasa el control a un bloque de función 610. El bloque de función 610 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 menor o no que el número de vistas. Si es así, entonces se pasa el control a un bloque de función 614. De otro modo, se pasa el control al bloque de función 624.

El bloque de función 614 analiza sintácticamente un elemento de sintaxis view_id[i] y pasa el control a un bloque de función 616. El bloque de función 616 analiza sintácticamente un elemento de sintaxis num_parts_minus1[view_id[i]] y pasa el control a un bloque de función 618. El bloque de función 618 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 menor o no que el valor actual del elemento de sintaxis num_parts[view_id[i]]. Si es así, entonces se pasa el control a un bloque de función 622. De otro modo, se pasa el control a un bloque de función 628.

El bloque de función 622 analiza sintácticamente los siguientes elementos de sintaxis, aumenta la variable j y luego 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].

El bloque de función 628 analiza sintácticamente un elemento de sintaxis upsample_view_flag[view_id [i]] y pasa el control a un bloque de decisión 630. El bloque de decisión 630 determina si el valor actual del elemento de sintaxis upsample_view_flag[view_id [i]] es igual o no a uno. Si es así, entonces se pasa el control a un bloque de función 632. De otro modo, se pasa el control a un bloque de decisión 634.

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

El bloque de decisión 634 determina si el valor actual del elemento de sintaxis upsample_filter[view_id[i]] es igual o no a tres. Si es así, entonces se pasa el control a un bloque de función 636. De otro modo, se pasa el control a un bloque de función 640.

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

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

El bloque de función 640 aumenta la variable i y devuelve el control al bloque de decisión 612.

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

Volviendo a la FIG. 7, un método ejemplar para codificación de imágenes para una pluralidad de vistas y profundidades usando el Estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 700.

45 El método 700 incluye un bloque de inicio 702 que pasa el control a un bloque de función 704. El bloque de función 704 dispone cada vista y la profundidad correspondiente en un instante de tiempo particular como una subimagen en formato de losa y pasa el control a un bloque de función 706. El bloque de función 706 fija un elemento de sintaxis num_coded_views_minus1 y pasa el control a un bloque de función 708. El bloque de función 708 fija los elementos de sintaxis org_pic_width_in_mbs_minus1 y org_pic_height_in_mbs_minus1 y pasa el control a un bloque de función 710. El bloque de función 710 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 menor o no que el número de vistas. Si es así, entonces se pasa el control a un bloque de función 714. De otro modo, se pasa el control a un bloque de función 724.

- 5 El bloque de función 714 fija un elemento de sintaxis `view_id[i]` y pasa el control a un bloque de función 716. El bloque de función 716 fija un elemento de sintaxis `num_parts[view_id[i]]` y pasa el control a un bloque de función 718. El bloque de función 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 menor o no que el valor actual del elemento de sintaxis `num_parts[view_id[i]]`. Si es así, entonces se pasa el control a un bloque de función 722. De otro modo, se pasa el control a un bloque de función 728.
- 10 El bloque de función 722 fija los siguientes elementos de sintaxis, aumenta la variable `j` y entonces 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]`.
- 15 El bloque de función 728 fija un elemento de sintaxis `upsample_view_flag[view_id[i]]` y pasa el control a un bloque de decisión 730. El bloque de decisión 730 determina si el valor actual del elemento de sintaxis `upsample_view_flag[view_id[i]]` es igual o no a uno. Si es así, entonces se pasa el control a un bloque de función 732. De otro modo, se pasa el control a un bloque de decisión 734.
- El bloque de decisión 734 determina si el valor actual del elemento de sintaxis `upsample_filter[view_id[i]]` es igual o no a tres. Si es así, entonces se pasa el control a un bloque de función 736. De otro modo, se pasa el control a un bloque de función 740.
- 20 El bloque de función 736 fija los siguientes elementos de sintaxis y pasa el control a un bloque de función 738: `vert_dim[view_id[i]]`; `hor_dim[view_id[i]]`; y `quantizer[view_id [i]]`.
- El bloque de función 738 fija los coeficientes de filtro para cada componente YUV y pasa el control al bloque de función 740.
- El bloque de función 740 aumenta la variable `i` y devuelve el control al bloque de decisión 712.
- 25 El bloque de función 724 escribe estos elementos de sintaxis a al menos uno de uno del Conjunto de Parámetros de Secuencia (SPS), el Conjunto de Parámetros de Imagen (PPS), mensaje de Información de Mejora Suplementaria (SEI), cabecera de unidad de Capa de Abstracción de Red (NAL) y la cabecera de segmento y pasa el control a un bloque de función 726. El bloque de función 726 codifica cada imagen usando el Estándar de AVC del MPEG-4 u otro códec de única vista y pasa el control a un bloque final 799.
- 30 Volviendo a la FIG. 8, un método ejemplar para decodificación de imágenes para una pluralidad de vistas y profundidades usando el Estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 800.
- El método 800 incluye un bloque de inicio 802 que pasa el control a un bloque de función 804. El bloque de función 804 analiza sintácticamente los siguientes elementos de sintaxis de al menos uno del Conjunto de Parámetros de Secuencia (SPS), el Conjunto de Parámetros de Imagen (PPS), mensaje de Información de Mejora Suplementaria (SEI), cabecera de unidad de Capa de Abstracción de Red (NAL) y cabecera de segmento y pasa el control a un bloque de función 806. El bloque de función 806 analiza sintácticamente un elemento de sintaxis `num_coded_views_minus1` y pasa el control a un bloque de función 808. El bloque de función 808 analiza sintácticamente los elementos de sintaxis `org_pic_width_in_mbs_minus1` y `org_pic_height_in_mbs_minus1` y pasa el control a un bloque de función 810. El bloque de función 810 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 menor o no que el número de vistas. Si es así, entonces se pasa el control a un bloque de función 814. De otro modo, se pasa el control a un bloque de función 824.
- 35 El bloque de función 814 analiza sintácticamente un elemento de sintaxis `view_id[i]` y pasa el control a un bloque de función 816. El bloque de función 816 analiza sintácticamente un elemento de sintaxis `num_parts_minus1[view_id[i]]` y pasa el control a un bloque de función 818. El bloque de función 818 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 menor o no que el valor actual del elemento de sintaxis `num_parts[view_id[i]]`. Si es así, entonces se pasa el control a un bloque de función 822. De otro modo, se pasa el control a un bloque de función 828.
- 40 El bloque de función 822 analiza sintácticamente los siguientes elementos de sintaxis, aumenta la variable `j` y entonces 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]`.
- 45 El bloque de función 828 analiza sintácticamente un elemento de sintaxis `upsample_view_flag[view_id[i]]` y pasa el control a un bloque de decisión 830. El bloque de decisión 830 determina si el valor actual del elemento de sintaxis
- 50
- 55

upsample_view_flag[view_id[i]] es igual o no a uno. Si es así, entonces se pasa el control a un bloque de función 832. De otro modo, se pasa el control a un bloque de decisión 834.

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

- 5 El bloque de decisión 834 determina si el valor actual del elemento de sintaxis upsample_filter[view_id[i]] es igual o no a tres. Si es así, entonces se pasa el control a un bloque de función 836. De otro modo, se pasa el control a un bloque de función 840.

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

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

El bloque de función 840 aumenta la variable *i* y devuelve el control al bloque de decisión 812.

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

- 20 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 se proporciona la profundidad 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 una losa 1000. La parte superior derecha de la losa 1000 es una señal de profundidad que tiene valores de profundidad que corresponden a la imagen en la parte superior izquierda de la losa 1000. El parte inferior derecha de la losa 1000 es una señal de profundidad que tiene valores de profundidad que corresponden a la imagen en la parte inferior izquierda de la losa 1000.

- 25 Volviendo a la FIG. 11, un ejemplo de 5 vistas enlosadas en un único cuadro se indica de manera general con el número de referencia 100. 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 de la losa 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.

- 30 Codificación/decodificación usando un estándar/recomendación de codificación/decodificación de vídeo multivista

- Volviendo a la FIG. 12, un codificador de Codificación de Vídeo Multivista (MVC) ejemplar se indica de manera general por el número de referencia 1200. El codificador 1200 incluye un combinador 1205 que tiene una salida conectada en comunicación de señal con una entrada de un transformador 1210. Una salida del transformador 1210 está conectada en comunicación de señal con una entrada del cuantificador 1215. Una salida del cuantificador 1215 está conectada en comunicación de señal con una entrada de un codificador de entropía 1220 y una entrada de un cuantificador inverso 1225. Una salida del cuantificador inverso 1225 está conectada en comunicación de señal con una entrada de un transformador inverso 1230. Una salida del transformador inverso 1230 está conectada en comunicación de señal con una primera entrada no inversora de un combinador 1235. Una salida del combinador 1235 está conectada en comunicación de señal con una entrada de un intrapredicador 1245 y una entrada de un filtro de desbloqueo 1250. Una salida del filtro de desbloqueo 1250 está conectada en comunicación de señal con una entrada de un almacén de imagen de referencia 1255 (para la vista *i*). Una salida del almacén de imagen de referencia 1255 está conectada en comunicación de señal con una primera entrada de un compensador de movimiento 1275 y una primera entrada de un estimador de movimiento 1280. Una salida del estimador de movimiento 1280 está conectada en comunicación de señal con una segunda entrada del compensador de movimiento 1275.

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

- 50 Una salida del decodificador de entropía 1220 está disponible como una salida del codificador 1200. Una entrada no inversora del combinador 1205 está disponible como una entrada del codificador 1200 y está conectada en comunicación de señal con una segunda entrada del estimador de disparidad 1270 y una segunda entrada del estimador de movimiento 1280. Una salida de un conmutador 1285 está conectada en comunicación de señal con una segunda entrada no inversora del combinador 1235 y con una entrada inversora del combinador 1205. El conmutador 1285 incluye una primera entrada conectada en comunicación de señal con una salida del compensador

- 55

de movimiento 1275, una segunda entrada conectada en comunicación de señal con una salida del compensador de disparidad 1265 y una tercera entrada conectada en comunicación de señal con una salida del intrapredicador 1245.

Un módulo de decisión de modo 1240 tiene una salida conectada al conmutador 1285 para control cuya entrada se selecciona por el conmutador 1285.

- 5 Volviendo a la FIG. 13, un decodificador de Codificación de Vídeo Multivista (MVC) ejemplar se indica de manera general por el número de referencia 1300. El decodificador 1300 incluye un decodificador de entropía 1305 que tiene una salida conectada en comunicación de señal con una entrada de un cuantificador inverso 1310. Una salida del cuantificador inverso está conectada en comunicación de señal con una entrada de un transformador inverso 1315. Una salida del transformador inverso 1315 está conectada en comunicación de señal con una primera entrada no inversora de un combinador 1320. Una salida del combinador 1320 está conectada en comunicación de señal con una entrada de un filtro de desbloqueo 1325 y una entrada de un intrapredicador 1330. Una salida del filtro de desbloqueo 1325 está conectada en comunicación de señal con una entrada de un almacén de imagen de referencia 1340 (para la vista i). Una salida del almacén de imagen de referencia 1340 está conectada en comunicación de señal con una primera entrada de un compensador de movimiento 1335.
- 10
- 15 Una salida de un almacén de imagen de referencia 1345 (para otras vistas) está conectada en comunicación de señal con una primera entrada de un compensador de disparidad 1350.

- Una entrada del codificador de entropía 1305 está disponible como una entrada al decodificador 1300, para recibir un flujo de bits residuo. Por otra parte, una entrada de un módulo de modo 1360 también está disponible como una entrada al decodificador 1300, para recibir la sintaxis de control para controlar qué entrada se selecciona por el conmutador 1355. Además, una segunda entrada del compensador de movimiento 1335 está disponible como una entrada del decodificador 1300, para recibir vectores de movimiento. También, una segunda entrada del compensador de disparidad 1350 está disponible como una entrada al decodificador 1300, para recibir vectores de disparidad.
- 20

- Una salida de un conmutador 1355 está conectada en comunicación de señal con una segunda entrada no inversora del combinador 1320. Una primera entrada del conmutador 1355 está conectada en comunicación de señal con una salida del compensador de disparidad 1350. Una segunda entrada del conmutador 1355 está conectada en comunicación de señal con una salida del compensador de movimiento 1335. Una tercera entrada del conmutador 1355 está conectada en comunicación de señal con una salida del intrapredicador 1330. Una salida del módulo de modo 1360 está conectada en comunicación de señal con el conmutador 1355 para control cuya entrada se selecciona por el conmutador 1355. Una salida del filtro de desbloqueo 1325 está disponible como una salida del decodificador 1300.
- 25
- 30

Volviendo a la FIG. 14, un método ejemplar para procesar imágenes para una pluralidad de vistas en la preparación para la codificación de las imágenes usando la extensión de codificación de vídeo multivista (MVC) del Estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 1400.

- 35 El método 1400 incluye un bloque de inicio 1405 que pasa el control a un bloque de función 1410. El bloque de función 1410 dispone cada N vistas, entre un total de M vistas, en un instante de tiempo particular como una superimagen en formato de losa y pasa el control a un bloque de función 1415. El bloque de función 1415 fija un elemento de sintaxis `num_coded_views_minus1` y pasa el control a un bloque de función 1420. El bloque de función 1420 fija un elemento de sintaxis `view_id[i]` para todas las (`num_coded_views_minus1 + 1`) vistas y pasa el control a un bloque de función 1425. El bloque de función 1425 fija la información de dependencias de referencia entre vistas para imágenes de anclaje y pasa el control a un bloque de función 1430. El bloque de función 1430 fija la información de dependencias de referencia entre vistas para imágenes no de anclaje y pasa el control a un bloque de función 1435. El bloque de función 1435 fija un elemento de sintaxis `pseudo_view_present_flag` y pasa el control a un bloque de decisión 1440. El bloque de decisión 1440 determina si el valor actual del elemento de sintaxis `pseudo_view_present_flag` es igual o no a verdadero. Si es así, entonces se pasa el control a un bloque de función 1445. De otro modo, se pasa el control a un bloque final 1499.
- 40
- 45

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

- 50 Volviendo a la FIG. 15, un método ejemplar para codificar imágenes para una pluralidad de vistas usando la extensión de codificación de vídeo multivista (MVC) del Estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 1500.

- El método 1500 incluye un bloque de inicio 1502 que tiene un parámetro de entrada `pseudo_view_id` y pasa el control a un bloque de función 1504. El bloque de función 1504 fija un elemento de sintaxis `num_sub_views_minus1` y pasa el control a un bloque de función 1506. El bloque de función 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 menor o no que el número de `sub_views`. Si es así, entonces se pasa el control a un bloque de función 1510. De otro modo, se pasa el control a un bloque de función 1520.
- 55

5 El bloque de función 1510 fija un elemento de sintaxis `sub_view_id[i]` y pasa el control a un bloque de función 1512. El bloque de función 1512 fija un elemento de sintaxis `num_parts_minus1[sub_view_id[i]]` y pasa el control a un bloque de función 1514. El bloque de función 1514 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 menor o no que el elemento de sintaxis `num_parts_minus1[sub_view_id[i]]`. Si es así, entonces se pasa el control a un bloque de función 1518. De otro modo, se pasa el control a un bloque de decisión 1522.

10 El bloque de función 1518 fija los siguientes elementos de sintaxis, aumenta 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]`.

El bloque de función 1520 codifica la imagen actual de la vista actual usando codificación de vídeo multivista (MVC) y pasa el control a un bloque final 1599.

El bloque de decisión 1522 determina si un elemento de sintaxis `tiling_mode` es igual o no a cero. Si es así, entonces se pasa el control a un bloque de función 1524. De otro modo, se pasa el control a un bloque de función 1538.

15 El bloque de función 1524 fija un elemento de sintaxis `flip_dir[sub_view_id[i]]` y un elemento de sintaxis `upsample_view_flag[sub_view_id[i]]` y pasa el control a un bloque de decisión 1526. El bloque de decisión 1526 determina si el valor actual del elemento de sintaxis `upsample_view_flag[sub_view_id[i]]` es igual o no a uno. Si es así, entonces se pasa el control a un bloque de función 1528. De otro modo, se pasa el control a un bloque de decisión 1530.

20 El bloque de función 1528 fija un elemento de sintaxis `upsample_filter[sub_view_id[i]]` y pasa el control al bloque de decisión 1530. El bloque de decisión 1530 determina si un valor del elemento de sintaxis `upsample_filter[sub_view_id[i]]` es igual o no a tres. Si es así, se pasa el control a un bloque de función 1532. De otro modo, se pasa el control a un bloque de función 1536.

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

El bloque de función 1536 aumenta la variable `i` y devuelve el control al bloque de decisión 1508.

30 El bloque de función 1538 fija un elemento de sintaxis `pixel_dist_x[sub_view_id[i]]` y el elemento de sintaxis `flip_dist_y[sub_view_id[i]]` y pasa el control a un bloque de función 1540. El bloque de función 1540 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 menor o no que el valor actual del elemento de sintaxis `num_parts[sub_view_id[i]]`. Si es así, entonces se pasa el control a un bloque de función 1544. De otro modo, se pasa el control al bloque de función 1536.

35 El bloque de función 1544 fija un elemento de sintaxis `num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]]` y pasa el control a un bloque de función 1546. El bloque de función 1546 fija los coeficientes para todos los filtros de enlosamiento de píxeles y pasa el control al bloque de función 1536.

Volviendo a la FIG. 16, un método ejemplar para procesar imágenes para una pluralidad de vistas en la preparación para decodificación de las imágenes usando la extensión de codificación de vídeo multivista (MVC) del Estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 1600.

40 El método 1600 incluye un bloque de inicio 1605 que pasa el control a un bloque de función 1615. El bloque de función 1615 analiza sintácticamente un elemento de sintaxis `num_coded_views_minus1` y pasa el control a un bloque de función 1620. El bloque de función 1620 analiza sintácticamente un elemento de sintaxis `view_id[i]` para todas las $(\text{num_coded_views_minus1} + 1)$ vistas y pasa el control a un bloque de función 1625. El bloque de función 1625 analiza sintácticamente la información de dependencia de referencia entre vistas para imágenes de anclaje y
45 pasa el control a un bloque de función 1630. El bloque de función 1630 analiza sintácticamente la información de dependencias de referencia entre vistas para imágenes no ancladas y pasa el control a un bloque de función 1635. El bloque de función 1635 analiza sintácticamente un elemento de sintaxis `pseudo_view_present_flag` y pasa el control a un bloque de decisión 1640. El bloque de decisión 1640 determina si el valor actual del elemento de sintaxis `pseudo_view_present_flag` es igual o no a verdadero. Si es así, entonces se pasa el control a un bloque de
50 función 1645. De otro modo, el control pasa a un bloque final 1699.

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

Volviendo a la FIG. 17, un método ejemplar para decodificación de imágenes para una pluralidad de vistas usando la extensión de codificación de vídeo multivista (MVC) del Estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 1700.

5 El método 1700 incluye un bloque de inicio 1702 que se inicia con el parámetro de entrada pseudo_view_id y pasa el control a un bloque de función 1704. El bloque de función 1704 analiza sintácticamente un elemento de sintaxis num_sub_views_minus1 y pasa el control a un bloque de función 1706. El bloque de función 1706 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 menor o no que el número de sub_views. Si es así, entonces se pasa el control a un bloque de función 1710. De otro modo, se pasa el control a un bloque de función 1720.

