

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 606 161**

51 Int. Cl.:

H04N 19/597 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **20.09.2013 PCT/JP2013/076448**

87 Fecha y número de publicación internacional: **03.04.2014 WO14051133**

96 Fecha de presentación y número de la solicitud europea: **20.09.2013 E 13774831 (5)**

97 Fecha y número de publicación de la concesión europea: **02.11.2016 EP 2901695**

54 Título: **Método para procesar uno o más vídeos de una escena en 3D**

30 Prioridad:

**28.09.2012 US 201213631852
16.01.2013 US 201313742461**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
23.03.2017

73 Titular/es:

**MITSUBISHI ELECTRIC CORPORATION (100.0%)
7-3 Marunouchi 2-chome
Chiyoda-ku, Tokyo 100-8310, JP**

72 Inventor/es:

**TIAN, DONG;
ZOU, FENG y
VETRO, ANTHONY**

74 Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

ES 2 606 161 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

Método para procesar uno o más vídeos de una escena en 3D

Campo técnico

5 Esta invención está relacionada generalmente con codificación de vídeos multivisión, y más particularmente con codificación de vídeos multivisión 3D para compresión predictiva. Antecedentes de la técnica

La codificación de vídeo multivisión es esencial para aplicaciones tales como televisión tridimensional (3DTV), televisión de punto de vista libre (FTV) y la vigilancia multicámara. La codificación de vídeo multivisión también se conoce como compresión dinámica de campo de luz. Como se emplea en esta memoria, la codificación puede incluir codificar, decodificar o ambos en un códec, por ejemplo.

10 Las imágenes de profundidad se asumen como parte del formato de datos en un estándar emergente de codificación de vídeo 3D. Usar las imágenes de profundidad como información lateral para realizar codificación predictiva se conoce como predicción por síntesis visual (VSP).

15 En codificación de vídeo convencional, p. ej., codificación según los estándares H.264 AVC (Advanced Video Codificación) y H.265 HEVC (High Rendimiento Video Codificación), se usa información de movimiento de bloques vecinos para derivar un vector de movimiento. El vector de movimiento derivado se usa entonces como predictor de vector de movimiento (MVP) para predecir el vector de movimiento para el bloque actual. Entonces, se codifica y transmite la diferencia de vector de movimiento (MVD) entre el vector de movimiento actual y el MVP.

20 La figura 1 muestra un método convencional para codificar un bloque actual. La etapa 110 deriva un vector de movimiento o de disparidad a partir de un bloque vecino, referido como MotionDerive. La etapa 120 determina un vector de movimiento o de disparidad, referido como MotionCurrent, para el bloque actual por aplicación de técnicas de estimación de movimiento cuya intención es minimizar diferencias residuales. La Etapa 130 calcula y codifica la diferencia de vector de movimiento: ($MotionDiff = MotionCurrent - MotionDerive$). Finalmente, la etapa 140 codifica el bloque residual.

25 La figura 2 muestra el codificador de la técnica anterior correspondiente. El elemento 201 muestra bloques en una parte de una imagen. En el elemento 201, un bloque actual se denota con una estrella "*", y un bloque vecino se denota con un punto ".". A partir los bloques vecinos como se muestra en el elemento 201, derivar 202 un vector de movimiento o vector de disparidad. El vector de movimiento o de disparidad derivado desde 202 sirve como predictor de vector de movimiento (MVP) 203.

30 Con referencia al búfer de imagen de referencia de textura 204, se realiza 205 estimación de movimiento para el bloque actual para producir un vector de movimiento (MotionCurrent) 206 para el bloque actual.

Después de calcular 207 la diferencia entre MVP y MotionCurrent, se obtiene una diferencia de vector de movimiento (MVD) 208, que se codifica 209 hasta el flujo de bits 210.

35 Otra salida de la estimación de movimiento 205 es la imagen de referencia, que sirve como predictor de textura 211. Entonces, se obtiene el residual de textura 213 al realizar 212 predicción de textura sobre la base del predictor de textura 211 y la imagen actual 215.

El residual de textura 213 también se codifica 214 como parte del flujo de bits.

La figura 3 muestra el decodificador. A partir de los bloques vecinos como se muestra en el elemento 301, derivar 302 un vector de movimiento o vector de disparidad. El vector de movimiento o de disparidad derivado sirve como predictor de vector de movimiento (MVP) 303.

40 A partir del flujo de bits 310 codificado, la diferencia de vector de movimiento (MVD) 308 se decodifica 309 y se alimenta a un sumador 307. El predictor de vector de movimiento 303 y la diferencia de vector de movimiento 308 se suman 307, y entonces se obtiene el vector de movimiento usado para el bloque actual MotionCurrent 306.

45 A partir del flujo de bits 310 codificado, se decodifica 314 la imagen de residual de textura 313. El vector de movimiento actual 306 y la imagen de residual de textura son entradas para el módulo de compensación de movimiento 305. Junto con búfer de textura de referencia 304, se realiza la compensación de movimiento, y finalmente se saca la imagen decodificada 315.

Compendio de la invención

50 Un método y un sistema para realizar predicción por síntesis visual (VSP) usando deformación hacia atrás (*backward warping*) para codificación de textura, donde un componente de profundidad se codifica después de un componente de textura.

La predicción por síntesis visual convencional basada en deformación hacia delante típicamente usa los

componentes de textura y de profundidad de una visión de referencia. Aunque la deformación hacia delante normalmente proporciona una alta fidelidad, una importante desventaja de la VSP basada en deformación hacia delante es un significativo aumento en la complejidad del códec.

5 En contraste, la predicción por síntesis visual (B-VSP) basada en deformación hacia atrás, como se describe en esta memoria, es menos compleja. Por lo tanto, la VSP basada en deformación hacia atrás se desea en ciertas aplicaciones. Adicionalmente, la VSP basada en deformación hacia atrás facilita la generación de referencia de VSP basada en bloque. El documento de Najunda Iyer K et al: "Multiview video coding using depth based 3D warping", ICME 2010, IEEE, describe VSP basada en deformación hacia atrás.

10 Desafortunadamente, la VSP basada en deformación hacia atrás normalmente requiere primera codificación en profundidad. Esto es, el componente de profundidad se codifica antes que el componente de textura correspondiente de la misma unidad de acceso. Una unidad de acceso es una colección de todos los componentes (textura y profundidad) en un mismo instante de tiempo. La primera codificación en profundidad impide el uso de herramientas de codificación en profundidad que necesitan acceder al correspondiente componente de textura.

15 Para explotar las ventajas de la VSP basada en deformación hacia atrás, al tiempo que se evita la primera codificación en profundidad, las realizaciones usan un vector de movimiento o de disparidad de bloques vecinos para derivar un bloque de profundidad de imágenes de profundidad previamente codificadas, tales como las imágenes de profundidad codificadas en la unidad de acceso anterior, o la misma unidad de acceso pero desde otros puntos de vista. Por tanto, se asume el bloque de profundidad derivado como un bloque de profundidad estimado para el bloque actual a codificar, que finalmente se usa para VSP basado en deformación hacia atrás.

20 Efecto de la invención

La predicción por síntesis visual convencional basada en deformación hacia delante típicamente usa los componentes de textura y de profundidad de una visión de referencia. La deformación hacia delante normalmente proporciona una alta fidelidad con un mayor coste. En contraste, nuestra predicción por síntesis visual usa deformación hacia atrás para codificación de textura, y no requiere que el componente de profundidad sea codificado antes que el componente de textura.

Adicionalmente, la VSP basada en deformación hacia atrás facilita la generación de referencia de VSP basada en bloque, que es menos compleja y tiene ventajas de implementación.

30 Para permitir la VSP basada en deformación hacia atrás, al tiempo que se evita la primera codificación en profundidad, la invención usa un vector de movimiento o de disparidad de bloques vecinos para derivar un bloque de profundidad de imágenes de profundidad previamente codificadas, tales como las imágenes de profundidad codificadas en la unidad de acceso anterior, o la misma unidad de acceso pero desde otros puntos de vista.

Breve descripción de los dibujos

La figura 1 es un diagrama de flujo de un método de la técnica anterior que usa información de movimiento vecino para codificar información de movimiento de un bloque actual;

35 La figura 2 es un diagrama de bloques de un codificador de la técnica anterior;

La figura 3 es un diagrama de bloques de un decodificador de la técnica anterior;

La figura 4 es un diagrama de flujo de un método que usa información de movimiento o de disparidad vecinos para codificar un bloque actual con predicción VSP hacia atrás según realizaciones de la invención;

40 La figura 5A es un diagrama de bloques de un codificador que usa predicción espacial según realizaciones de la invención;

La figura 5B es un diagrama de bloques de un codificador que usa predicción temporal según realizaciones de la invención;

La figura 6A es un diagrama de bloques de un decodificador que usa predicción espacial según realizaciones de la invención;

45 La figura 6B es un diagrama de bloques de un decodificador que usa predicción temporal según realizaciones de la invención;

La figura 7 es un diagrama de flujo de un método de codificación según realizaciones de la invención;

La figura 8 es un diagrama de flujo de un método de decodificación según realizaciones de la invención;

50 La figura 9 es un ejemplo para estimar los valores de profundidad a partir de bloques vecinos según realizaciones de la invención;

La figura 10 es otro ejemplo para estimar los valores de profundidad a partir de bloques vecinos según realizaciones de la invención; y

La figura 11 es un diagrama de flujo que muestra el proceso de decodificación de un bloque en una imagen de anclaje de profundidad que usa el modo B-VSP según realizaciones de la invención.

5 Descripción de realizaciones

Las realizaciones de nuestra invención proporcionan un método y un sistema para generar una imagen de referencia de predicción por síntesis visual (VSP) sobre la base de una técnica de deformación hacia atrás. El método no usa primera codificación en profundidad. El método usa información de movimiento o de disparidad de bloques vecinos a un bloque actual para derivar un bloque de profundidad para el bloque actual a codificar. Entonces, el bloque de profundidad derivado se usa para realizar deformación hacia atrás. Como se emplea en esta memoria, la síntesis visual genera un video sintético a partir de uno o más videos de una escena, en la que cada video de una escena 3D es adquirido por una cámara correspondiente dispuesta en una pose particular, y en el que una visión de cada cámara se superpone con la visión de al menos otra cámara.

Comparado con la síntesis visual convencional que usa la información de movimiento de bloques vecinos, las realizaciones se caracterizan por derivar un bloque de profundidad y formar un bloque de muestra para predecir el bloque actual sobre la base de un bloque de profundidad derivado.

Las realizaciones describen diferentes maneras de usar la información de movimiento o de disparidad de los bloques vecinos. Específicamente, la información de movimiento o de disparidad derivada se usa directamente para derivar un bloque de profundidad, DepthDerive, que se usa además para formar un predictor para los valores de muestra en el bloque actual.

La figura 4 muestra un método para codificar según realizaciones de la invención. Las etapas se realizan para cada bloque actual 401.

Etapas 410, obtener vector de movimiento y de disparidad para bloques vecinos

Etapas 420, determinar un bloque de profundidad sobre la base de una imagen de profundidad de referencia correspondiente y el vector de movimiento o de disparidad

Etapas 430, generar un bloque de predicción sobre la base de los valores de profundidad usando deformación hacia atrás

Etapas 440, realizar codificación predictiva para el bloque actual usando el bloque de predicción

La figura 5A un diagrama de bloques de codificador comparable. A partir de los bloques vecinos mostrados en el elemento 501, se deriva 502 un vector de movimiento o vector de disparidad MotionDerive (MVP) 503. Opcionalmente, se puede codificar y señalar 550 MotionDerive o información lateral que ayudaría al decodificador a determinar MotionDerive, y es parte del flujo de bits codificado 510.