10 El bloque de función 1710 analiza sintácticamente un elemento de sintaxis sub_view_id[i] y pasa el control a un bloque de función 1712. El bloque de función 1712 analiza sintácticamente un elemento de sintaxis num_parts_minus1[sub_view_id[i]] y pasa el control a un bloque de función 1714. El bloque de función 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 menor o no que el elemento de sintaxis num_parts_minus1[sub_view_id[i]]. Si es así, entonces se pasa el control a un bloque de función de 1718. De otro modo, se pasa el control a un bloque de decisión 1722.

El bloque de función 1718 fija los siguientes elementos de sintaxis, aumenta 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].

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

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

El bloque de decisión 1722 determina si un elemento de sintaxis tiling_mode es igual o no a cero. Si es así, entonces se pasa el control a un bloque de función 1724. De otro modo, se pasa el control a un bloque de función 1738.

30 El bloque de función 1724 analiza sintácticamente un elemento de sintaxis flip_dir[sub_view_id[i]] y un elemento de sintaxis upsample_view_flag[sub_view_id[i]] y pasa el control a un bloque de decisión 1726. El bloque de decisión 1726 determina si el valor actual del elemento de sintaxis upsample_view_flag[sub_view_id[i]] es igual o no a uno. Si es así, entonces se pasa el control a un bloque de función 1728. De otro modo, el control se pasa a un bloque de decisión 1730.

35 El bloque de función 1728 analiza sintácticamente un elemento de sintaxis upsample_filter[sub_view_id[i]] y pasa el control al bloque de decisión 1730. El bloque de decisión 1730 determina si un valor del elemento de sintaxis upsample_filter[sub_view_id[i]] es igual o no a tres. Si es así, se pasa el control a un bloque de función 1732. De otro modo, se pasa el control a un bloque de función 1736.

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

El bloque de función 1736 aumenta la variable i y devuelve el control al bloque de decisión 1708.

45 El bloque de función 1738 analiza sintácticamente un elemento de sintaxis pixel_dist_x[sub_view_id[i]] y el elemento de sintaxis flip_dist_y[sub_view_id[i]] y pasa el control a un bloque de función 1740. El bloque de función 1740 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 menor o no que el valor actual del elemento de sintaxis num_parts[sub_view_id[i]]. Si es así, entonces se pasa el control a un bloque de función 1744. De otro modo, se pasa el control al bloque de función 1736.

50 El bloque de función 1744 analiza sintácticamente un elemento de sintaxis num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]] y pasa el control a un bloque de función 1746. El bloque de función 1746 analiza los coeficientes para todos los filtros de enlosamiento de píxeles y pasa el control al bloque de función 1736.

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

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

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

Volviendo a la FIG. 19, un método ejemplar para codificar imágenes para una pluralidad de vistas y profundidades usando la extensión de codificación de vídeo multivista (MVC) del Estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 1900.

El método 1900 incluye un bloque de inicio 1902 que pasa el control a un bloque de función 1904. El bloque de función 1904 fija un elemento de sintaxis `num_sub_views_minus1` y pasa el control a un bloque de función 1906. El bloque de función 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 menor o no que el número de `sub_views`. Si es así, entonces se pasa el control a un bloque de función de 1910. De otro modo, se pasa el control a un bloque de función 1920.

El bloque de función 1910 fija un elemento de sintaxis `sub_view_id[i]` y pasa el control a un bloque de función 1912. El bloque de función 1912 fija un elemento de sintaxis `num_parts_minus1[sub_view_id[i]]` y pasa el control a un bloque de función 1914. El bloque de función 1914 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 menor o no que el elemento de sintaxis `num_parts_minus1[sub_view_id[i]]`. Si es así, entonces se pasa el control a un bloque de función 1918. De otro modo, se pasa el control a un bloque de decisión 1922.

El bloque de función 1918 fija los siguientes elementos de sintaxis, aumenta 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 de función 1920 codifica la profundidad actual para la vista actual usando codificación de vídeo multivista (MVC) y pasa el control a un bloque final 1999. La señal de profundidad se puede codificar similar a la forma que se codifica su señal de vídeo correspondiente. Por ejemplo, la señal de profundidad para una vista se puede incluir en una losa que incluye solamente otras señales de profundidad o solamente las señales de vídeo o tanto las señales de profundidad como de vídeo. La losa (seudovista) entonces se trata como una única vista para MVC y también hay presumiblemente otras losas que se tratan como otras vistas para MVC.

El bloque de decisión 1922 determina si un elemento de sintaxis `tiling_mode` es igual o no a cero. Si es así, entonces se pasa el control a un bloque de función 1924. De otro modo, se pasa el control a un bloque de función 1938.

El bloque de función 1924 fija un elemento de sintaxis `flip_dir[sub_view_id[i]]` y un elemento de sintaxis `upsample_view_flag[sub_view_id[i]]` y pasa el control a un bloque de decisión 1926. El bloque de decisión 1926 determina si valor actual del elemento de sintaxis `upsample_view_flag[sub_view_id[i]]` es igual o no a uno. Si es así, entonces se pasa el control a un bloque de función de 1928. De otro modo, se pasa el control a un bloque de decisión 1930.

El bloque de función 1928 fija un elemento de sintaxis `upsample_filter[sub_view_id[i]]` y pasa el control al bloque de decisión 1930. El bloque de decisión 1930 determina si un valor del elemento de sintaxis `upsample_filter[sub_view_id[i]]` es igual o no a tres. Si es así, se pasa el control a un bloque de función de 1932. De otro modo, se pasa el control a un bloque de función 1936.

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

El bloque de función 1936 aumenta la variable `i` y devuelve el control al bloque de decisión 1908.

- El bloque de función 1938 fija un elemento de sintaxis `pixel_dist_x[sub_view_id[i]]` y el elemento de sintaxis `flip_dist_y[sub_view_id[i]]` y pasa el control a un bloque de función 1940. El bloque de función 1940 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 menor o no que el valor actual del elemento de sintaxis `num_parts[sub_view_id[i]]`. Si es así, entonces se pasa el control a un bloque de función 1944. De otro modo, se pasa el control al bloque de función 1936.
- El bloque de función 1944 fija un elemento de sintaxis `num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]]` y pasa el control a un bloque de función 1946. El bloque de función 1946 fija los coeficientes de todos los filtros de enlosamiento de píxeles y pasa el control al bloque de función 1936.
- 10 Volviendo a la FIG. 20, un método ejemplar para procesar imágenes para una pluralidad de vistas y profundidades en la preparación para decodificar las imágenes usando la extensión de codificación de vídeo multivista (MVC) del Estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 2000.
- El método 2000 incluye un bloque de inicio 2005 que pasa el control a un bloque de función 2015. El bloque de función 2015 analiza sintácticamente un elemento de sintaxis `num_coded_views_minus1` y pasa el control a un bloque de función 2020. El bloque de función 2020 analiza sintácticamente un elemento de sintaxis `view_id[i]` para todas las $(\text{num_coded_views_minus1} + 1)$ profundidades que corresponden a `view_id[i]` y pasa el control a un bloque de función 2025. El bloque de función 2025 analiza sintácticamente la información de dependencia de referencia entre vistas para imágenes de profundidad de anclaje y pasa el control a un bloque de función 2030. El bloque de función 2030 analiza sintácticamente la información de dependencia de referencia entre vistas para imágenes de profundidad no de anclaje y pasa el control a un bloque de función 2035. El bloque de función 2035 analiza sintácticamente un elemento de sintaxis `pseudo_view_present_flag` y pasa el control a un bloque de decisión 2040. El bloque de decisión 2040 determina si el valor actual del elemento de sintaxis `pseudo_view_present_flag` es igual o no a verdadero. Si es así, entonces se pasa el control a un bloque de función 2045. De otro modo, se pasa el control a un bloque final 2099.
- 25 El bloque de función 2045 analiza sintácticamente los siguientes elementos de sintaxis y pasa el control a un bloque de función 2050: `tiling_mode`; `org_pic_width_in_mbs_minus1`; y `org_pic_height_in_mbs_minus1`. El bloque de función 2050 llama a un elemento de sintaxis `pseudo_view_info(view_id)` para cada vista codificada y pasa el control al bloque final 2099.
- Volviendo a la FIG. 21, un método ejemplar para decodificar imágenes para una pluralidad de vistas y profundidades usando la extensión de codificación de vídeo multivista (MVC) del Estándar de AVC del MPEG-4 se indica de manera general por el número de referencia 2100.
- El método 2100 incluye un bloque de inicio 2102 que se inicia con el parámetro de entrada `pseudo_view_id` y pasa el control a un bloque de función 2104. El bloque de función 2104 analiza sintácticamente un elemento de sintaxis `num_sub_views_minus1` y pasa el control a un bloque de función 2106. El bloque de función 2106 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 menor o no que el número de `sub_views`. Si es así, entonces se pasa el control a un bloque de función 2110. De otro modo, se pasa el control a un bloque de función 2120.
- 35 El bloque de función 2110 analiza sintácticamente un elemento de sintaxis `sub_view_id[i]` y pasa el control a un bloque de función 2112. El bloque de función 2112 analiza sintácticamente un elemento de sintaxis `num_parts_minus1[sub_view_id[i]]` y pasa el control a un bloque de función 2114. El bloque de función 2114 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 menor o no que el elemento de sintaxis `num_parts_minus1[sub_view_id[i]]`. Si es así, entonces se pasa el control a un bloque de función 2118. De otro modo, se pasa el control al bloque de decisión 2122.
- 40 El bloque de función 2118 fija los siguientes elementos de sintaxis, aumenta 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]`.
- 45 El bloque de función 2120 decodifica la imagen actual usando codificación de vídeo multivista (MVC) y pasa el control a un bloque de función 2121. El bloque de función 2121 separa cada vista de la imagen usando la sintaxis de alto nivel y pasa el control a un bloque final 2199. La separación de cada vista usando la sintaxis de alto nivel es como se describió previamente.
- 50 El bloque de decisión 2122 determina si un elemento de sintaxis `tiling_mode` es igual o no a cero. Si es así, entonces se pasa el control a un bloque de función 2124. De otro modo, se pasa el control a un bloque de función 2138.
- El bloque de función 2124 analiza sintácticamente un elemento de sintaxis `flip_dir[sub_view_id[i]]` y un elemento de sintaxis `upsample_view_flag[sub_view_id[i]]` y pasa el control a un bloque de decisión 2126. El bloque de decisión 2126 determina si el valor actual del elemento de sintaxis `upsample_view_flag[sub_view_id[i]]` es igual o no a uno. Si
- 55

es así, entonces se pasa el control a un bloque de función 2128. De otro modo, se pasa el control a un bloque de decisión 2130.

5 El bloque de función 2128 analiza sintácticamente un elemento de sintaxis `upsample_filter[sub_view_id[i]]` y pasa el control al bloque de decisión 2130. El bloque de decisión 2130 determina si un valor del elemento de sintaxis `upsample_filter[sub_view_id[i]]` es igual o no a tres. Si es así, se pasa el control a un bloque de función 2132. De otro modo, se pasa el control a un bloque de función 2136.

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

El bloque de función 2136 aumenta la variable *i* y devuelve el control al bloque de decisión 2108.

15 El bloque de función 2138 analiza sintácticamente un elemento de sintaxis `pixel_dist_x(sub_view_id[i])` y el elemento de sintaxis `flip_dist_y[sub_view_id[i]]` y pasa el control a un bloque de función 2140. El bloque de función 2140 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 menor o no que el valor actual del elemento de sintaxis `num_parts sub_view_id[[i]]`. Si es así, entonces se pasa el control a un bloque de función 2144. De otro modo, se pasa el control al bloque de función 2136.

20 El bloque de función 2144 analiza sintácticamente un elemento de sintaxis `num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]]` y pasa el control a una bloque de función 2146. El bloque de función 2146 analiza sintácticamente los coeficientes para todos los filtros de enlosamiento del píxeles y pasa el control al bloque de función 2136.

Volviendo a la FIG. 22, ejemplos de enlosamiento a nivel de píxel se indican de manera general por el número de referencia 2200. La FIG. 22 se describe más adelante aún más.

Enlosamiento de vista usando AVC o MVC del MPEG-4

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

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

35 Una vez que la profundidad se codifica como una losa del cuadro entero, la sintaxis de alto nivel debería indicar qué losa es la señal de profundidad de manera que el representador pueda usar la señal de la profundidad adecuadamente.

40 En el caso cuando la secuencia de entrada (tal como la mostrada en la FIG. 1) se codifique usando un codificador de Estándar de AVC del MPEG-4 (o un codificador que corresponde a un estándar y/o 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 Suplementaria (SEI). Una realización del método propuesto se muestra en la TABLA 1 donde la sintaxis está presente en un mensaje de Información de Mejora Suplementaria (SEI).

45 En el caso cuando las secuencias de entrada de las pseudovistas (tales como las mostradas en la FIG. 1) se codifiquen usando la extensión de codificación de vídeo multivista (MVC) del codificador de Estándar de AVC del MPEG-4 (o un codificador que corresponde a un estándar de codificación de vídeo con respecto a un estándar y/o recomendación de codificación de vídeo diferente), la sintaxis de alto nivel propuesta puede estar presente en el SPS, el PPS, la cabecera de segmento, un mensaje de SEI o un perfil especificado. Una realización del método propuesto se muestra en la TABLA 1. La TABLA 1 muestra elementos de sintaxis presentes en la estructura de Conjunto de Parámetros de Secuencia (SPS), incluyendo los elementos de sintaxis propuestos de acuerdo con una
50 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)
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 de sintaxis para el elemento de sintaxis pseudo_view_info de la TABLA 1, de acuerdo con una realización de los presentes principios.

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)
}		
}		

	C	Descriptor
pseudo_view_info(pseudo_view_id){		
if (tiling_mode == 0) {		
flip_dir[sub_view_id[i]][i]	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_filter[sub_view_id[i]] ==3) {		
vert_dim[sub_view_id[i]]	5	ue(v)
hor_dim[sub_view_id[i]]	5	ue(v)
quantizer[sub_view_id[i]]	5	ue(v)
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]		
} // 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 de sintaxis presentados en la TABLA 1 y 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 enlosadas a nivel de imagen. Un valor de 1 indica que el enlosamiento se hace a nivel de pixel.

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

- 10 num_coded_views_minus1 plus1 indica el número de vistas codificadas soportado por el flujo de bits. El valor de num_coded_views_minus1 está en el ámbito de 0 a 1023, ambos inclusive.

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

La variable para la anchura de imagen en unidades de macrobloques se deriva como sigue:

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

- 15 La variable para anchura de imagen para el componente luminancia se deriva como sigue:

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

La variable para anchura de imagen para los componentes de croma se deriva como sigue:

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

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

5 La variable para la altura de imagen en unidades de macrobloques se deriva como sigue:

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

La variable para la altura de imagen para el componente de luminancia se deriva como sigue:

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

La variable para la altura de imagen para los componentes de croma se deriva como sigue:

10 $\text{PicHeightInSamplesC} = \text{PicHeightInMbs} * \text{MbHeightC}$

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

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

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

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

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

20 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 términos de una región rectangular especificada en coordenadas del cuadro para la salida.

Las variables CropUnitX y CropUnitY se derivan como sigue:

- Si chroma_format_idc es igual a 0, CropUnitX y CropUnitY se derivan como sigue:

25 $\text{CropUnitX} = 1$

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

- De otro modo (chroma_format_idc es igual a 1, 2 o 3), CropUnitX y CropUnitY se derivan como sigue:

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

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

30 El rectángulo de recorte de cuadro incluye muestras de luminancia con coordenadas de cuadro horizontal a partir de lo siguiente:

35 $\text{CropUnitX} * \text{frame_crop_left_offset}$ a $\text{PicWidthInSamplesL} - (\text{CropUnitX} * \text{frame_crop_right_offset} + 1)$ y coordenadas de cuadro vertical a partir de $\text{CropUnitY} * \text{frame_crop_top_offset}$ a $(16 * \text{FrameHeightInMbs}) - (\text{CropUnitY} * \text{frame_crop_bottom_offset} + 1)$, ambos inclusive. El valor de frame_crop_left_offset estará en el intervalo de 0 a $(\text{PicWidthInSamplesL}/\text{CropUnitX}) - (\text{frame_crop_right_offset} + 1)$, ambos inclusive; y el valor frame_crop_top_offset estará en el intervalo de 0 a $(16 * \text{FrameHeightInMbs}/\text{CropUnitY}) - (\text{frame_crop_bottom_offset} + 1)$, ambos inclusive.

40 Cuando chroma_format_idc no es igual a 0, las muestras especificadas correspondientes de las dos disposiciones de croma son las muestras que tienen coordenadas de cuadro (x / SubWidthC y / SubHeightC), donde (x, y) son las coordenadas de cuadro de las muestras de luminancia especificadas.

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

num_parts[view_id[i]] especifica el número de partes en las que se divide 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 con la vista identificada por view_id[i].

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

flip_dir[view_id [i]][j] especifica la dirección de volteado para la parte actual. flip_dir igual a 0 indica que no hay volteado, flip_dir igual a 1 indica volteado en una dirección horizontal, flip_dir igual a 2 indica volteado en una dirección vertical y flip_dir igual a 3 indica volteado en las direcciones horizontal y 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 j actual se sitúa en la imagen final reconstruida de la vista con view_id 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] necesita ser muestreada ascendentemente. upsample_view_flag[view_id[i]] igual a 0 especifica que la imagen con view_id igual a view_id[i] no se muestreará ascendentemente. upsample_view_flag[view_id[i]] igual a 1 especifica que la imagen con
15 view_id igual a view_id[i] se muestreará ascendentemente.

upsample_filter[view_id[i]] indica el tipo de filtro que se va a usar para muestreo ascendente. upsample_filter[view_id[i]] igual a 0 indica que se debería usar el filtro de AVC de 6 derivaciones, upsample_filter[view_id[i]] igual a 1 indica que se debería usar el filtro de AVC de 4 derivaciones, upsample_filter[view_id[i]] 2 indica que se debería usar el filtro bilineal, upsample_filter[view_id[i]] igual a 3 indica que
20 se transmiten los coeficientes de filtros personalizados. Cuando upsample_filter[view_id[i]] no está presente se fija a 0. En esta realización, usamos un filtro personalizado 2D. Se puede extender fácilmente a un filtro de 1D y algún otro filtro no lineal.

vert_dim[view_id[i]] especifica la dimensión vertical del filtro personalizado de 2D.

hor_dim[view_id[i]] especifica la dimensión horizontal del filtro personalizado de 2D.

25 quantizer[view_id[i]] especifica el factor de cuantificación para cada coeficiente de filtro.

filter_coefs[view_id[i]][yuv][y][x] especifica los coeficientes de filtro cuantificados. yuv señala el componente para el que se aplican los coeficientes de filtro. yuv igual a 0 especifica el componente Y, yuv igual a 1 especifica el componente U y yuv igual a 2 especifica el componente V.

30 pixel_dist_x[sub_view_id[i]] y pixel_dist_y[sub_view_id[i]] especifica, respectivamente, la distancia en la dirección horizontal y la dirección vertical en la pseudovista final reconstruida entre píxeles vecinos en la vista con sub_view_id igual a sub_view_id[i].

num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]][j] más uno indica el número de los coeficientes de filtro cuando el modo de enlosamiento se fija igual a 1.