Con referencia al búfer de imagen de referencia de profundidad 551 que usa MotionDerive, se estima 504 un bloque de profundidad DepthDerive 505 para el bloque actual.

Con referencia al búfer de imagen de referencia de textura 504 que usa DepthDerive, se realiza 506 una deformación hacia atrás para obtener un predictor de textura 507.

Entonces, la predicción de textura 508 obtiene el residual de textura 509 sobre la base del predictor de textura 507 y la imagen actual 515, que se codifica 511 hasta el flujo de bits.

La figura 6A muestra el decodificador. A partir del flujo de bits 610 codificado, se puede señalar 650 información lateral 653 para derivar selección de vector de movimiento o de disparidad. Si se está usando la información lateral 653, se alimenta al bloque 601 como una entrada.

A partir de los bloques vecinos como se muestra en el elemento 601, derivar 602 un vector de movimiento o vector de disparidad MotionDerive 603.

Con referencia al búfer de imagen de referencia de profundidad 651 que usa MotionDerive, se estima 604 un bloque de profundidad DepthDerive 605 para el bloque actual.

Con referencia al búfer de imagen de referencia de textura 611 que usa DepthDerive 605, se realiza 606 una deformación hacia atrás 612 para obtener un predictor de textura 607.

A partir del flujo de bits codificado, se decodifica 613 el residual de textura 609 para obtener el predictor de textura 607 y el residual de textura 609 para predicción de textura 608 para sacar la imagen de textura decodificada 615.

La técnica usada por las realizaciones se puede referir como predicción compensada en profundidad, a diferencia de predicción compensada en movimiento o disparidad, como en la técnica anterior.

Se describen tres realizaciones que usan diferentes imágenes y profundidades disponibles de una imagen de referencia.

5 Realización 1: está disponible imagen de anclaje, profundidad de visiones de referencia

En esta realización, se asume que componentes de textura y de profundidad de una imagen de referencia se codifican ambos antes que la textura y profundidad actuales. P. ej., el orden de codificación para el caso de dos visiones es T0D0T1D1, donde T y D representan la textura y profundidad, y 0 y 1 representan dos visiones. Nota, la textura y profundidad de la visión 0 se codifican antes que la textura y profundidad para la visión 1.

10 Para imágenes de anclaje, todos los bloques se codifican ya sea en modos Intra o en modos de predicción Inter-visión. En un ejemplo, hay tres bloques vecinos para el bloque actual, que son izquierdo (A), superior (B) y superior-derecho (C). La descripción se puede expandir fácilmente para usar bloques vecinos adicionales.

Si los bloques vecinos (A, B y C) se codifican en modos de predicción Inter-visión, entonces cada bloque se asocia con un vector de disparidad, denotado por DisparityA, DisparityB y DisparityC.

15 Considerando la similitud entre el bloque actual y los bloques vecinos, un vector de disparidad para el bloque actual se puede derivar a partir de los bloques vecinos, denotado por DisparityDerive. En una implementación,

DisparityDerive = Media (Disparity, DisparityB, DisparityC).

20 En otra implementación, la derivación se puede señalar para que mezcle una lista de candidatos e índice como se define en el estándar emergente HEVC. Esto es, se construye una lista de candidatos a partir de los bloques vecinos. Entonces, se señala un índice en el flujo de bits para que indique el candidato específico usado para DisparityDerive.

Con DisparityDerive, se puede identificar y acceder a un bloque de profundidad DepthDerive en la imagen de profundidad de la visión de referencia. Se usa DepthDerive para hacer la deformación hacia atrás para el bloque actual.

25 El proceso general usa los bloques vecinos como se ha descrito anteriormente cuando los bloques vecinos se codifican usando modos de predicción Inter-visión. A continuación, se describen los casos cuando un bloque vecino no se codifica en modos de predicción Inter-visión.

Si el bloque vecino se codifica en modo Intra, entonces el vector de disparidad correspondiente es cero.

30 Si el bloque vecino se codifica usando modos de predicción VSP, entonces no se usa un único vector para el bloque entero. En cambio, en ese caso, se puede determinar un vector de disparidad a partir del bloque de profundidad usado para su predicción VSP. El vector de disparidad puede ser el valor promedio del bloque de profundidad entero, o dando más peso a los píxeles de profundidad que están más cerca del bloque actual.

35 Antes, se deriva primero el vector de disparidad DisparityDerive y entonces se usa una única disparidad para acceder al bloque de profundidad DepthDerive. Como alternativa, se pueden usar múltiples disparidades derivadas, y así, múltiples bloques de profundidad derivados. Los múltiples bloques de profundidad derivados se determinan según un promedio ponderado para obtener un único bloque de profundidad a usar para la deformación hacia atrás.

La figura 7 muestra el proceso de codificación para un bloque en una imagen de anclaje usando la B-VSP como modo de codificación de candidato.

40 Etapa 701, Dejar que A, B y C sean los bloques vecinos, DisparityX sea su vector de disparidad (con X sustituido por A, B o C)

Etapa 702, Si algún bloque en A, B y C se codifica en modo Intra, establecer su vector de disparidad a cero

Etapa 703, Si algún bloque en A, B y C se codifica en modos de predicción VSP, entonces establecer su vector de disparidad como el valor medio del bloque de profundidad usado para su predicción VSP

Etapa 704, Establecer DisparityDerive para que sea la media del DisparityX

45 Etapa 705, Obtener el bloque de profundidad en la imagen de profundidad correspondiente, DepthDerive

Etapa 706, Usar DepthDerive para realizar VSP basada en deformación hacia atrás

Etapa 707, El modo B-VSP se compara con los otros modos de codificación y se selecciona si el coste de tasa de distorsión es menos.

ES 2 606 161 T3

La figura 8 es el proceso de decodificación de un bloque en una imagen de anclaje que usa el modo B-VSP propuesto.

Etapa 801, Comprobar si se señala el modo B-VSP para el bloque actual. En caso afirmativo, ir a la Etapa 802. De lo contrario, ir a la Etapa 808.

- 5 Etapa 802, Dejar que A, B y C sean los bloques vecinos, DisparityX sea su vector de disparidad (con X sustituido por A, B o C)

Etapa 803, Si algún bloque en A, B y C se codifica en modo Intra, entonces establecer el vector de disparidad a cero

Etapa 804, Si algún bloque en A, B y C se codifica en modos de predicción VSP, entonces establecer su vector de disparidad como el valor medio del bloque de profundidad usado para su predicción VSP

- 10 Etapa 805, Establecer DisparityDerive para que sea la media de DisparityX

Etapa 806, Obtener el bloque de profundidad DepthDerive en la imagen de profundidad correspondiente.

Etapa 807, Usar DepthDerive para realizar VSP basada en deformación hacia atrás Ir a la Etapa 809.

Etapa 808, Decodificar el bloque actual en otros modos de codificación

Etapa 809, Fin del procedimiento.

- 15 Realización 2: está disponible imagen de anclaje o de no anclaje, profundidad de visiones de referencia

En esta realización, se asume que todos los componentes de textura en una unidad de acceso se codifican antes que los componentes de profundidad. P. ej., el orden de codificación para el caso de dos visiones es T0T1D0D1, donde la textura de visión 0 y visión 1 se codifican antes que la profundidad. Esta realización también se puede aplicar cuando no hay dependencia de la textura en la codificación de la profundidad.

- 20 En un proceso similar, la disparidad DisparityDerive se puede derivar de los bloques vecinos. Sin embargo, como los datos de profundidad no están disponibles como referencia, no se puede acceder a un bloque de profundidad por píxel. En cambio, se interpola un bloque de profundidad usando las disparidades DisparityA, DisparityB y DisparityC. En esta realización, se está usando una interpolación lineal. También se podrían usar otras interpolaciones.

- 25 La figura 9 muestra un ejemplo de interpolación, en la que se establece un único valor de profundidad (X, Y y Z) para un bloque vecino (A, B y C) a partir de su valor de disparidad DisparityA, DisparityB y DisparityC. Los valores de profundidad para el bloque actual 4x4 se interpolan de la siguiente manera:

$$a = (X+Y) / 2;$$

$$b = (X + 2*Y) / 3;$$

$$c = (2*X + Y) / 3;$$

- 30 $d = (X + 3*Y) / 4;$

$$e = (3*X + Y) / 4;$$

$$f = (X + 4*Y) / 5; y$$

$$g = (4*X + Y) / 5,$$

- 35 donde a, b, c, d, e, f, y g son píxeles en el bloque actual. En la implementación anterior, no se usa el valor de profundidad Z del bloque C. Al valor de profundidad que está más cerca del píxel a interpolar se le da más peso. Son posibles otros métodos de interpolación para llenar el bloque de profundidad.

Adicionalmente, el valor de profundidad de un bloque vecino puede variar cuando el bloque se codifica usando modos B-VSP. En ese caso, el proceso de interpolación para cada píxel puede ser diferente.

- 40 En un ejemplo, como se muestra en la figura 10, el valor de profundidad de un píxel en la fila r y la columna c, (r, c) se puede interpolar como

$$\text{Profundidad}(r, c) = (Xc + Yr) / 2.$$

Realización 3: está disponible imagen no anclaje, profundidad de visiones de referencia

Para imágenes no anclaje, es más probable que los bloques vecinos usen predicción temporal, distinta a modos de predicción inter-visión. Sin embargo, no es necesario diferenciar predicción temporal y predicción inter-visión. En

cambio, en esta realización, la textura de referencia asociada del vector de movimiento o de disparidad tiene disponible una imagen de profundidad de referencia correspondiente. De otro modo, se aplica el caso que se describe para la Realización 2.

5 Como en la Realización 1, los vectores de movimiento o de disparidad de bloques vecinos se usan para derivar un vector de movimiento o de disparidad, que se usa para identificar y acceder a un bloque de profundidad de la imagen de profundidad de referencia correspondiente.

10 Si un bloque vecino se codifica en moda Intra, entonces se usa un vector de disparidad cero respecto a la referencia inter-visión, en lugar de un vector de movimiento cero respecto a la referencia temporal, cuando no hay mecanismo en el flujo de bits para señalar al bloque vecino a usar para derivar el vector de movimiento o de disparidad. De otro modo, cuando hay una lista de candidatos, se implementa un mecanismo de índice. Se añade un vector de disparidad cero y un vector de movimiento cero a la lista de candidatos cuando hay un bloque vecino codificado con Intra. Entonces, el codificador selecciona el candidato óptimo sobre la base del coste de distorsión de tasa.

Realización 4: VSP basada en deformación hacia atrás para imagen de profundidad

15 Las realizaciones anteriores también son aplicables para codificar la imagen de profundidad de una visión dependiente. Es imposible implementar una predicción VSP basada en deformación hacia atrás para profundidad en la VSP convencional porque se requiere que la imagen de profundidad haga deformación hacia atrás. Nuestras realizaciones no requieren el bloque actual.

20 A diferencia de las realizaciones anteriores, en las que los vectores de movimiento o de disparidad de los bloques vecinos se utilizan para que un bloque de profundidad haga VSP hacia atrás, se usan los valores de muestra del bloque vecino para estimar un bloque de profundidad. Como la imagen actual es una imagen de profundidad, los bloques vecinos tienen profundidades por píxel. Es más preciso usar los valores de muestra, en lugar del vector de movimiento o de disparidad a nivel de bloques de esos bloques para determinar el bloque de profundidad para que haga B-VSP.

25 La figura 11 muestra la correspondiente decodificación. La diferencia con la realización mostrada en la figura 8 es que no se usan los vectores de movimiento o de disparidad de bloques vecinos.