35 pixel_tiling_filter_coefs[sub_view_id [i]][j] señala los coeficientes de filtro que se requieren para representar un filtro que se puede usar para filtrar la imagen enlosada.

Ejemplos de enlosamiento a nivel de píxel

Volviendo a la FIG. 22, dos ejemplos que muestran la composición de una pseudovista enlosando píxeles de cuatro vistas se indican respectivamente por los números de referencia 2210 y 2220, respectivamente. Las cuatro vistas se indican colectivamente por el número de referencia 2250. Los valores de sintaxis para el primer ejemplo en la FIG.
40 22 se proporcionan en la TABLA 3 de más adelante.

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

5 El desplazamiento indica que los píxeles que corresponden a una vista deberían comenzar en una cierta ubicación de desplazamiento. Esto se muestra en la FIG. 22 (2220). Esto se puede hacer, por ejemplo, cuando dos vistas producen imágenes en las que objetos comunes aparecen desplazados de una vista a otra. Por ejemplo, si una primera y segunda cámaras (que representan una primera y segunda vistas) toman imágenes de un objeto, el objeto puede parecer que está desplazado cinco píxeles a la derecha en la segunda vista en comparación con la primera vista. Esto significa que pixel(i-5, j) en la primera vista corresponde al pixel(i, j) en la segunda vista. Si los píxeles de las dos vistas están simplemente enlosados píxel por píxel, entonces puede que no haya mucha correlación entre píxeles vecinos en la losa y las ganancias de codificación espacial pueden ser pequeñas. Por el contrario, desplazando el enlosamiento de manera que pixel(i-5, j) de la vista una se coloque cerca del pixel(i, j) de la vista dos, se puede aumentar la correlación espacial y también se puede aumentar la ganancia de codificación espacial. Esto se deduce porque, por ejemplo, los píxeles correspondientes para el objeto en la primera y segunda vistas están siendo enlosados unos cerca de otros.

De esta manera, la presencia de loc_left_offset y loc_top_offset puede beneficiar a la eficacia de codificación. La información de desplazamiento se puede obtener por medios externos. Por ejemplo, la información de posición de las cámaras o los vectores de disparidad global entre las vistas se puede usar para determinar tal información de desplazamiento.

20 Como resultado del desplazamiento, algunos píxeles en la seudovista no se asignan a valores de píxeles de cualquier vista. Continuando con el ejemplo anterior, cuando se enlosa el pixel(i-5, j) de la vista uno junto al pixel(i, j) de la vista dos, para valores de i = 0 4 no hay ningún pixel(i-5, j) de la vista uno para enlosar, de manera que esos píxeles están vacíos en la losa. Para esos píxeles en la seudovista (losa) que no se asignan a valores de píxel de cualquier vista, al menos una implementación usa un procedimiento de interpolación similar al procedimiento de interpolación subpíxel en la compensación de movimiento en el AVC. Es decir, los píxeles de losa vacía se pueden interpolar a partir de los píxeles vecinos. Tal interpolación puede provocar una correlación espacial mayor en la losa y una ganancia de codificación mayor para la losa.

En codificación de vídeo, podemos elegir un tipo de codificación diferente para cada imagen, tal como imágenes I, P y B. Para codificación de vídeo multivista, además, definimos imágenes de anclaje y no de anclaje. En una realización, proponemos que la decisión de agrupación se pueda basar en el tipo de imagen. Esta información de agrupación se señalada en la sintaxis de alto nivel.

35 Volviendo a la FIG. 11, un ejemplo de 5 vistas enlosadas en un único cuadro se indica de manera general por el número de referencia 1100. En particular, la secuencia de baile se muestra con 5 vistas enlosadas en un único cuadro. Adicionalmente, se puede ver que la quinta vista se divide en dos partes de manera que se pueda disponer en un cuadro rectangular. Aquí, cada vista es de tamaño de QVGA de manera que la dimensión total de cuadro es 640x600. Dado que 600 no es un múltiplo de 16 se debería extender a 608.

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

TABLA 4

	Valor
multiview_display_info(payloadSize) {	
num_coded_views_minus1	5
org_pic_width_in_mbs_minus1	40
org_pic_height_in_mbs_minus1	30
view_id[0]	0
num_parts[view_id[0]]	1
depth_flag[view_id[0]][0]	0
flip_dir[view_id[0]][0]	0
loc_left_offset[view_id[0]][0]	0
loc_top_offset[view_id[0]][0]	0
frame_crop_left_offset[view_id[0]][0]	0
frame_crop_right_offset[view_id[0]][0]	320
frame_crop_top_offset[view_id[0]][0]	0
frame_crop_bottom_offset[view_id[0]][0]	240
upsample_view_flag[view_id[0]]	1
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
loc_top_offset[view_id[1]][0]	0
frame_crop_left_offset[view_id[1]][0]	320
frame_crop_right_offset[view_id[1]][0]	640
frame_crop_top_offset[view_id[1]][0]	0
frame_crop_bottom_offset[view_id[1]][0]	320
upsample_view_flag[view_id[1]]	1
if(upsample_view_flag[view_id[1]]) {	
vert_dim[view_id[1]]	6
hor_dim[view_id[1]]	6

quantizer[view_id[1]]	32
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
.....(igualmente para la vista 2, 3)	
view_id[4]	4
num_parts[view_id[4]]	2
depth_flag[view_id[0]][0]	0
flip_dir[view_id[4]][0]	0
loc_left_offset[view_id[4]][0]	0
loc_top_offset[view_id[4]][0]	0
frame_crop_left_offset[view_id[4]][0]	0
frame_crop_right_offset[view_id[4]][0]	320
frame_crop_top_offset[view_id[4]][0]	480
frame_crop_bottom_offset[view_id[4]][0]	600
flip_dir[view_id[4]][1]	0
loc_left_offset[view_id[4]][1]	0
loc_top_offset[view_id[4]][1]	120
frame_crop_left_offset[view_id[4]][1]	320
frame_crop_right_offset[view_id[4]][1]	640
frame_crop_top_offset[view_id[4]][1]	480
frame_crop_bottom_offset[view_id[4]][1]	600
upsample_view_flag[view_id[4]]	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 general de sintaxis para la transmisión de información multivista para el ejemplo mostrado en la TABLA 4.

TABLA 5

multiview_display_info(payloadSize) {	C	Descriptor
num_coded_views_minus1	5	ue(v)
org_pic_width_in_mbs_minus1	5	ue(v)

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

Con 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 receptor multimedia digital u otro dispositivo que recibe vídeo codificado y proporciona, por ejemplo, vídeo decodificado para mostrar a un usuario o para almacenamiento. De esta manera, el dispositivo 2300 puede proporcionar su salida a un televisor, un monitor de ordenador o un ordenador u otro dispositivo de procesamiento.

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

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

De manera más general, los decodificadores de diversas implementaciones descritas en esta solicitud pueden proporcionar una salida decodificada que incluye una losa entera. Adicional o alternativamente, los decodificadores pueden proporcionar una salida decodificada que incluye solamente una o más imágenes seleccionadas (imágenes o señales de profundidad, por ejemplo) de la losa.

5 Como se señaló anteriormente, se puede usar una sintaxis de alto nivel para realizar señalización de acuerdo con una o más realizaciones de los presentes principios. La sintaxis de alto nivel se puede usar, por ejemplo, pero no se limita a, señalización de cualquiera de los siguientes: el número de vistas codificadas presentes en el cuadro más grande; la anchura y la altura original de todas las vistas; para cada vista codificada, el identificador de vista que corresponde a la vista; para cada vista codificada, el número de partes del cuadro en las que se divide una vista; para cada parte de la vista, la dirección de volteado (que puede ser, por ejemplo, no voltear, solamente volteado horizontal, solamente volteado vertical o volteado horizontal y vertical); para cada parte de la vista, la posición izquierda en píxeles o el número de macrobloques donde la parte actual pertenece en el cuadro final para la vista; para cada parte de la vista, la posición superior de la parte en píxeles o número de macrobloques donde la parte actual pertenece en el cuadro final para la vista; para cada parte de la vista, la posición izquierda, en el cuadro decodificado/codificado grande actual, de la ventana de recorte en píxeles o el número de macrobloques; para cada parte de la vista, la posición derecha, en el cuadro decodificado/codificado grande actual, de la ventana de recorte en píxeles o número de macrobloques; para cada parte de la vista, la posición superior, en el cuadro decodificado/codificado grande actual, de la ventana de recorte en píxeles o número de macrobloques; y, para cada parte de la vista, la posición inferior, en el cuadro decodificado/codificado grande actual, de la ventana de recorte en píxeles o número de macrobloques; para cada vista codificada si la vista necesita ser muestreada ascendentemente antes de la salida (donde si el muestreo ascendente necesita ser realizado, una sintaxis de alto nivel se puede usar para indicar el método para muestreo ascendente (incluyendo, pero no limitado a, un filtro de 6 derivaciones de AVC, un filtro de 4 derivaciones de SVC, un filtro bilineal o un filtro lineal o no lineal de 1D, 2D personalizado).

25 Se tiene que señalar que los términos “codificador” y “decodificador” connotan estructuras generales y no se limitan a ninguna función o rasgo particular. Por ejemplo, un decodificador puede recibir una portadora modulada que transporta un flujo de bits codificado y demodular el flujo de bits codificado, así como decodificar el flujo de bits.

30 Se han descrito diversos métodos. Muchos de estos métodos se detallan para proporcionar una amplia descripción. Se señala, no obstante, que se contemplan variaciones que pueden variar uno o muchos de los rasgos específicos descritos para estos métodos. Además, muchos de los rasgos que se exponen son conocidos en la técnica y, en consecuencia, no se describen con gran detalle.

Además, se ha hecho referencia al uso de sintaxis de alto nivel para enviar cierta información en diversas implementaciones. Se tiene que entender, no obstante, que otras implementaciones usan sintaxis de nivel inferior o de hecho otros mecanismos en conjunto (tales como, por ejemplo, enviar información como parte de los datos codificados) para proporcionar la misma información (o variaciones de esa información).

35 Diversas implementaciones proporcionan enlosamiento y señalización adecuada para permitir que múltiples vistas (imágenes, de manera más general) sean enlosadas en una única imagen, codificadas como una única imagen y enviadas como una única imagen. La información de señalización puede permitir a un procesador posterior separar las vistas/imágenes. También, las múltiples imágenes que están enlosadas podrían ser vistas, pero al menos una de las imágenes podría ser información de profundidad. Estas implementaciones pueden proporcionar una o más ventajas. Por ejemplo, los usuarios pueden querer mostrar múltiples vistas de una forma enlosada y estas diversas implementaciones proporcionan una forma eficiente para codificar y transmitir o almacenar tales vistas enlosándolas anterior a codificarlas y transmitir/almacenarlas de manera enlosada.

45 Implementaciones que enlosan múltiples vistas en el contexto de AVC y/o MVC también proporcionan ventajas adicionales. AVC se usa ostensiblemente solamente para una única vista, de manera que no se espera una vista adicional. No obstante, tales implementaciones basadas en AVC pueden proporcionar múltiples vistas en un entorno de AVC debido a que las vistas enlosadas se pueden disponer de manera que, por ejemplo, un decodificador sabe que las imágenes enlosadas 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.).

50 Adicionalmente, MVC ya incluye múltiples vistas, de manera que no se espera que múltiples vistas sean incluidas en una única seudovista. Además, MVC tiene un límite en el número de vistas que se pueden soportar y tales implementaciones basadas en MVC aumentan eficazmente el número de vistas que se pueden soportar permitiendo (como en las implementaciones basadas en AVC) que vistas adicionales sean enlosadas. Por ejemplo, cada seudovista puede corresponder a una de las vistas soportadas de MVC y el decodificador puede saber que cada “vista soportada” incluye en realidad cuatro vistas en un orden de enlosamiento predispuesto. De esta manera, en tal implementación, el número de vistas posibles es cuatro veces el número de “vistas soportadas”.

55 Las implementaciones descritas en la presente memoria se pueden implementar, por ejemplo, en un método o proceso, un aparato o un programa software. Incluso si solamente se discute en el contexto de una única forma de implementación (por ejemplo, discutido solamente como un método), la implementación de los rasgos discutidos también se puede implementar en otras formas (por ejemplo, un aparato o programa). Un aparato se puede

implementar, por ejemplo, en hardware, software y microprogramas adecuados. Los métodos se pueden implementar, por ejemplo, en un aparato tal como, por ejemplo, un procesador, que se refiere a dispositivos de procesamiento en general, incluyendo, por ejemplo, un ordenador, un microprocesador, un circuito integrado o un dispositivo de lógica programable. Los dispositivos de procesamiento también incluyen dispositivos de comunicación, tales como, por ejemplo, ordenadores, teléfonos móviles, asistentes digitales portátiles/personales (“PDA”) y otros dispositivos que facilitan la comunicación de información entre usuarios finales.

Las implementaciones de los diversos procesos y rasgos descritos en la presente memoria se pueden incorporar en una variedad de diferentes equipos o aplicaciones, particularmente, por ejemplo, equipos o aplicaciones asociados con codificación y decodificación de datos. Ejemplos de equipos incluyen codificadores de vídeo, decodificadores de vídeo, códec de vídeo, servidores web, receptores multimedia digitales, ordenadores portátiles, ordenadores personales, teléfonos móviles, PDA y otros dispositivos de comunicación. Como debería quedar claro, el equipo puede ser móvil e incluso estar instalado en un vehículo móvil.

Adicionalmente, los métodos se pueden implementar por instrucciones que se realizan por un procesador y tales instrucciones se pueden almacenar en un medio legible por procesador tal como, por ejemplo, un circuito integrado, un portador de software u otro dispositivo de almacenamiento tal como, por ejemplo, un disco duro, un disquete compacto, una memoria de acceso aleatorio (“RAM”) o una memoria de sólo lectura (“ROM”). Las instrucciones pueden formar un programa de aplicaciones incorporado tangiblemente en un medio legible por procesador. Como debería quedar claro, un procesador puede incluir un medio legible por procesador que tiene, por ejemplo, instrucciones para llevar a cabo un proceso. Tales programas de aplicaciones se pueden cargar en y ejecutar por, una máquina que comprende cualquier arquitectura adecuada. Preferiblemente, la máquina se implementa en una plataforma de ordenador que tiene hardware tal como una o más unidades centrales de proceso (“CPU”), una memoria de acceso aleatorio (“RAM”) e interfaces de entrada/salida (“I/O”). La plataforma de ordenador también puede incluir un sistema operativo y un código de microinstrucciones. Los diversos procesos y funciones descritos en la presente memoria pueden ser o bien parte del código de microinstrucciones o bien parte del programa de aplicaciones o cualquier combinación de los mismos, los cuales se pueden ejecutar por una CPU. Además, otras diversas unidades periféricas se pueden conectar a la plataforma informática tales como una unidad de almacenamiento de datos adicional y una unidad de impresión.

Como debería ser evidente para un experto en la técnica, las implementaciones también pueden producir una señal formateada para transportar información que se puede, por ejemplo, almacenar o transmitir. La información puede incluir, por ejemplo, instrucciones para realizar un método o datos producidos por una de las implementaciones descritas. Tal señal se puede formatear, por ejemplo, como una onda electromagnética (por ejemplo, usando una parte del espectro de radiofrecuencia) o como una señal en banda base. El formateado puede incluir, por ejemplo, codificar un flujo de datos, producir la sintaxis y modular una portadora con el flujo de datos codificado y la sintaxis. La información que transporta la señal puede ser, por ejemplo, analógica o información digital. La señal se puede transmitir sobre una variedad de diferentes enlaces cableados o inalámbricos, como es conocido.

Se tiene que entender además que, debido a que algunos de los componentes y métodos del sistema constituyente representados en los dibujos anexos se implementan preferiblemente en software, las conexiones reales entre los componentes del sistema o los bloques de función del proceso pueden diferir dependiendo de la manera en que se programen los presentes principios. Dadas las enseñanzas de la presente memoria, un experto ordinario en la técnica pertinente será capaz de contemplar estas implementaciones o configuraciones y similares de los presentes principios.

Se han descrito un número de implementaciones. Sin embargo, se entenderá que se pueden hacer diversas modificaciones. Por ejemplo, los elementos de las diferentes implementaciones se pueden combinar, complementar, modificar o eliminar para producir otras implementaciones. Adicionalmente, un experto en la técnica entenderá que otras estructuras y procesos se pueden sustituir por las descritas y las implementaciones resultantes realizarán al menos sustancialmente la(s) misma(s) función(funciones), de al menos sustancialmente la(s) misma(s) forma(s), para lograr al menos sustancialmente el(los) mismo(s) resultado(s) que las implementaciones descritas.

REIVINDICACIONES

1. Un método que comprende:

5 disponer una primera imagen y una segunda imagen para formar una única imagen, la primera imagen que corresponde a una primera vista de un vídeo multivista y la segunda imagen que corresponde a una segunda vista del vídeo multivista;

generar información que indica cómo la primera imagen y la segunda imagen se combinan en la única imagen, en donde la información generada indica que al menos una de la primera imagen y la segunda imagen está volteada individualmente en una o más de una dirección horizontal y una dirección vertical,

10 - la primera imagen que no está volteada horizontalmente y la segunda imagen que está volteada horizontalmente, la primera imagen y la segunda imagen que están dispuestas lado a lado o

- la primera imagen que no está volteada verticalmente y la segunda imagen que está volteada verticalmente, la primera imagen y la segunda imagen que están dispuestas de arriba a abajo;

y

15 codificar la única imagen y la información generada para formar un flujo de bits, en donde el flujo de bits también incluye información de muestreo ascendente que indica si al menos una de la primera imagen y la segunda imagen va a ser muestreada ascendentemente.

2. El método que se define en la reivindicación 1, en donde la información de muestreo ascendente indica además que ni la primera imagen ni la segunda imagen requiere muestreo ascendente.

20 3. El método que se define en la reivindicación 1, en donde la información de indicación de muestreo ascendente indica que al menos la primera imagen requiere muestreo ascendente; y

en donde la disposición además comprende muestreo descendente de la primera vista para producir la primera imagen.

25 4. El método que se define en la reivindicación 3, en donde la información de indicación de muestreo ascendente indica que la primera imagen y la segunda imagen requieren muestreo ascendente; y

en donde la disposición además comprende muestreo descendente de la primera vista para producir la primera imagen, el muestreo descendente de la segunda vista para producir la segunda imagen y la disposición de la primera imagen y la segunda imagen en la única imagen.

5. Un método que comprende:

30 determinar a partir de un flujo de bits recibido una única imagen e información de muestreo ascendente, en donde la única imagen incluye una primera imagen y una segunda imagen dispuestas como la única imagen, la primera imagen que corresponde a una primera vista de un vídeo multivista y la segunda imagen que corresponde a una segunda vista del vídeo multivista, en donde la información de muestreo ascendente indica si al menos una de la primera imagen y la segunda imagen va a ser muestreada ascendentemente;

35 acceder a la información que indica cómo la primera imagen y la segunda imagen se combinan en la única imagen, en donde la información accedida indica que al menos una de la primera imagen y la segunda imagen está volteada individualmente en una o más de una dirección horizontal y una dirección vertical:

- la primera imagen que no está volteada horizontalmente y la segunda imagen que está volteada horizontalmente, la primera imagen y la segunda imagen que están dispuestas lado a lado o

40 - la primera imagen que no está volteada verticalmente y la segunda imagen que está volteada verticalmente, la primera imagen y la segunda imagen que están dispuestas de arriba a abajo;

y

decodificar la única imagen en la primera imagen y la segunda imagen.

45 6. El método que se define en la reivindicación 5, en donde la información de muestreo ascendente indica que ni la primera imagen ni la segunda imagen requiere muestreo ascendente.

7. El método que se define en la reivindicación 5, en donde la información de muestreo ascendente indica que al menos la primera imagen requiere muestreo ascendente; y

50 en donde la decodificación además comprende desensamblar la primera imagen y la segunda imagen, ambas a partir de la única imagen.

8. El método que se define en cualquiera de las reivindicaciones 1 a 7, en donde la información de muestreo ascendente incluye una indicación de un tipo de filtro para uso en muestreo ascendente.
9. El método que se define en la reivindicación 8, en donde la información de muestreo ascendente incluye además un conjunto de coeficientes de filtro, en donde cada coeficiente de filtro define un valor para un coeficiente particular de un filtro especificado por el tipo de filtro.
10. El método que se define en la reivindicación 9, en donde la información de muestreo ascendente indica que la primera imagen y la segunda imagen requieren muestreo ascendente; y
- en donde la decodificación comprende además muestreo ascendente de la primera imagen para producir una primera vista y muestreo ascendente de la segunda imagen para producir una segunda vista.
11. El método que se define en cualquiera de las reivindicaciones 1 a 10, en donde dicha información de muestreo ascendente se formatea en un mensaje de acuerdo con una sintaxis de alto nivel, en donde el mensaje incluye la información de muestreo ascendente.
12. Un aparato configurado para realizar un método según una o más de las reivindicaciones 1 a 11.
13. Una señal de vídeo formateada para incluir información, la señal de vídeo que comprende:
- una sección de imagen codificada que incluye una codificación de una única imagen, la única imagen que incluye una primera imagen y una segunda imagen dispuestas en la única imagen, la primera imagen que corresponde a una primera vista de un vídeo multivista y la segunda imagen que corresponde a una segunda vista del vídeo multivista; y
- una sección de señalización que incluye una codificación de información que indica cómo la primera imagen y la segunda imagen se combinan en la única imagen, en donde la información indica que al menos una de las múltiples imágenes está volteada individualmente en una o más de una dirección horizontal y una dirección vertical,
- la primera imagen que no está volteada horizontalmente y la segunda imagen que se está volteada horizontalmente, la primera imagen y la segunda imagen que se están dispuestas lado a lado o
- la primera imagen que no está volteada verticalmente y la segunda imagen que está volteada verticalmente, la primera imagen y la segunda imagen que están dispuestas de arriba a abajo,
- en donde la sección de señalización también indica si al menos una de la primera imagen y la segunda imagen va a ser muestreada ascendentemente.
14. Un medio legible por procesador que tiene almacenada en el mismo una estructura de señal de vídeo, la estructura de señal de vídeo que comprende:
- una sección de imagen codificada que incluye una codificación de una única imagen, la única imagen que incluye una primera imagen y una segunda imagen dispuestas en la única imagen, la primera imagen que corresponde a una primera vista de un vídeo multivista y la segunda imagen que corresponde a una segunda vista del vídeo multivista; y
- una sección de señalización que incluye una codificación de información que indica cómo la primera imagen y la segunda imagen se combinan en la única imagen, en donde la información indica que al menos una de las múltiples imágenes está volteada individualmente en una o más de una dirección horizontal y una dirección vertical,
- la primera imagen que no está volteada horizontalmente y la segunda imagen que está volteada horizontalmente, la primera imagen y la segunda imagen que están dispuestas lado a lado o
- la primera imagen que no está volteada verticalmente y la segunda imagen que está volteada verticalmente, la primera imagen y la segunda imagen que están dispuestas de arriba a abajo,
- en donde la sección de señalización también indica si al menos una de la primera imagen y la segunda imagen va a ser muestreada ascendentemente.
15. Un medio legible por máquina que tiene almacenado en el mismo instrucciones ejecutables por máquina que, cuando se ejecutan, implementan un método según una o más de las reivindicaciones 1 a 11.

100

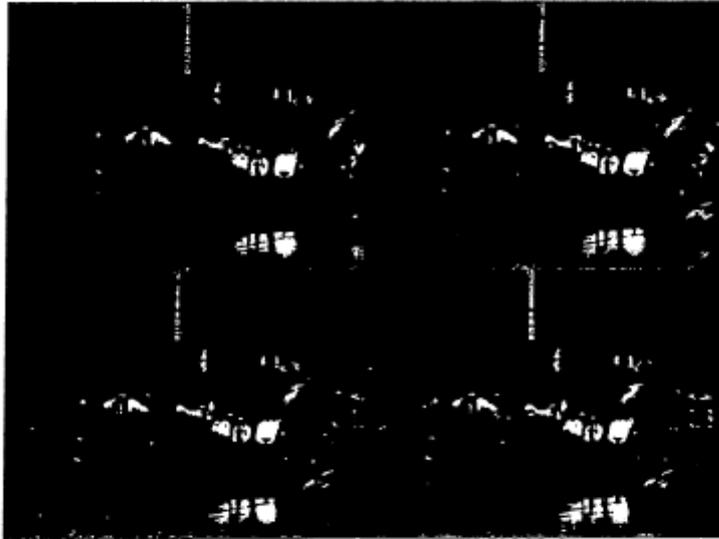


FIG. 1

200

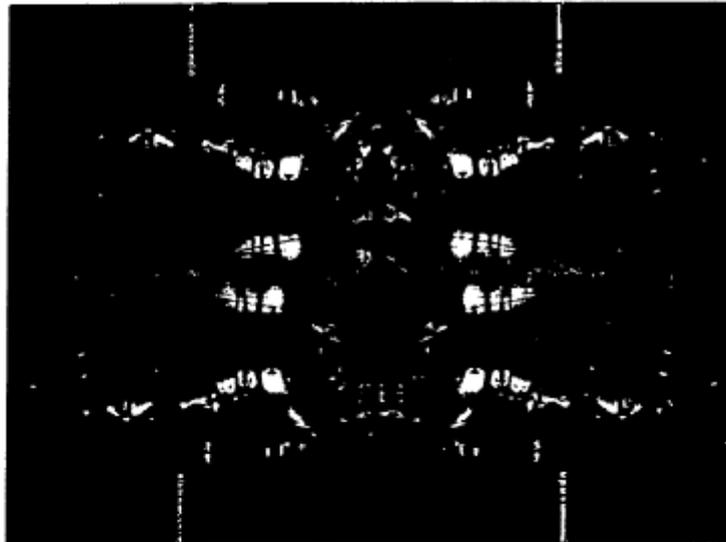


FIG. 2

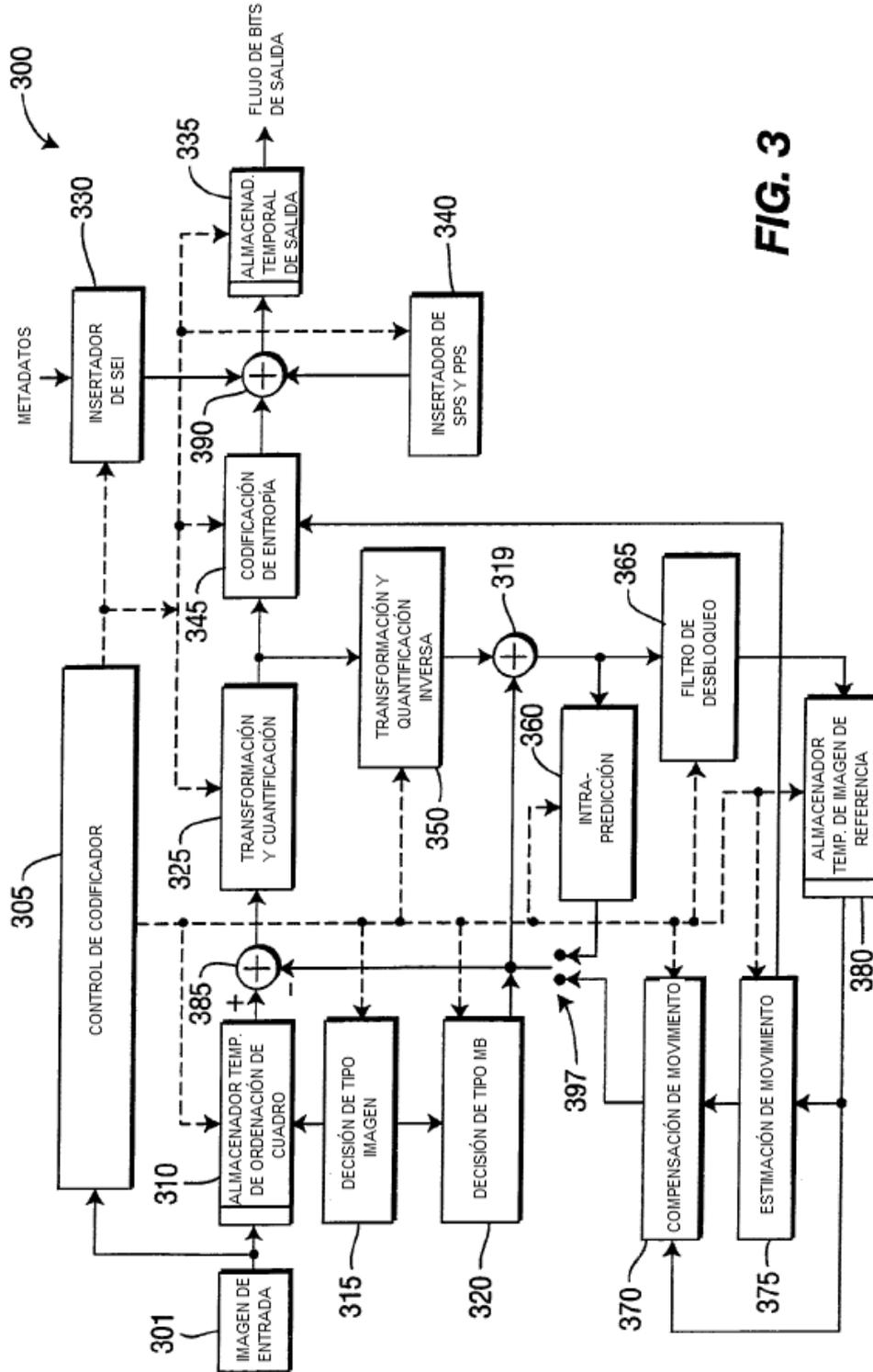


FIG. 3

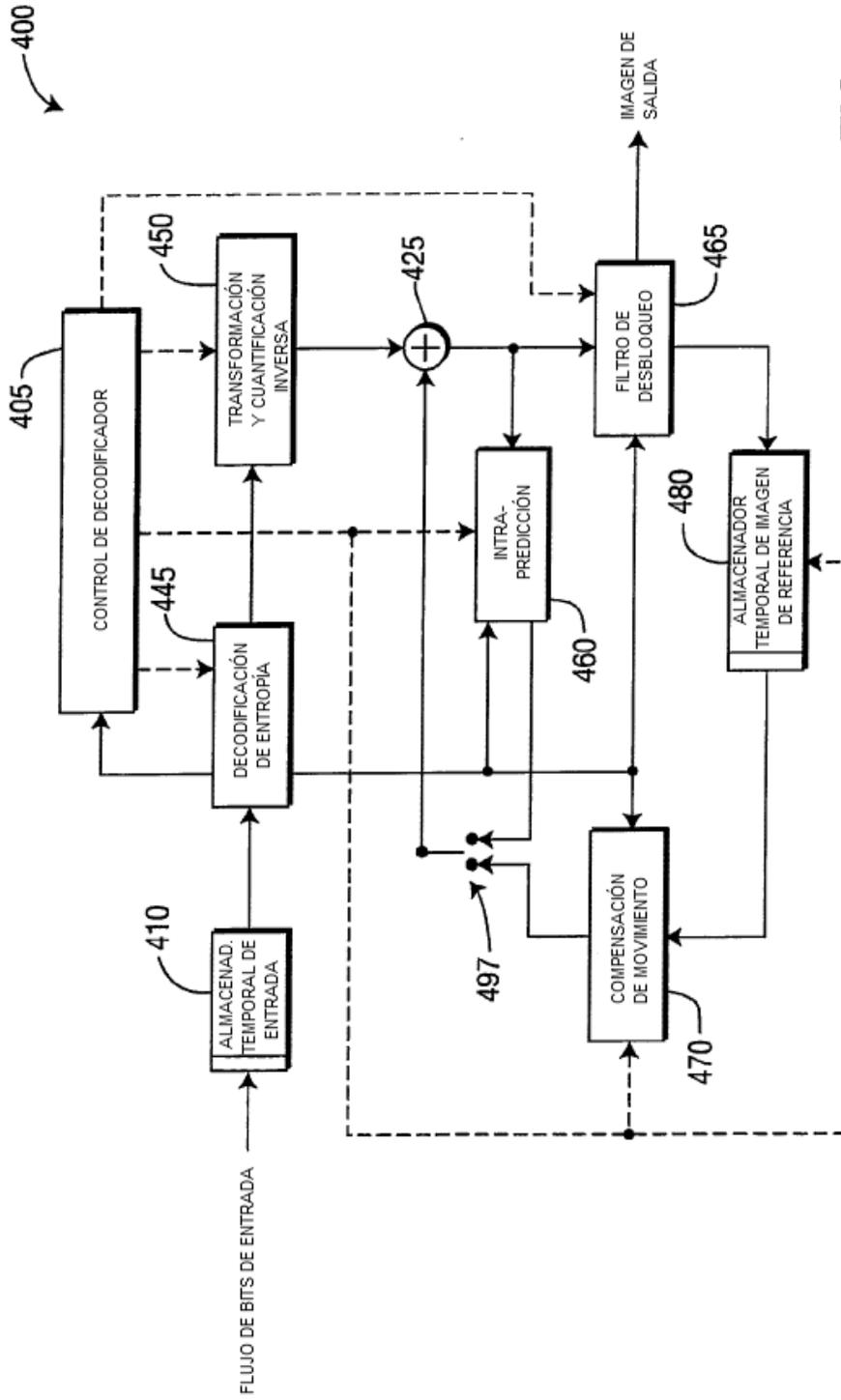
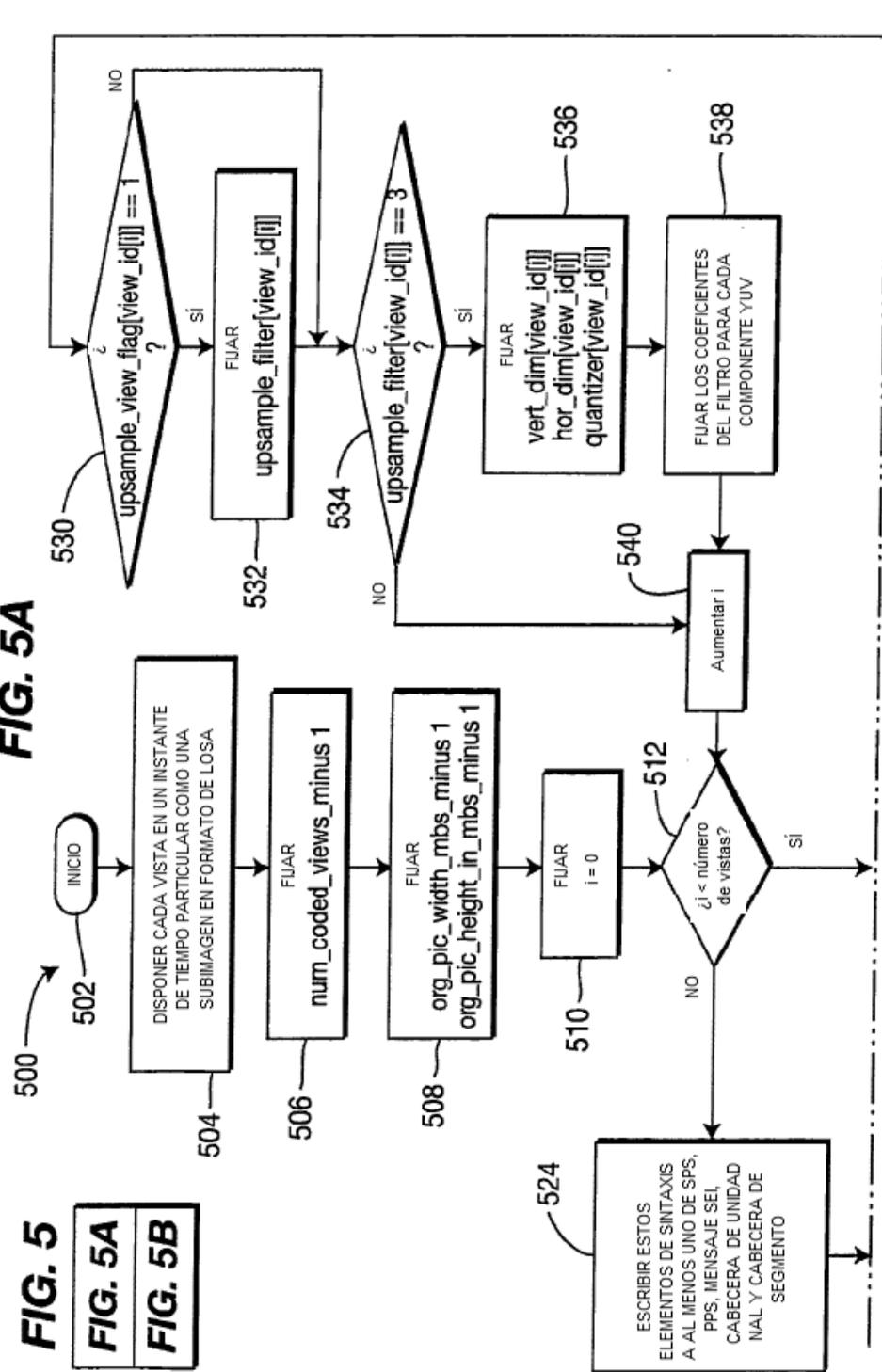