Etapa 1101, Comprobar si se señala el modo B-VSP para el bloque actual. En caso afirmativo, ir a la Etapa 1102. De lo contrario, ir a la Etapa 1104.

Etapa 1102, Usar el bloque vecino (que son bloques de profundidad) para estimar la profundidad DepthDerive del bloque actual.

30 Etapa 1103, Usar DepthDerive para realizar VSP basada en deformación hacia atrás Ir a la Etapa 1105.

Etapa 1104, Decodificar el bloque actual en otros modos de codificación

Etapa 1105, Fin del procedimiento.

Realización 5. deformación en dominio de movimiento

35 En las realizaciones anteriores, la deformación se realizó en el dominio espacial o de píxel. En otras palabras, el predictor para la textura de los píxeles en la visión actual se deriva por deformación de los valores de píxel de la visión de referencia a la visión actual, que se hace en los bloques 506 y 611 de la figura 5A un y figura 6A, respectivamente.

Sin embargo, también existe una relación de deformación 3D en el dominio temporal. Esto es, el movimiento temporal para un objeto en la visión actual parece tener movimiento similar en la visión de referencia.

40 Así, en esta realización, se aplica la deformación en el dominio espacial y en el dominio temporal. Por lo tanto, se modifica el procesamiento en los bloques 506 y 611 como se muestra en la figura 5B y figura 6B, el campo de movimiento (vector de movimiento, y/o información de imagen de referencia) se asigna a los píxeles correspondientes en la imagen actual. Además, el bloque 508 y 608 realizan la predicción temporal usando la información de movimiento de campo deformado.

45 Obsérvese que la deformación en el campo de movimiento se puede realizar a nivel de píxel, o a nivel de bloque, de manera similar al proceso de deformación de textura. En el contexto de HEVC, la unidad de deformación más grande está en el nivel de unidad de predicción (PU).

50 Si la deformación del campo de movimiento no se realiza a nivel de píxel, una profundidad máxima del bloque de profundidad derivado se convierte en el vector de disparidad, y finalmente el vector de disparidad derivado se asume como una compensación de traslación para que el bloque actual acceda a la información de movimiento.

Cabe señalar que la derivación de vector de disparidad por referencia a un bloque de profundidad en imagen de

referencia de profundidad se puede extender además a otros bloques vecinos regulares espaciales o temporales, siempre que desde la imagen de profundidad se pueda acceder a un bloque de profundidad.

Por tanto, la deformación puede estar en el dominio espacial como se describe en la Solicitud relacionada, el dominio temporal como se describe en esta Solicitud, o en ambos dominios espacial y temporal.

5 **Efecto de la invención**

La predicción por síntesis visual convencional basada en deformación hacia delante típicamente usa los componentes de textura y de profundidad de una visión de referencia. La deformación hacia delante normalmente proporciona una alta fidelidad con un mayor coste. En contraste, nuestra predicción por síntesis visual usa deformación hacia atrás para codificación de textura, y no requiere que el componente de profundidad sea codificado antes que el componente de textura. Adicionalmente, la VSP basada en deformación hacia atrás facilita la generación de referencia de VSP basada en bloque, que es menos compleja y tiene ventajas de implementación.

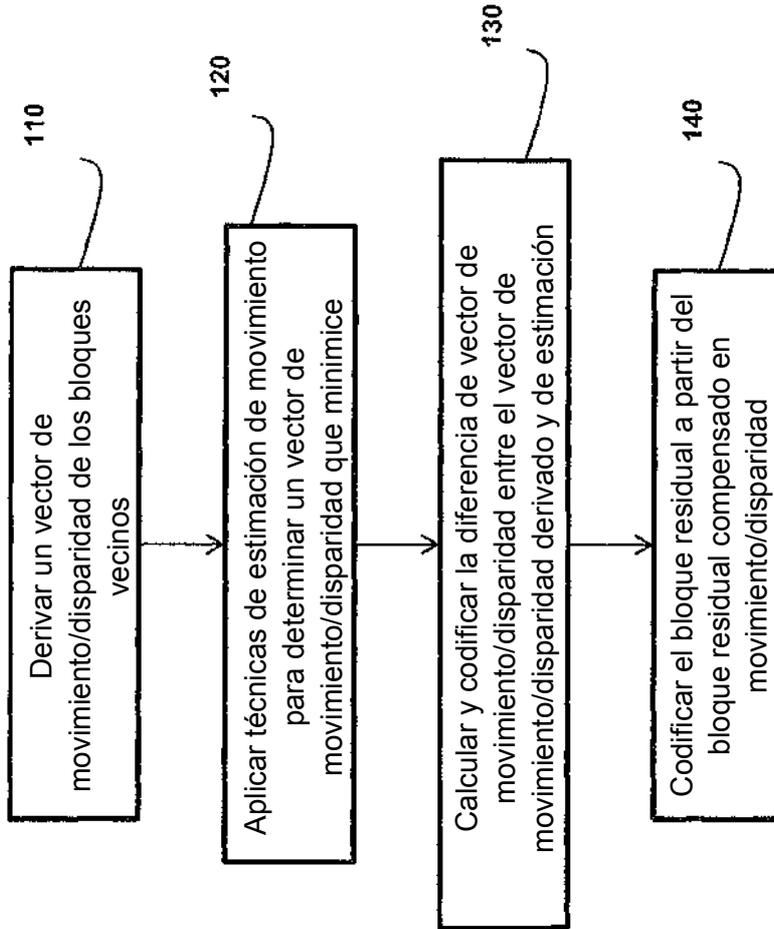
Para permitir la VSP basada en deformación hacia atrás, al tiempo que se evita la primera codificación en profundidad, la invención usa un vector de movimiento o de disparidad de bloques vecinos para derivar un bloque de profundidad de imágenes de profundidad previamente codificadas, tales como las imágenes de profundidad codificadas en la unidad de acceso anterior, o la misma unidad de acceso pero desde otros puntos de vista.