FIG. 4

FIG. 5
FIG. 5A
FIG. 5B



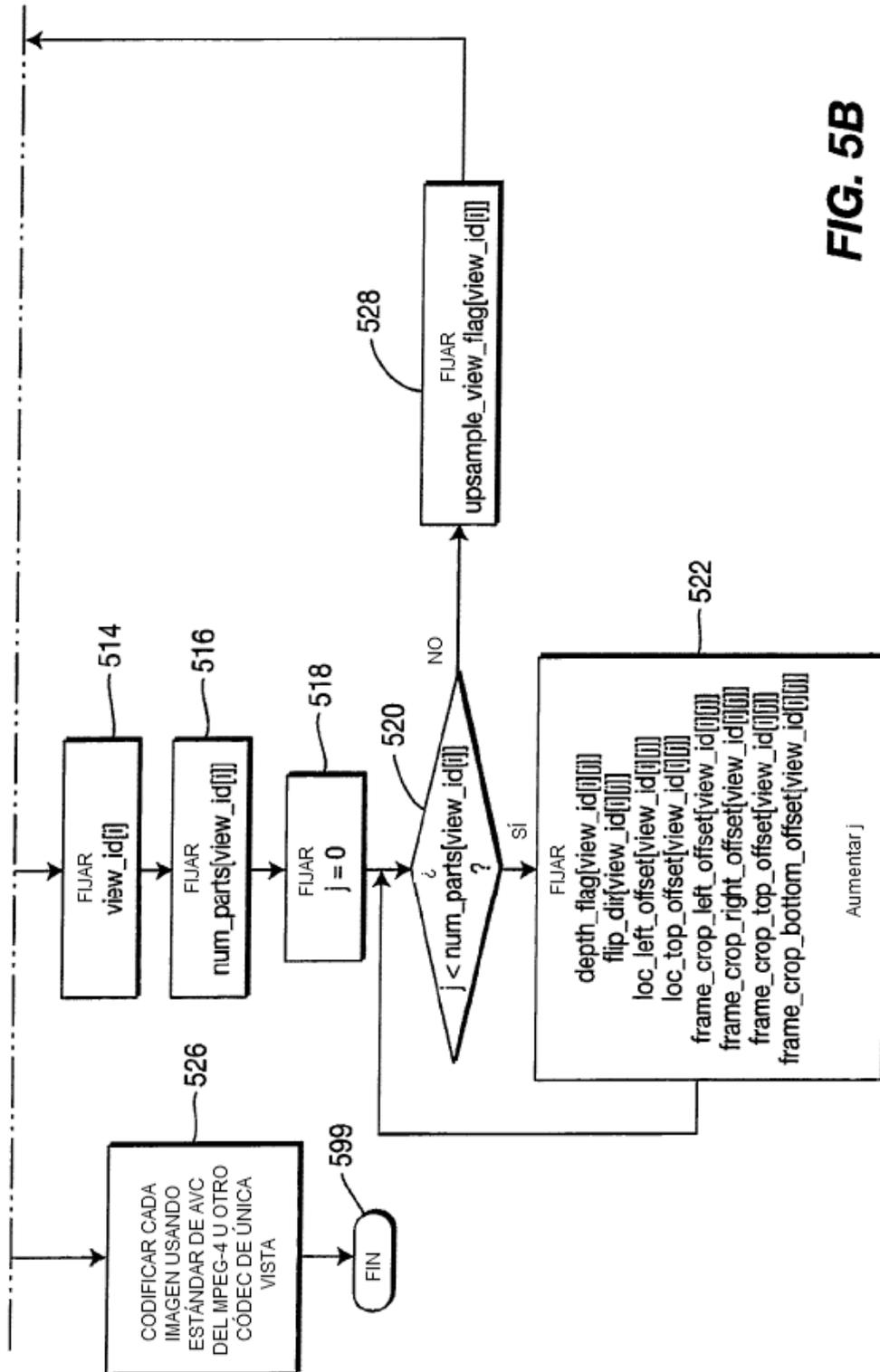
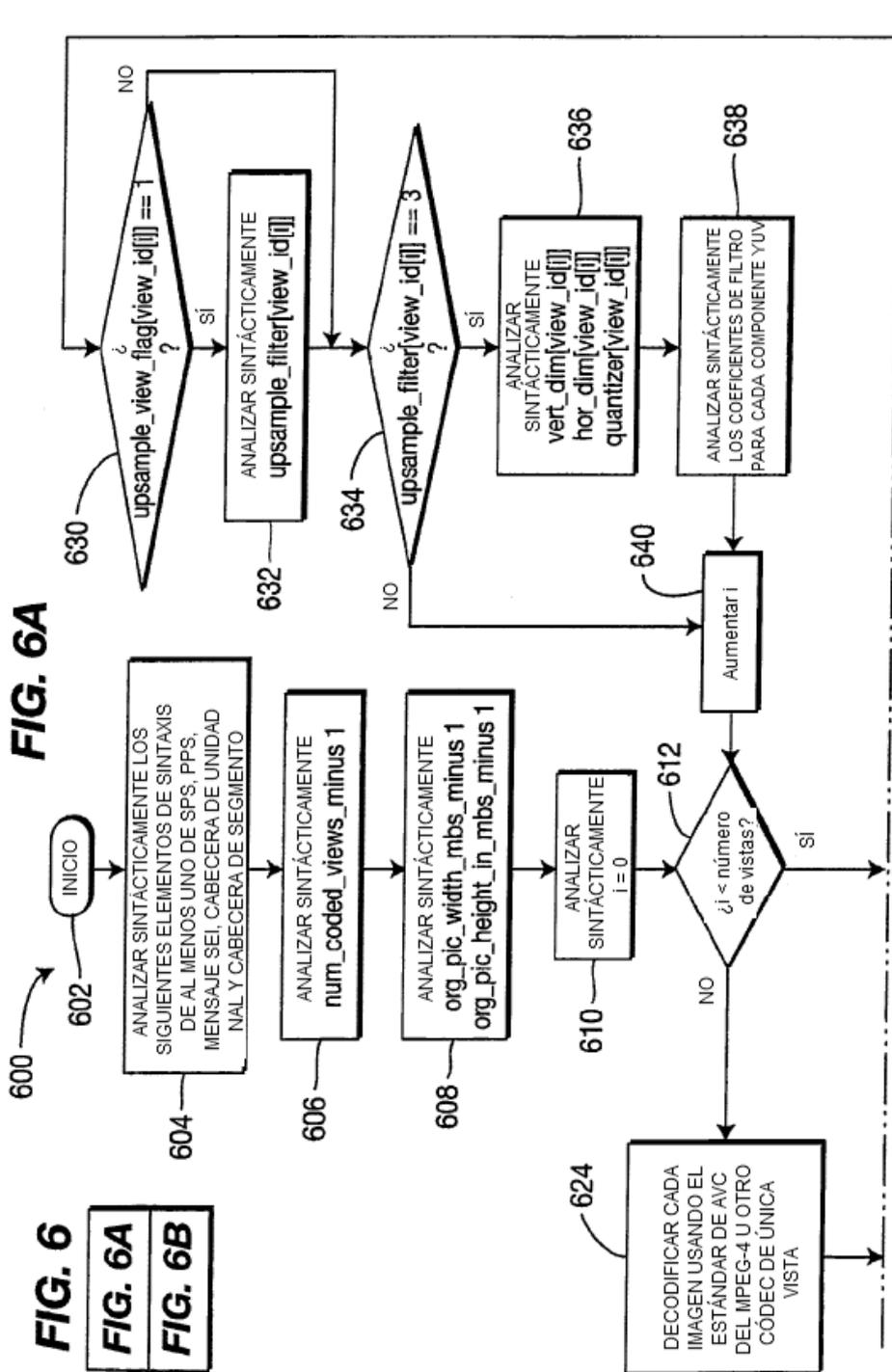


FIG. 5B



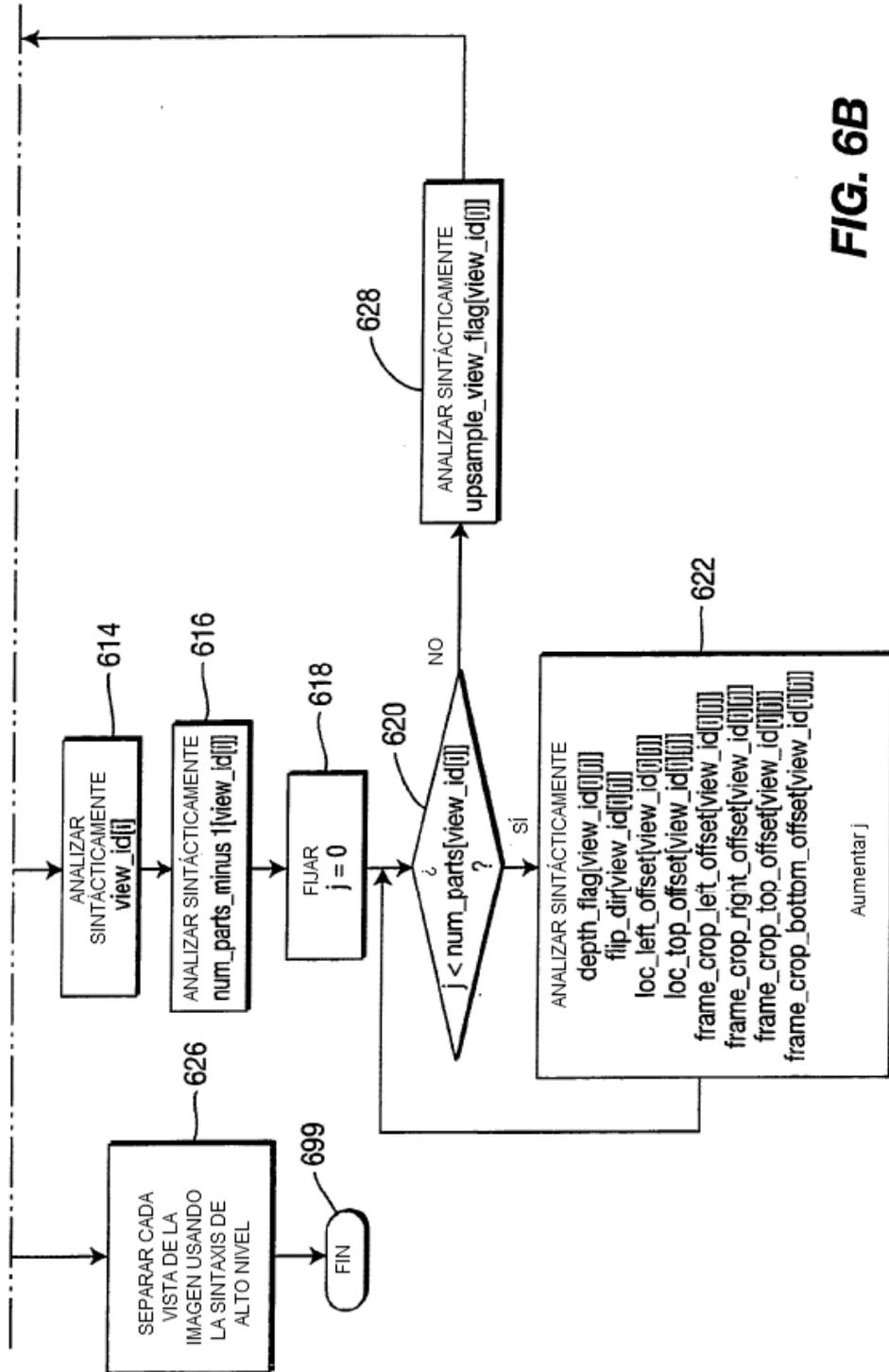
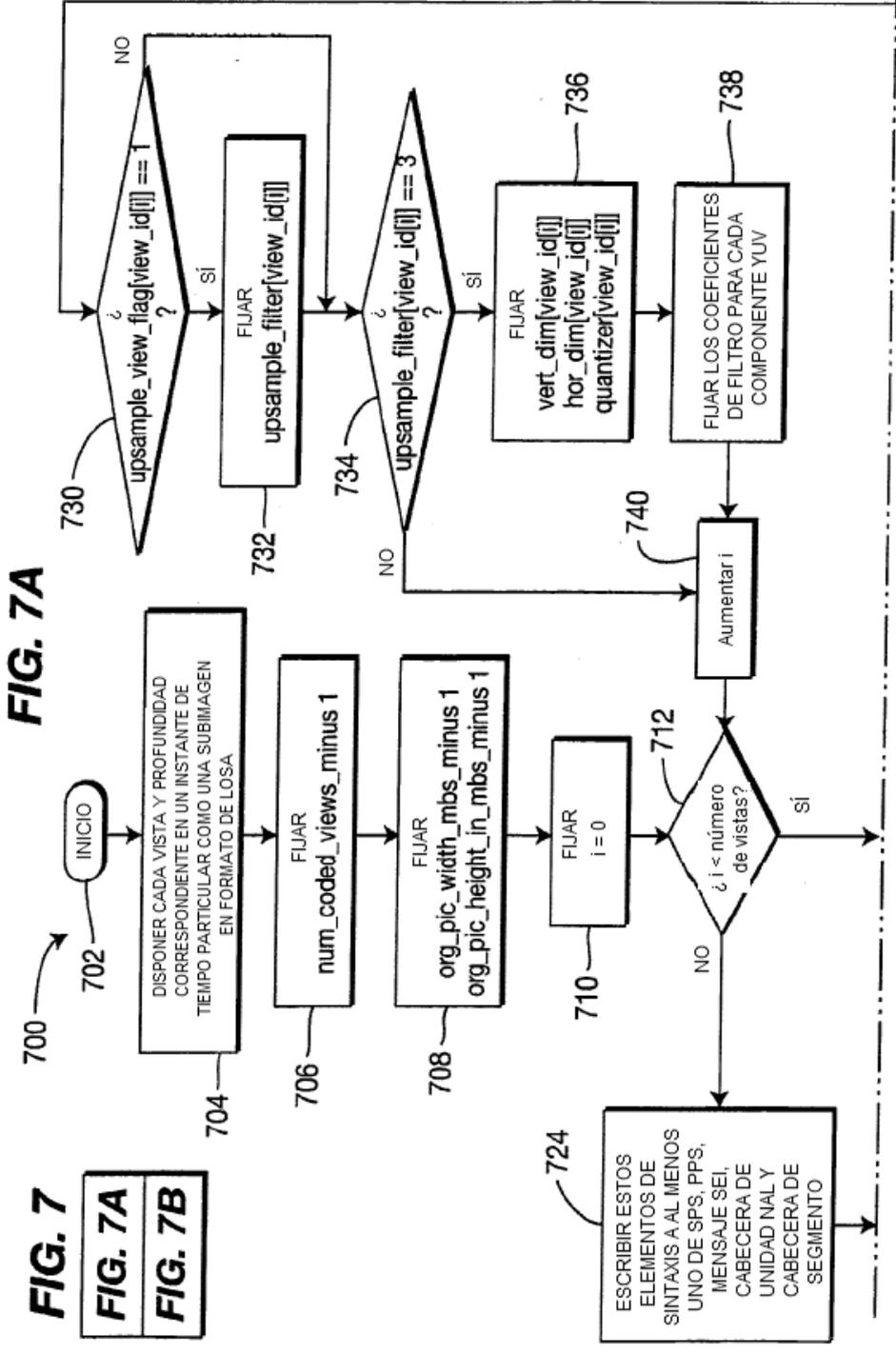


FIG. 6B



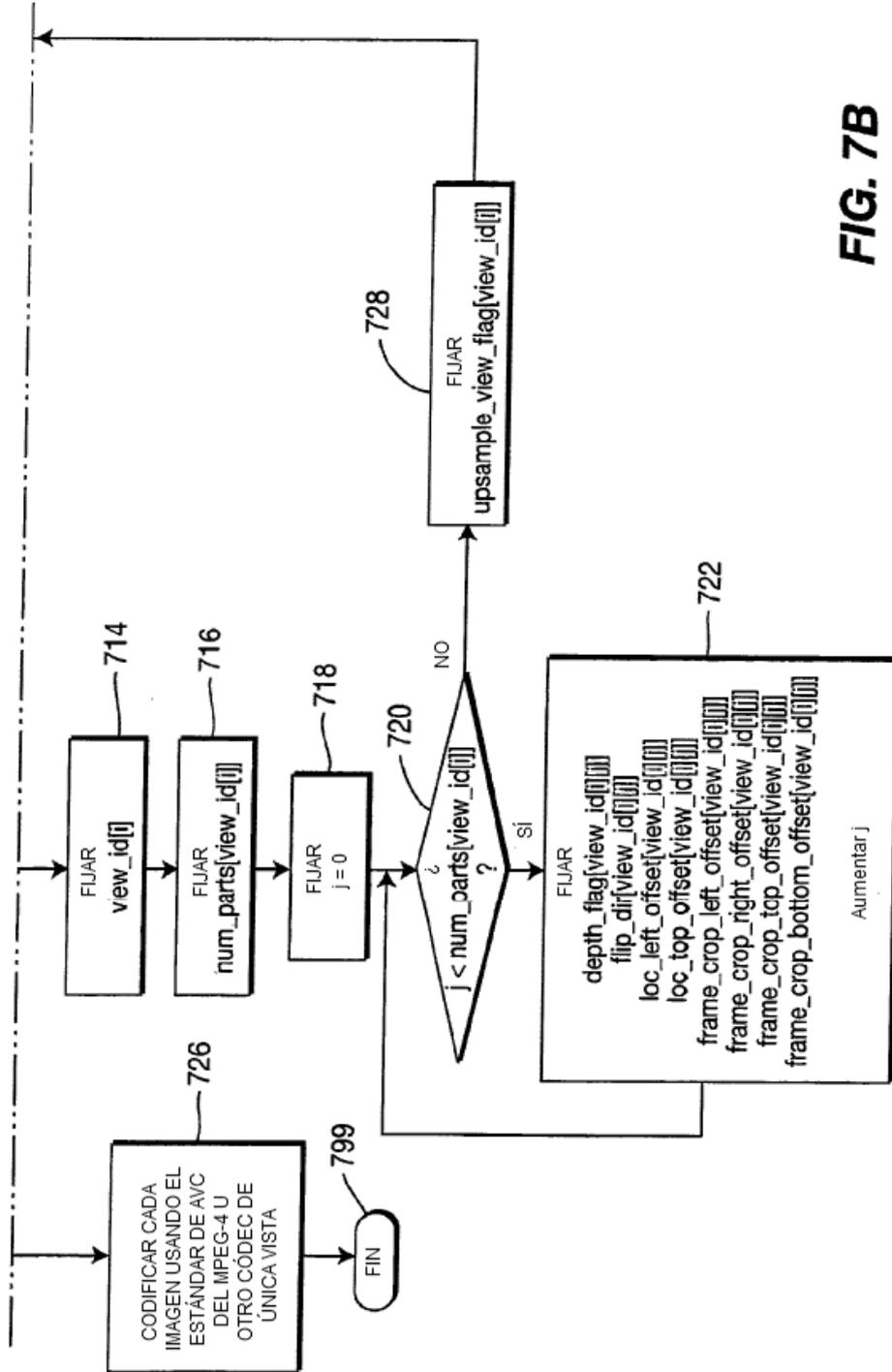
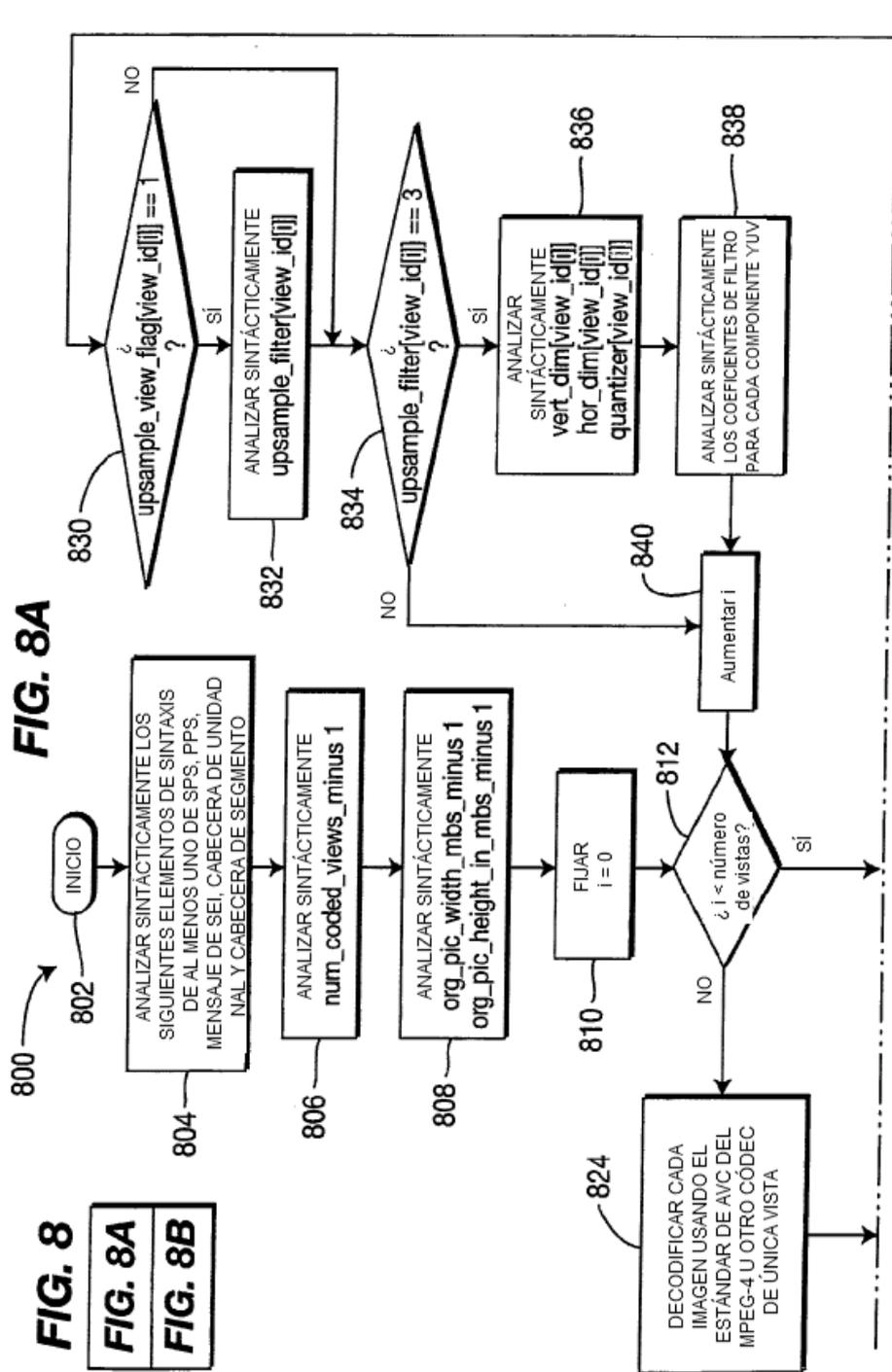