REIVINDICACIONES

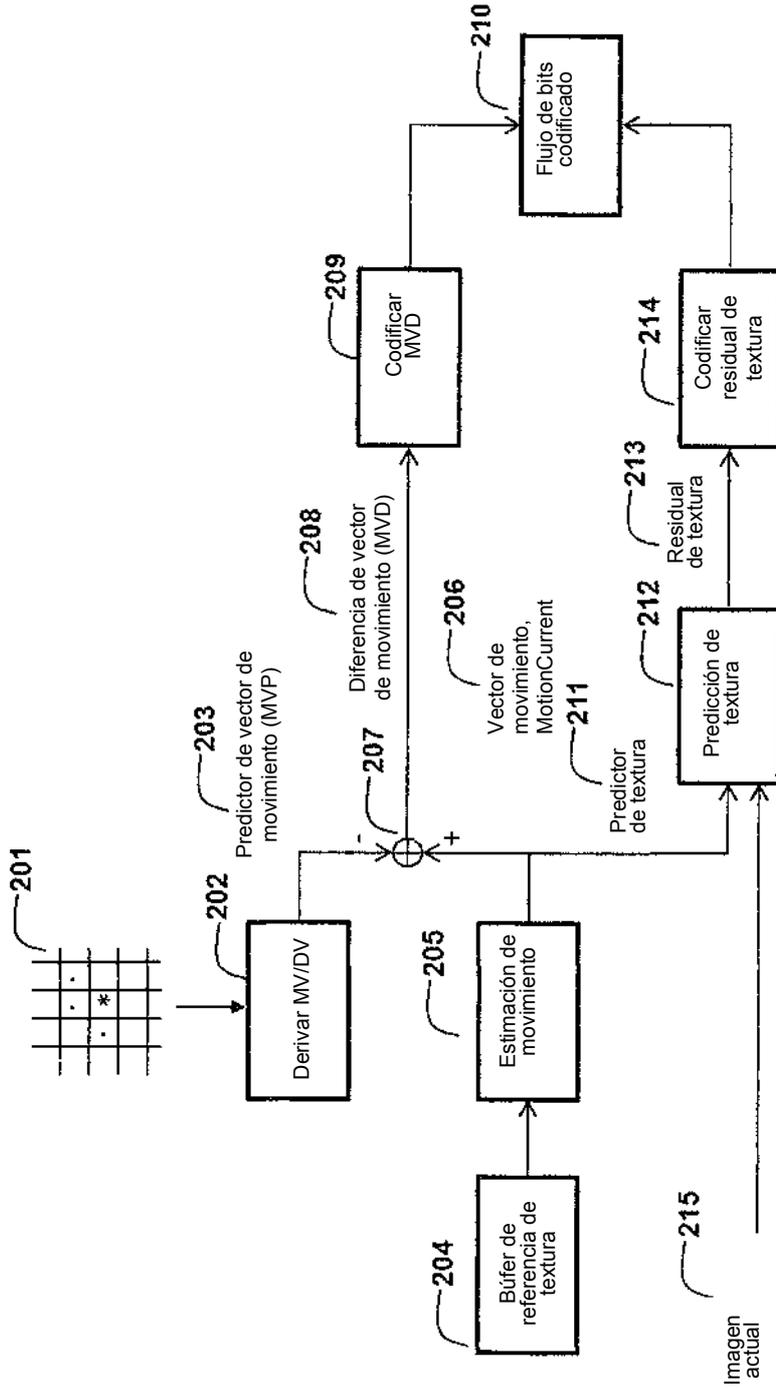
1. Un método para procesar uno o más videos de una escena en 3D, en el que cada video es adquirido por una cámara correspondiente dispuesta en una pose particular, que comprende para cada bloque actual en una imagen actual (515) a procesar las etapas de:
- 5 obtener (410) un vector de movimiento o de disparidad (503) a partir de bloques vecinos (501) del bloque actual;
- determinar (420) un bloque de profundidad (505) para el bloque actual a partir de un componente de profundidad correspondiente de una imagen de referencia sobre la base del vector de movimiento o de disparidad (503) obtenido a partir de los bloques vecinos (501) del bloque actual;
- 10 sintetizar (430), usando distorsión hacia atrás (506), un campo de movimiento correspondiente a una visión actual sobre la base de una imagen de referencia correspondiente a una visión de referencia y el bloque de profundidad determinado (505); y
- codificar (440) predictivamente el bloque actual usando el campo de movimiento, en donde las etapas (410, 420, 430, 440) se realizan en un procesador.
- 15 2. El método de la reivindicación 1, en donde las etapas (410, 420, 430, 440) se realizan en un codificador.
3. El método de la reivindicación 1, en donde se codifican componentes de textura y de profundidad de una imagen de referencia antes de que los componentes de textura y de profundidad de la imagen actual (515).
4. El método de la reivindicación 1, en donde componentes de textura de todas las visiones se codifican antes que los componentes de profundidad de la imagen actual (515).
- 20 5. El método de la reivindicación 1, en donde el vector de movimiento o de disparidad se deriva de una media de información de movimiento o de disparidad en los bloques vecinos (501).
6. El método de la reivindicación 1, en donde el vector de movimiento o de disparidad se obtiene sobre la base de información lateral señalada en un flujo de bits.
7. El método de la reivindicación 6, en donde la información lateral es un índice a una lista de candidatos de los vectores de movimiento o de disparidad.
- 25 8. El método de la reivindicación 1, en donde el bloque de profundidad (505) se basa en un promedio ponderado de múltiples bloques de profundidad.
9. El método de la reivindicación 2, en donde la codificación se basa en coste de distorsión de tasa.
10. El método de la reivindicación 1, en donde la imagen actual (515) es un componente de textura.
- 30 11. El método de la reivindicación 1, en donde la imagen actual (515) es un componente de profundidad.
12. El método de la reivindicación 1, en donde el vector de movimiento o vector de disparidad de los bloques vecinos (501) se usa para codificar los bloques vecinos.
13. El método de la reivindicación 11, en donde los vectores de disparidad de los bloques vecinos (501) se calculan sobre la base de valores de píxel en los bloques vecinos (501).
- 35 14. El método de la reivindicación 1, en donde el bloque de profundidad derivado (505) se interpola directamente a partir de los vectores de disparidad de los bloques vecinos (501).
15. El método de la reivindicación 1, en donde el procesamiento se usa en una aplicación de vídeo 3D.
16. El método de la reivindicación 1, en donde las etapas de obtención (410) y determinación (420) se realizan iterativamente.
- 40 17. El método de la reivindicación 1, en donde la etapa de sintetización (430) se realiza en un nivel de píxel.
18. El método de la reivindicación 1, en donde la etapa de sintetización (430) se realiza en un nivel de bloque.
19. El método de la reivindicación 1, que comprende además las etapas de:
- sintetizar, usando la deformación hacia atrás (506), un bloque de predicción correspondiente a la visión actual sobre la base de la imagen de referencia correspondiente a la visión de referencia y el bloque de profundidad determinado (505); y
- 45