FIG. 7B



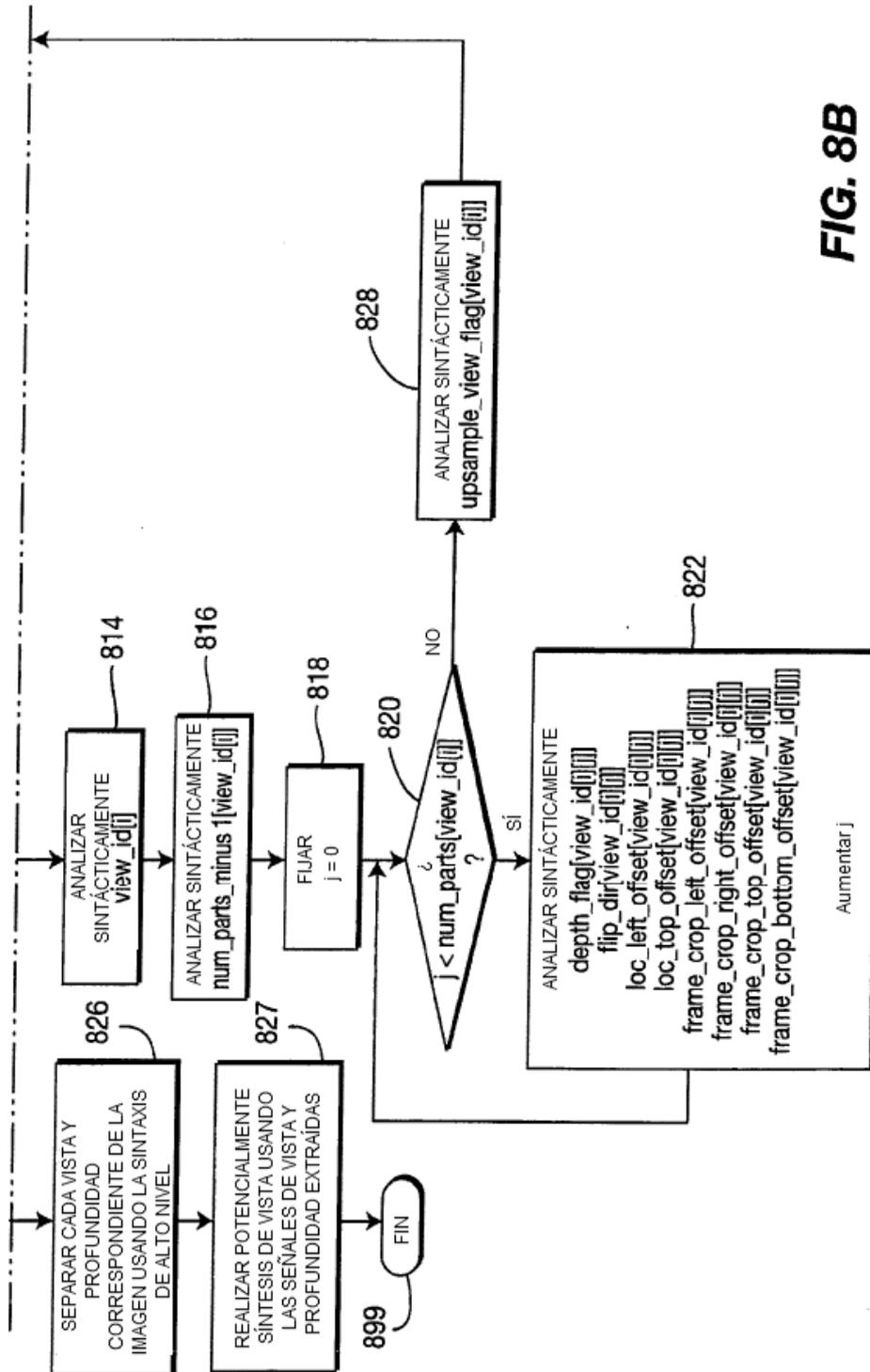


FIG. 8B

900



FIG. 9

1000

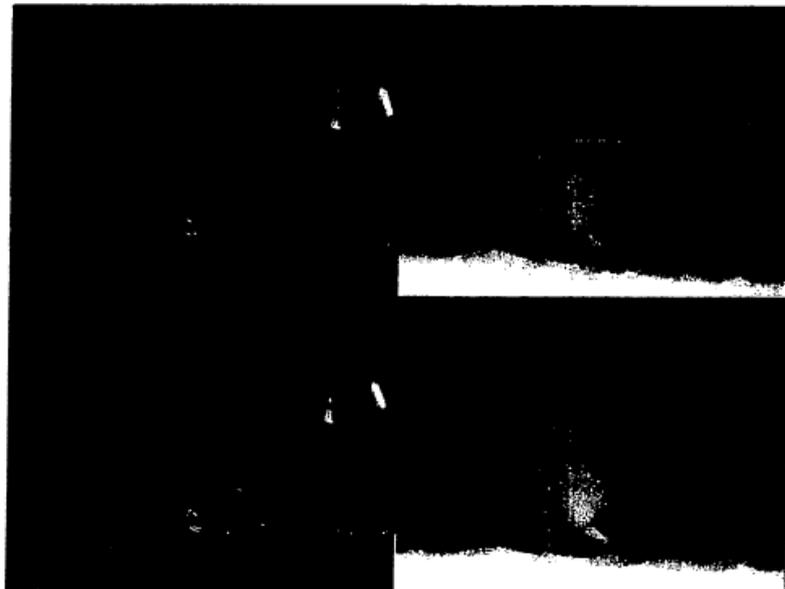


FIG. 10

1100

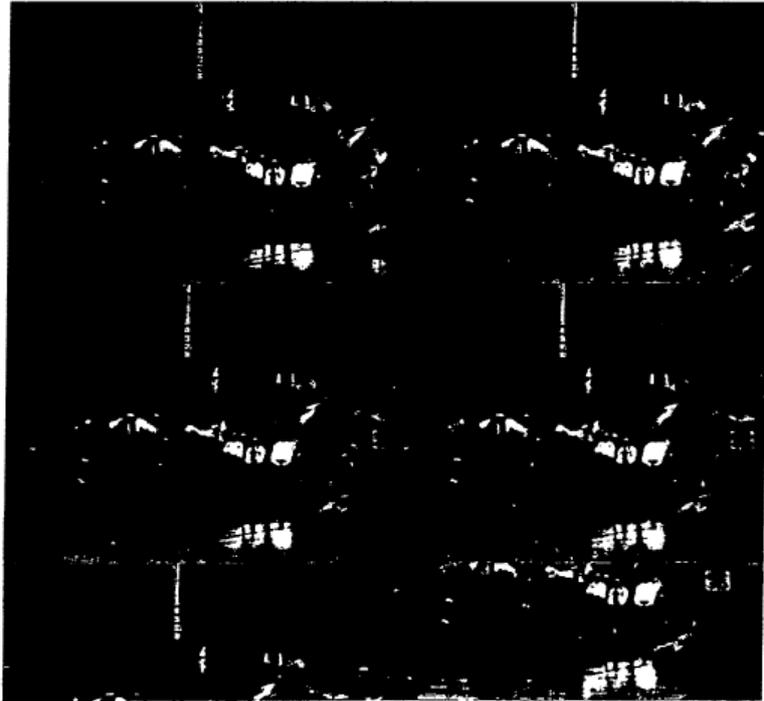


FIG. 11

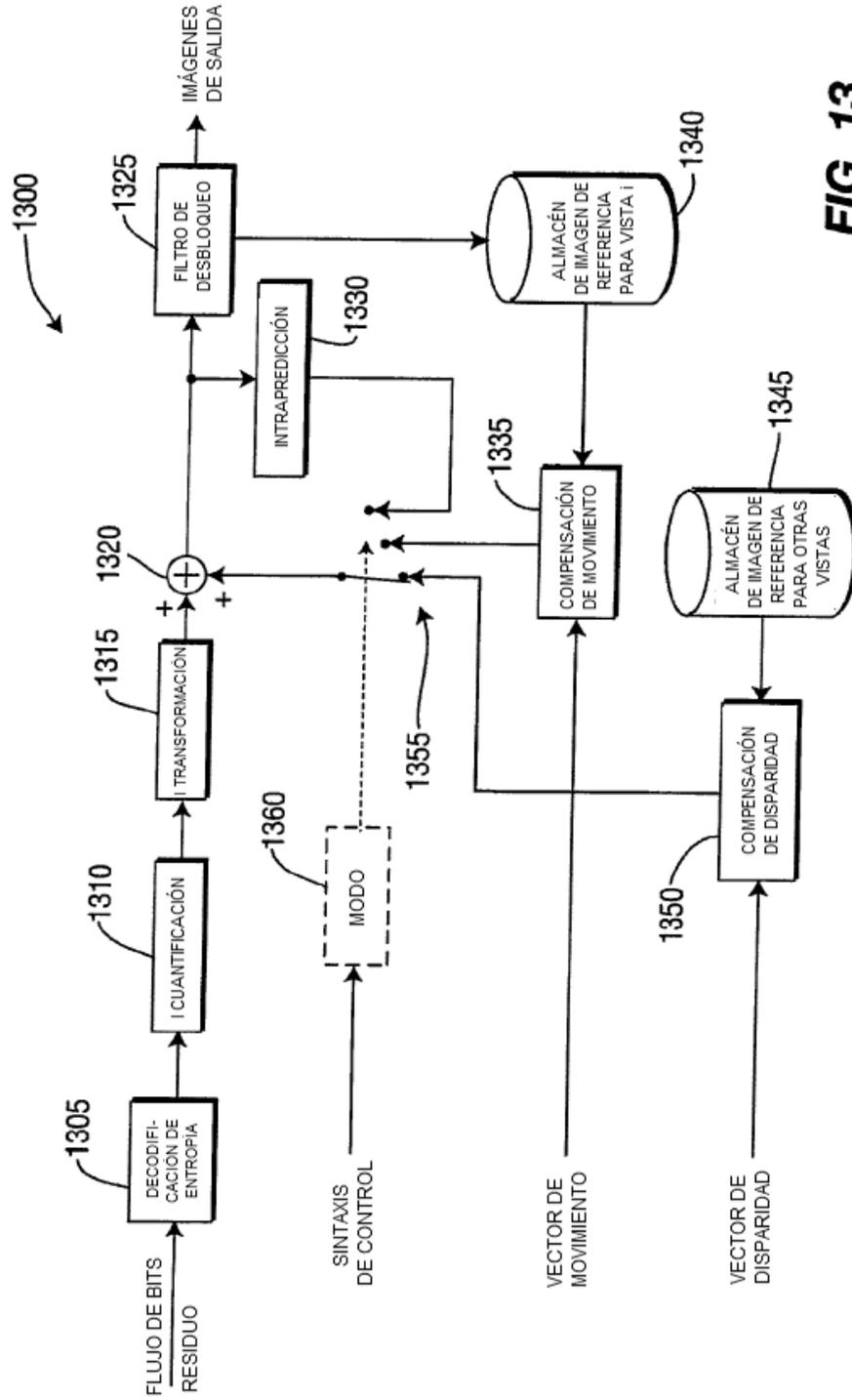


FIG. 13

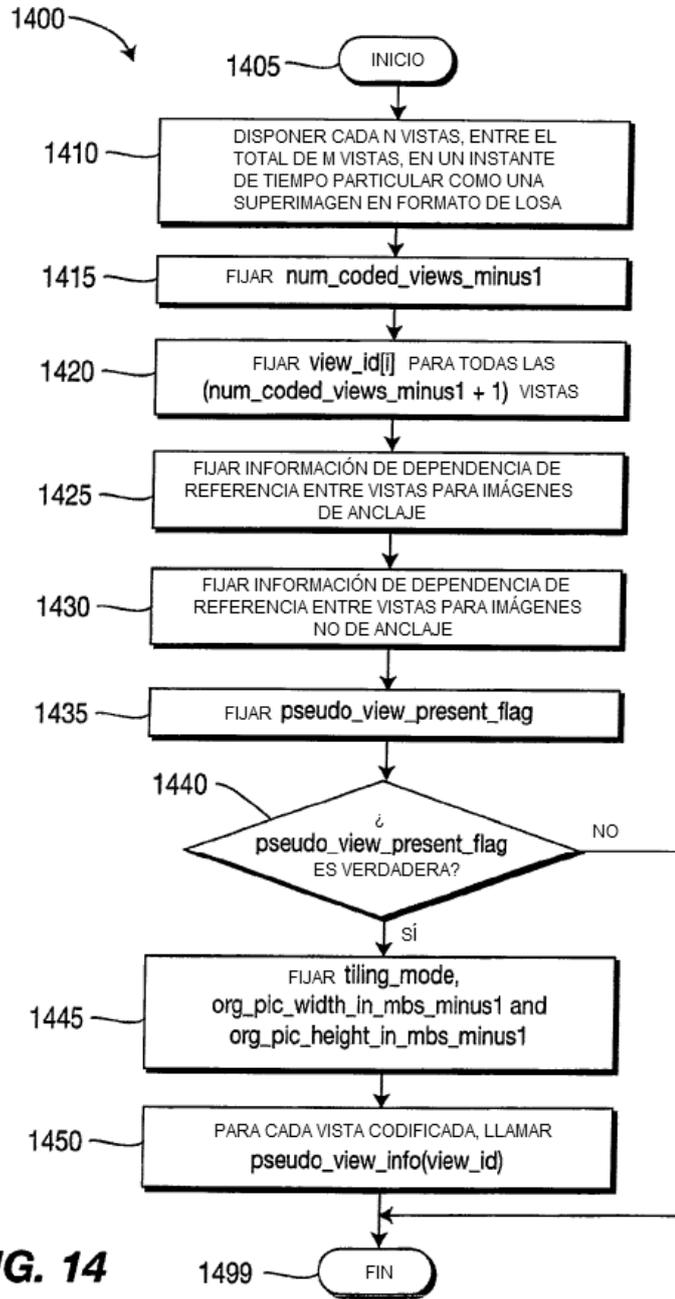
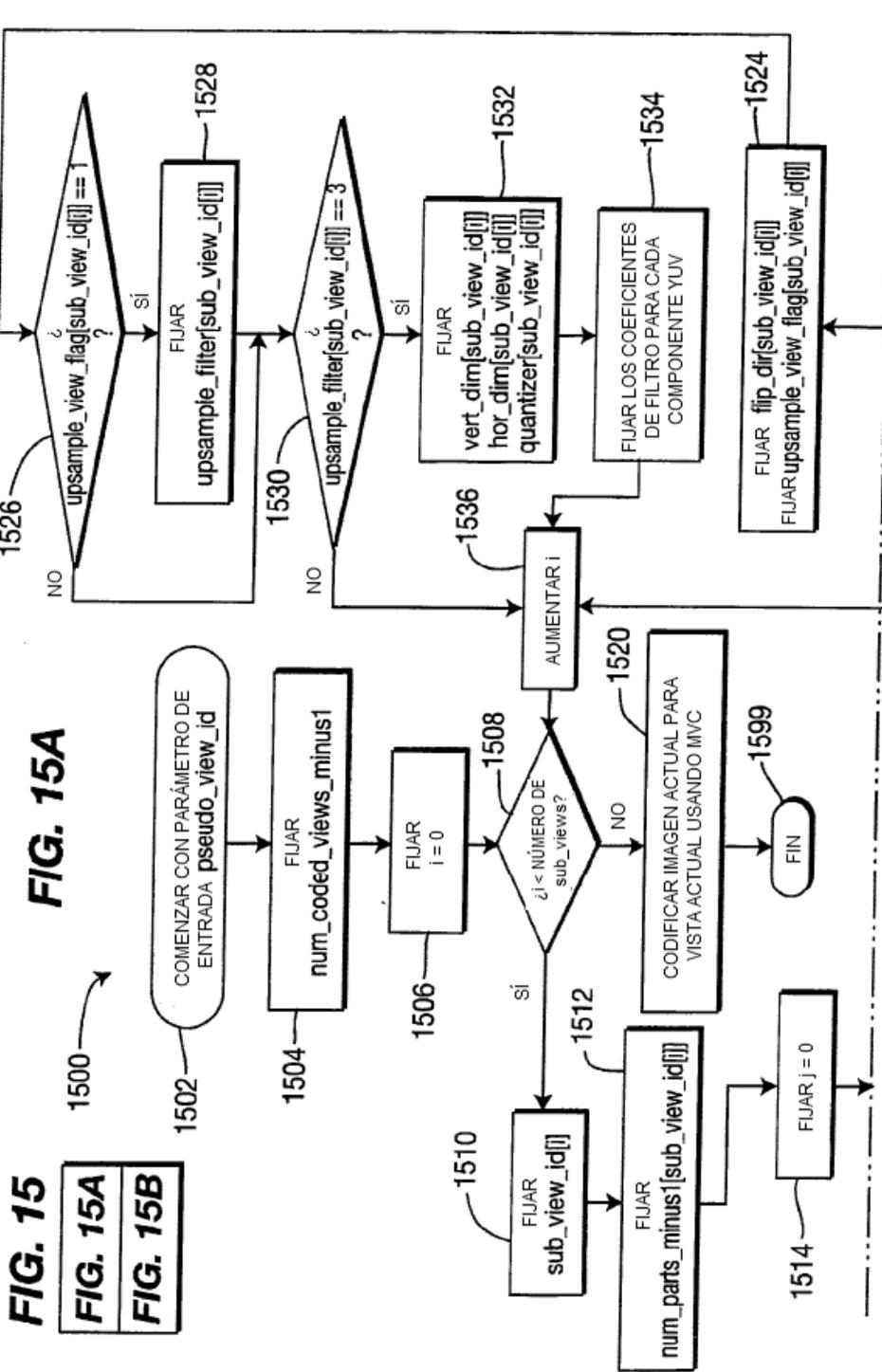


FIG. 14



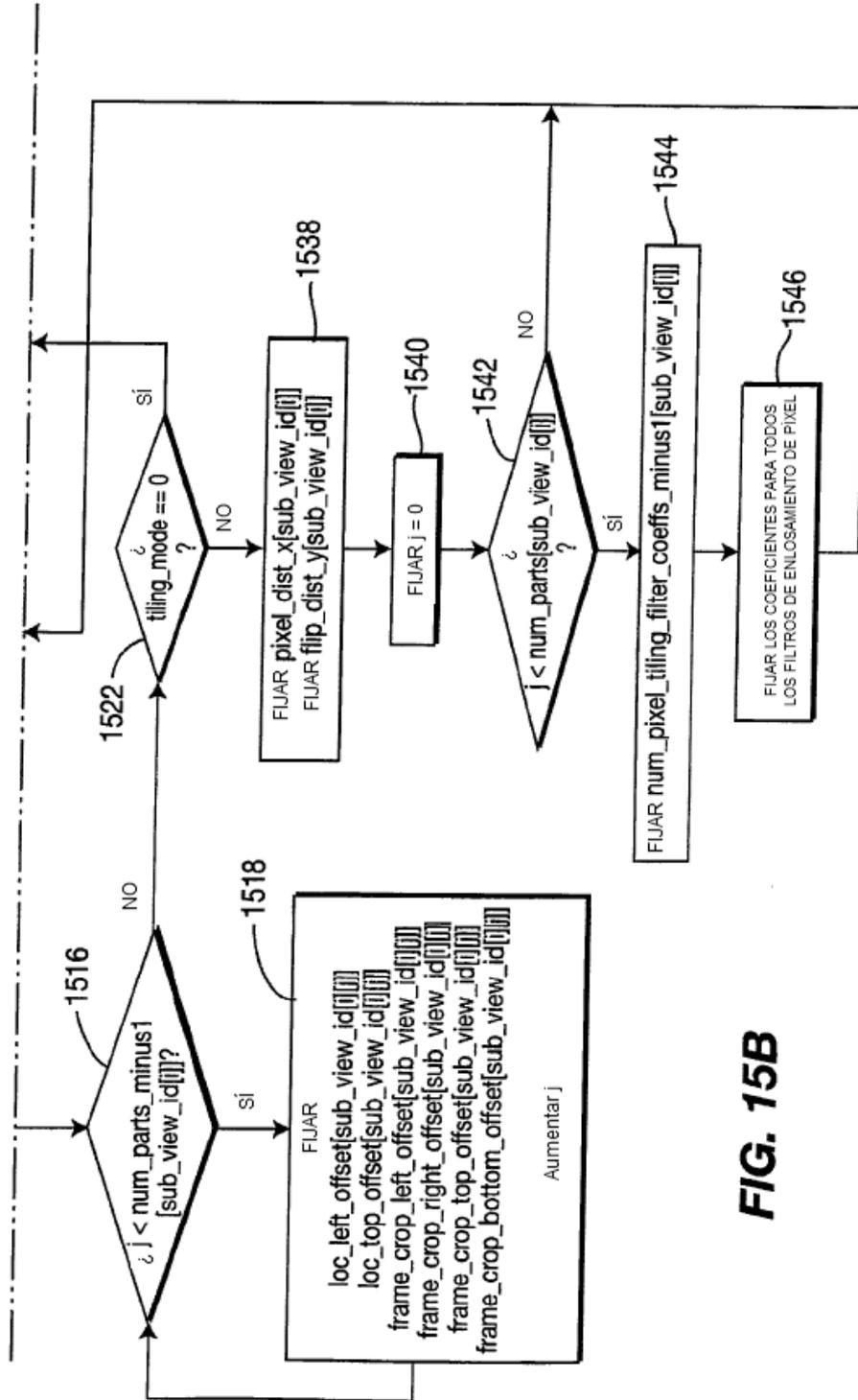


FIG. 15B

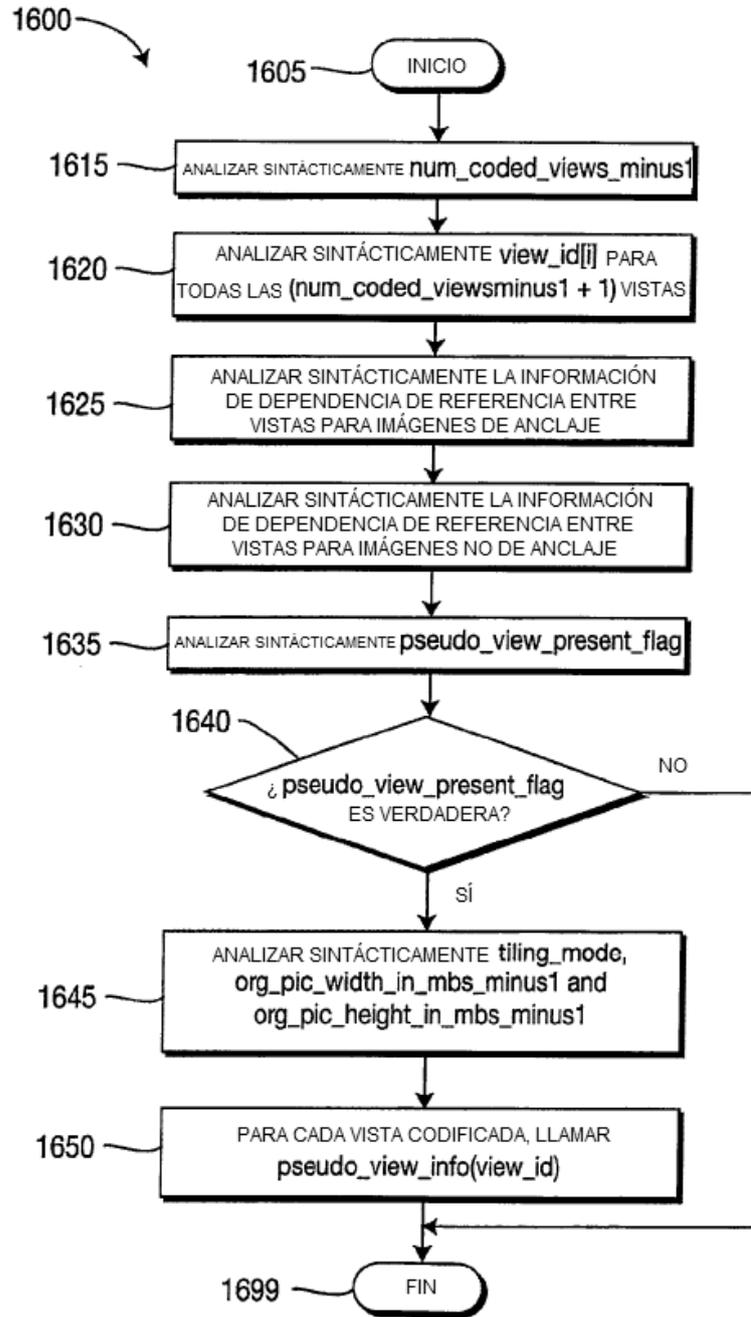
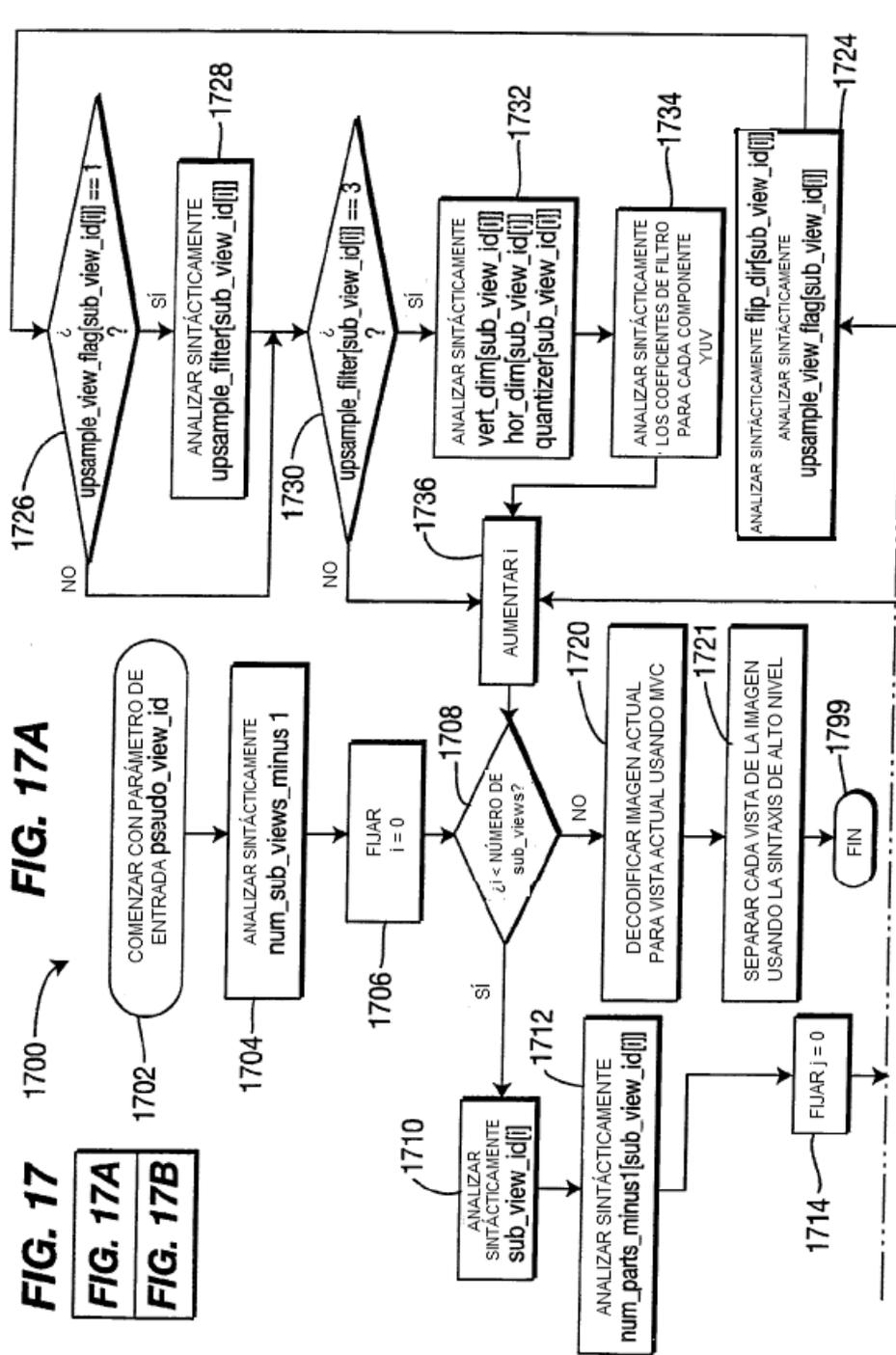


FIG. 16



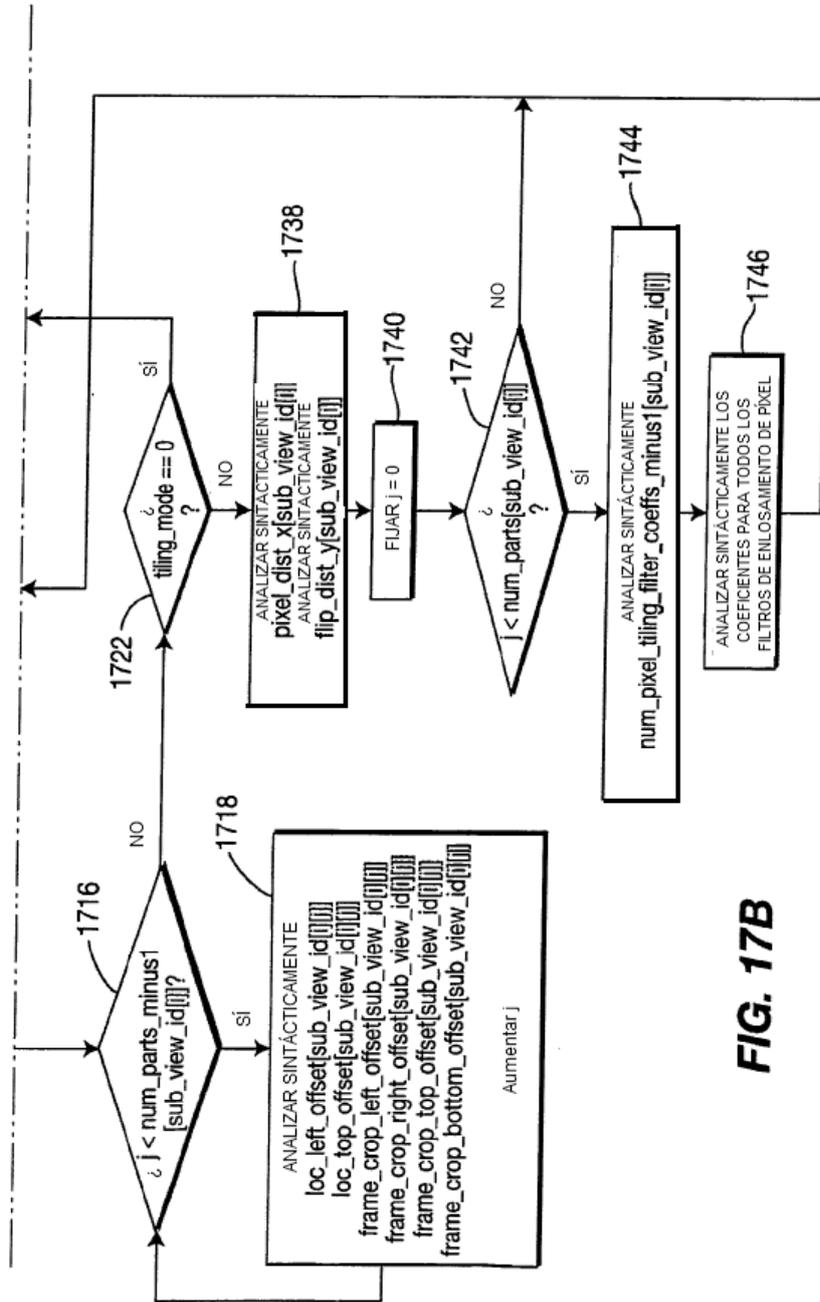
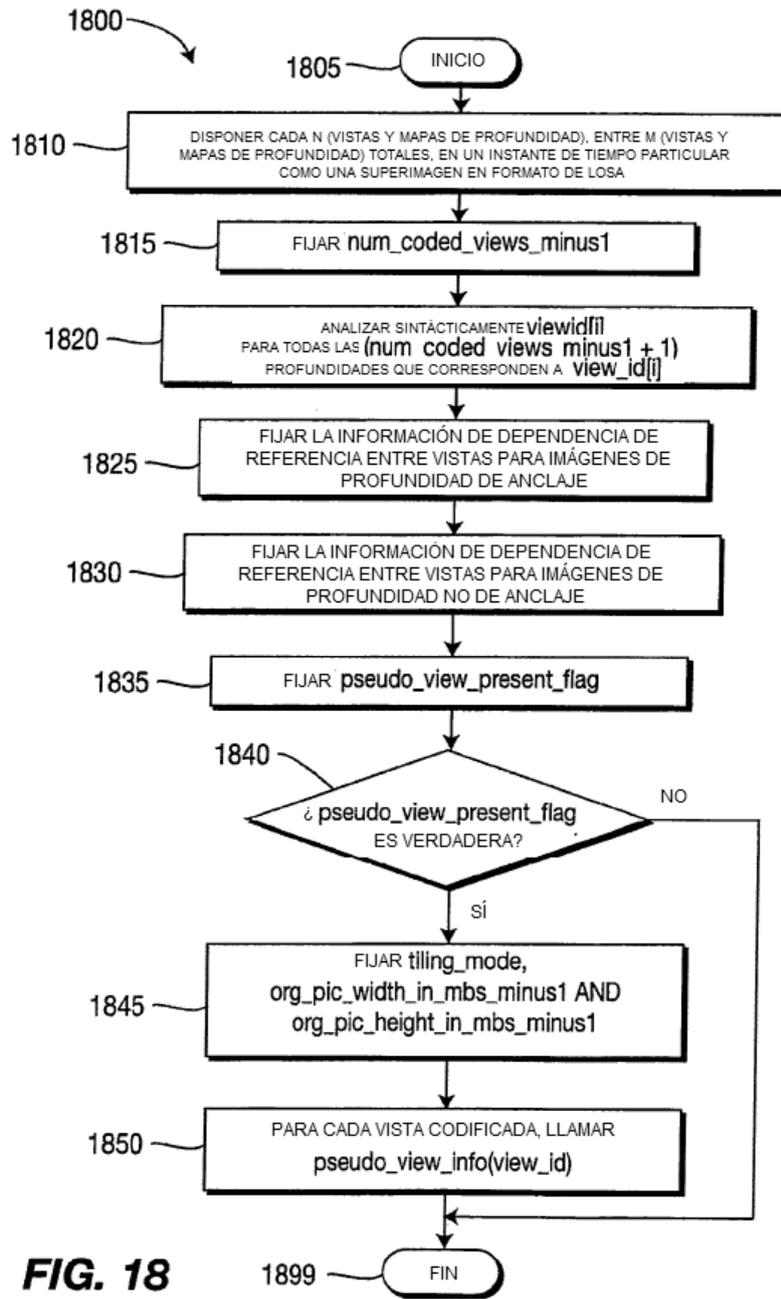
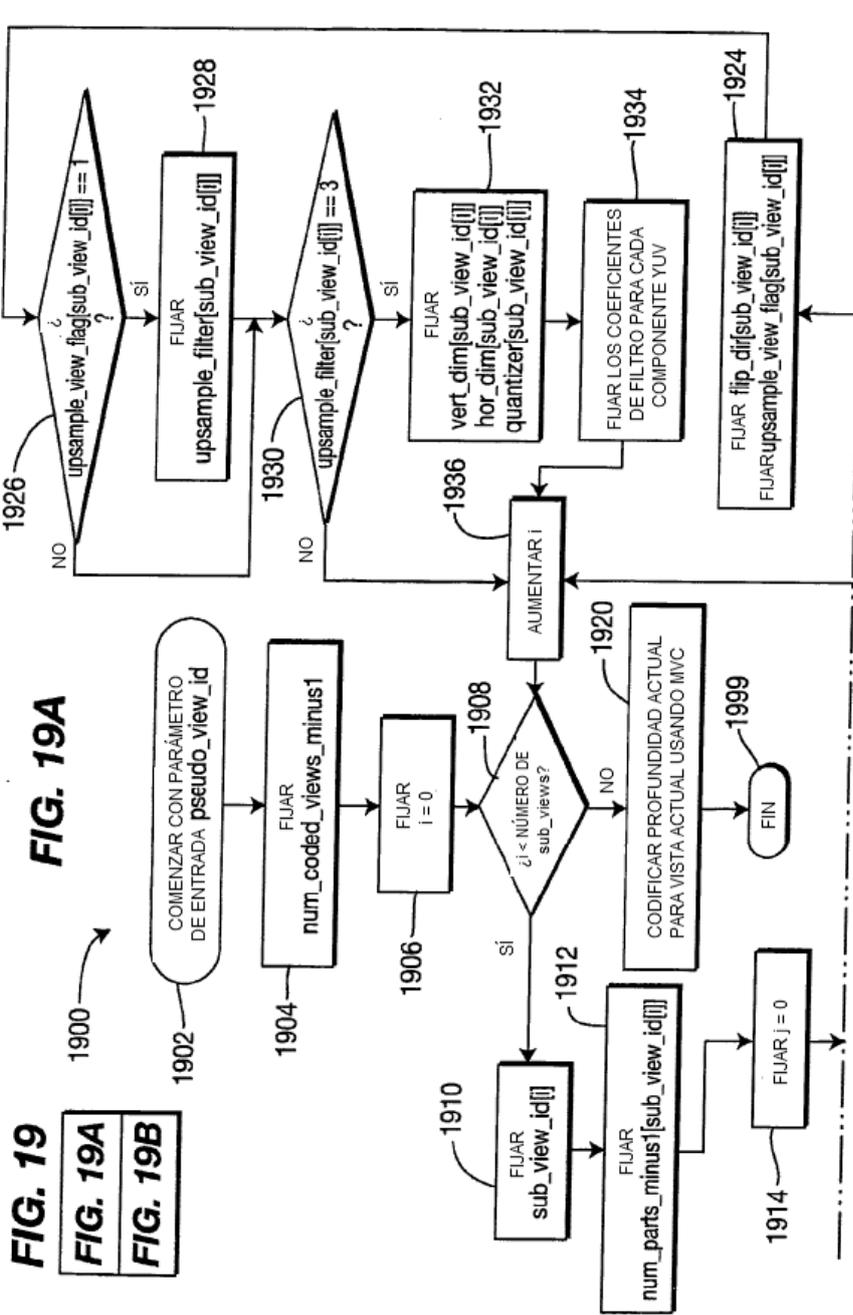