combinar, para el bloque actual, el bloque de predicción y un bloque residual decodificado para producir un bloque de una imagen de salida.

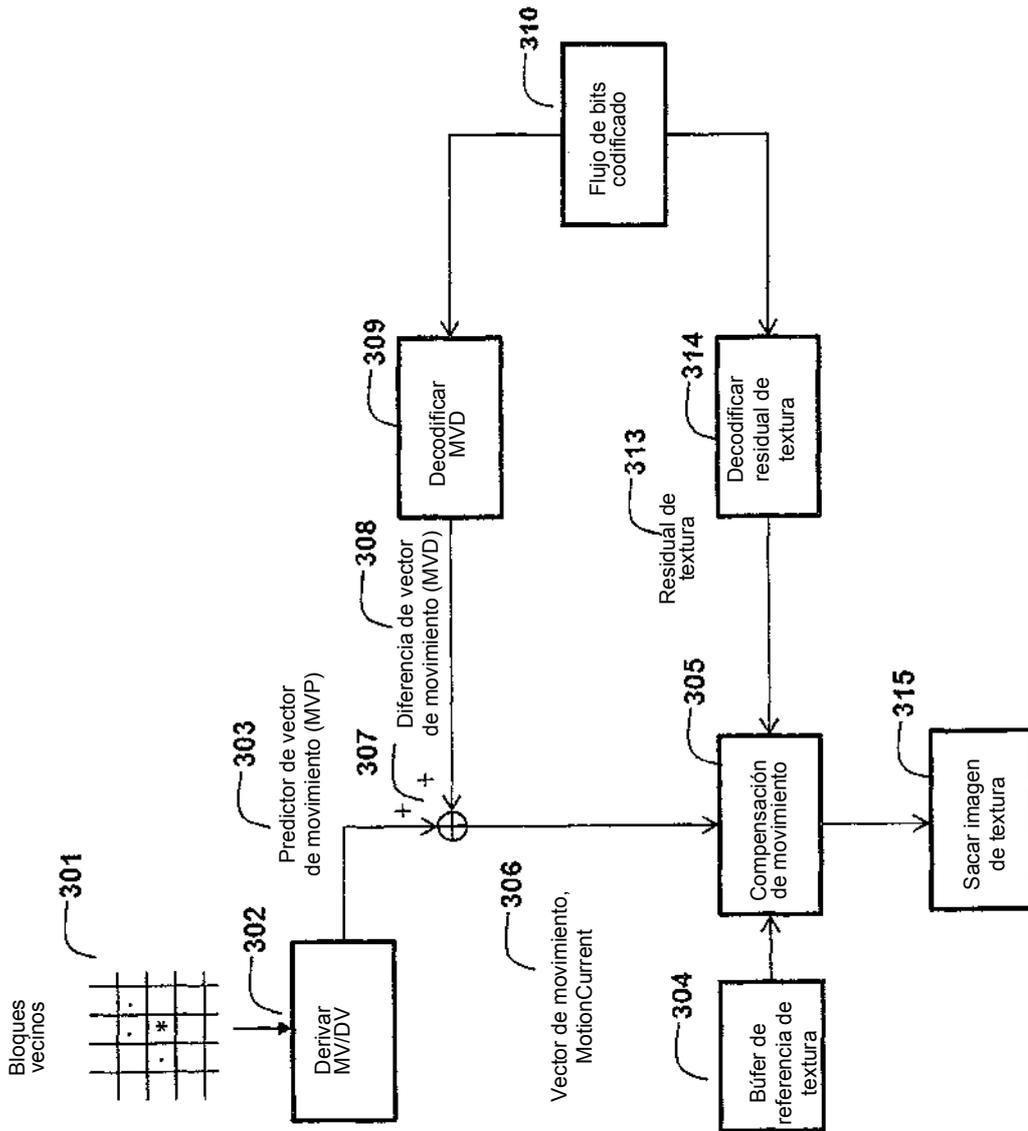
[Fig. 1]