FIG. 17B





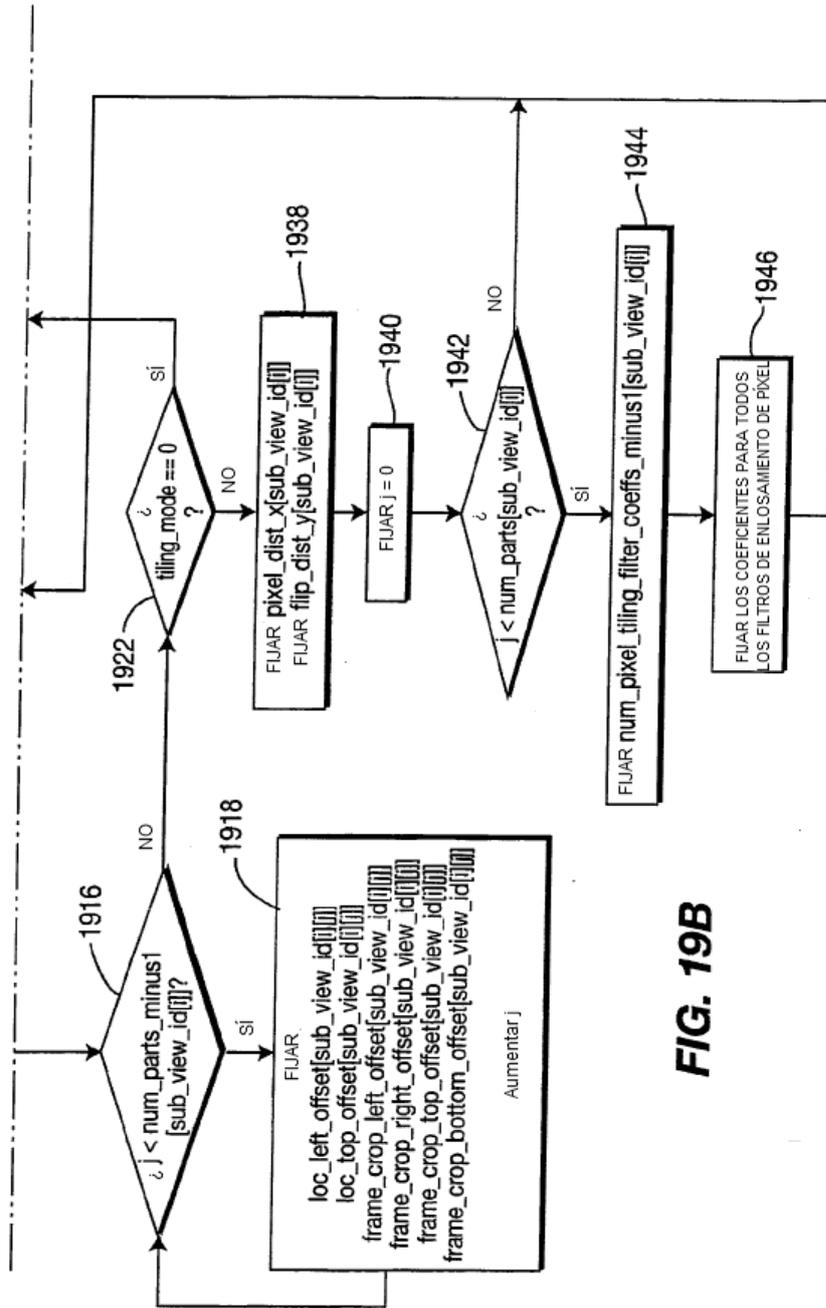


FIG. 19B

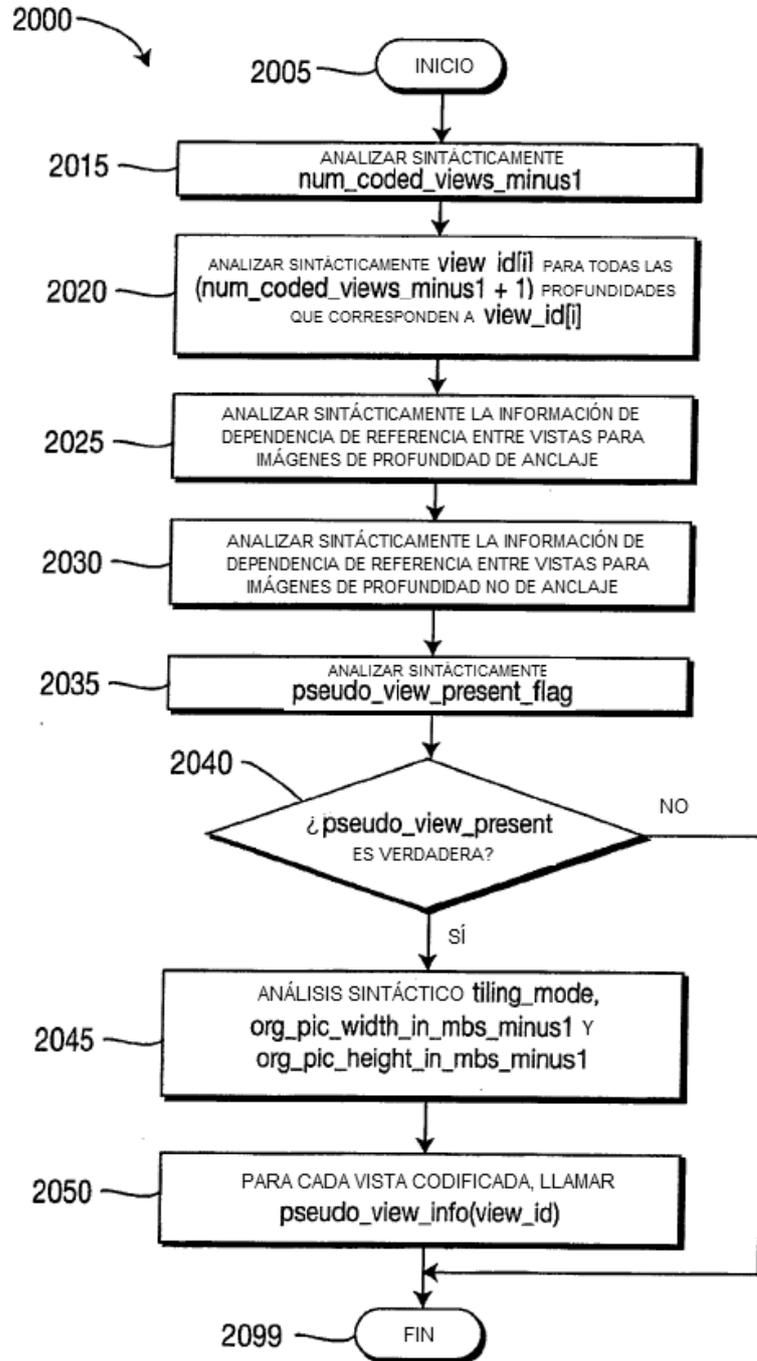
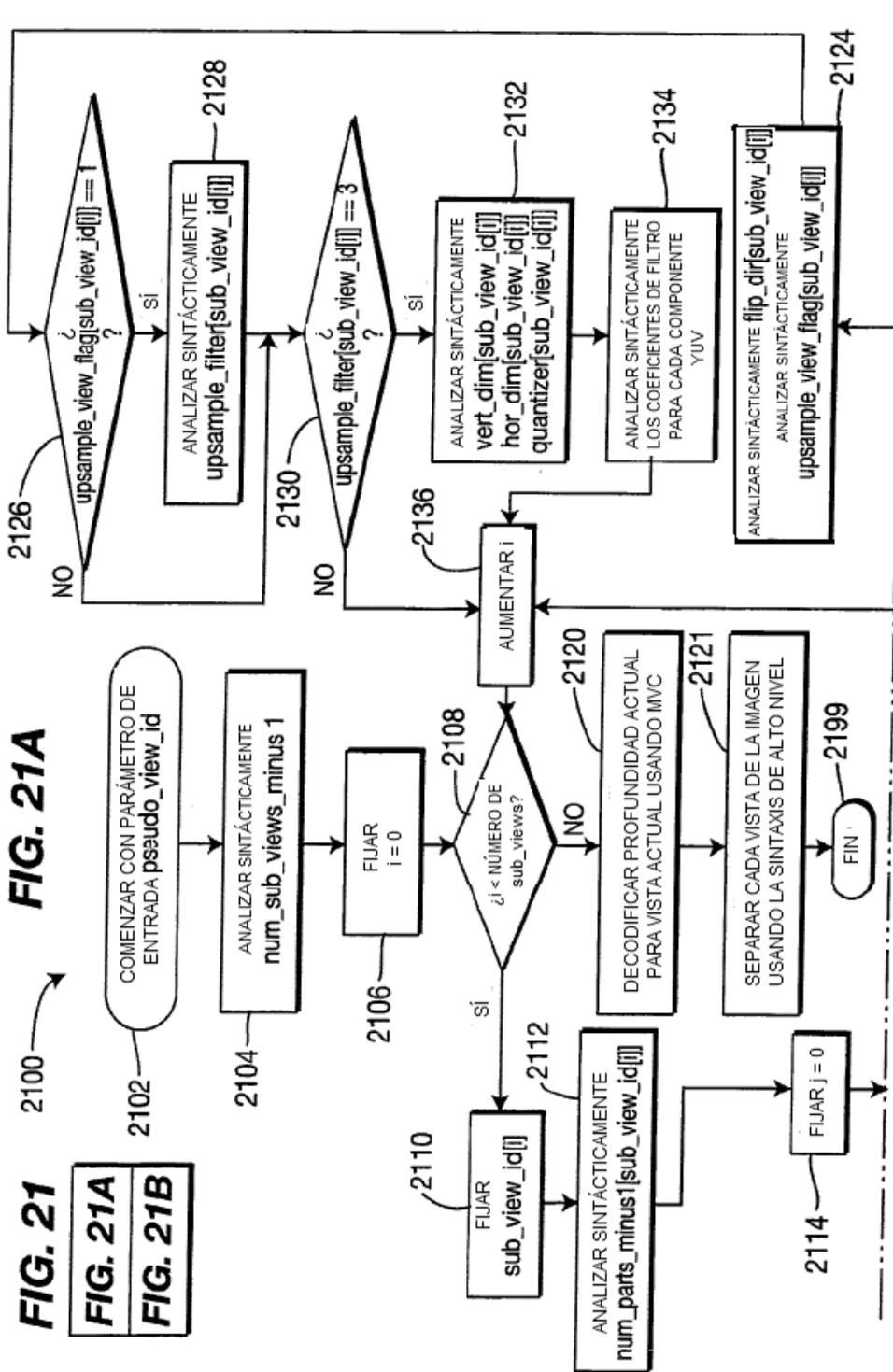


FIG. 20



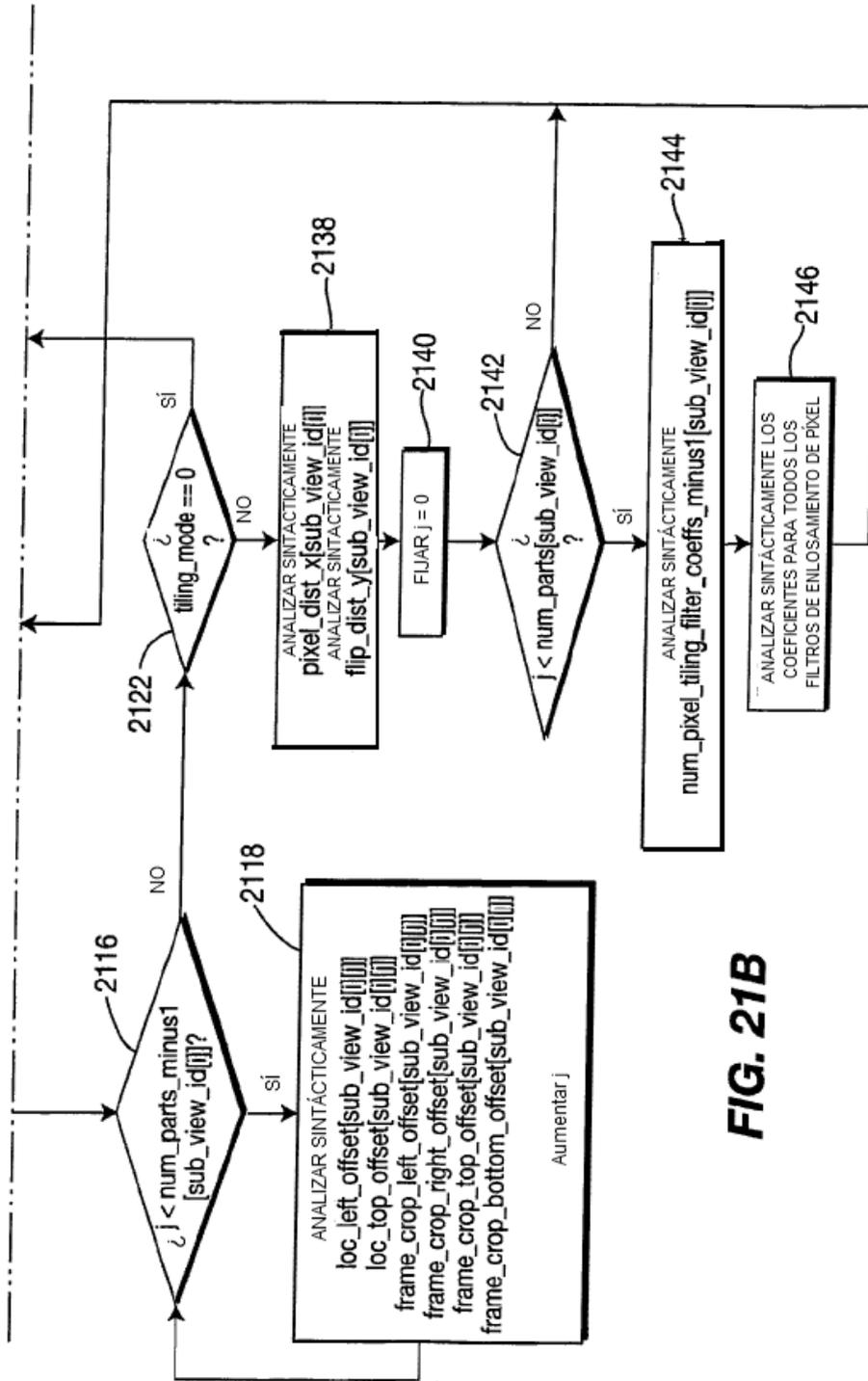


FIG. 21B

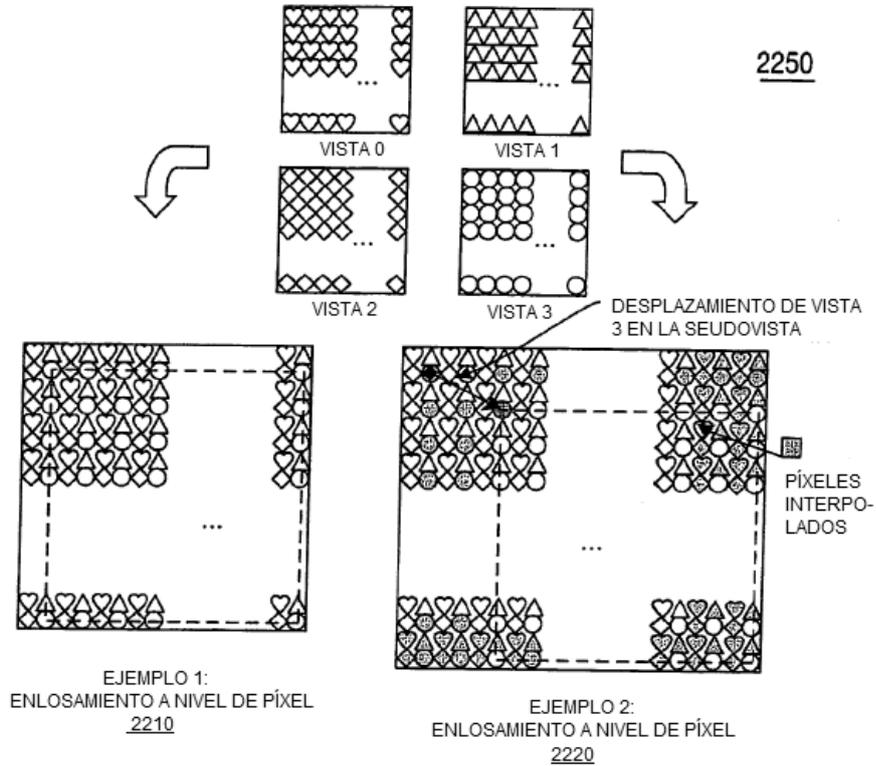


FIG. 22