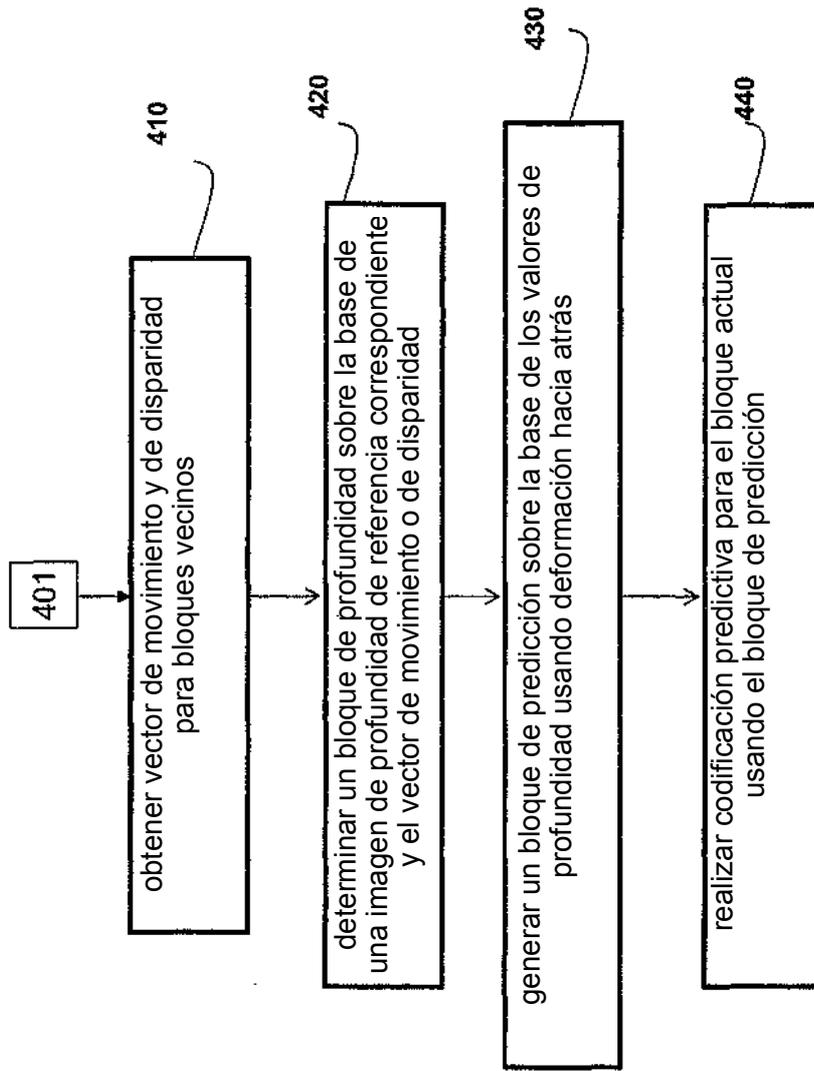
[Fig. 2]



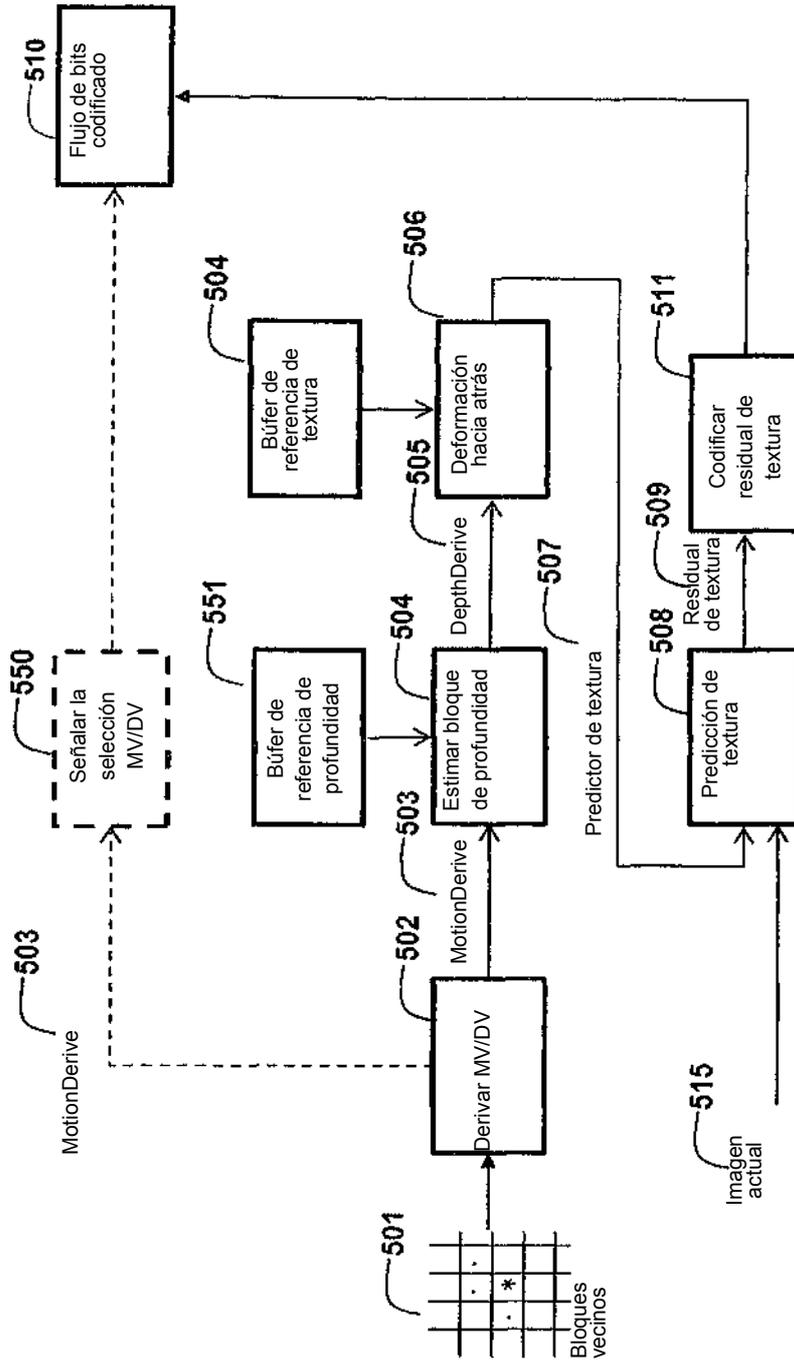
[Fig. 3]



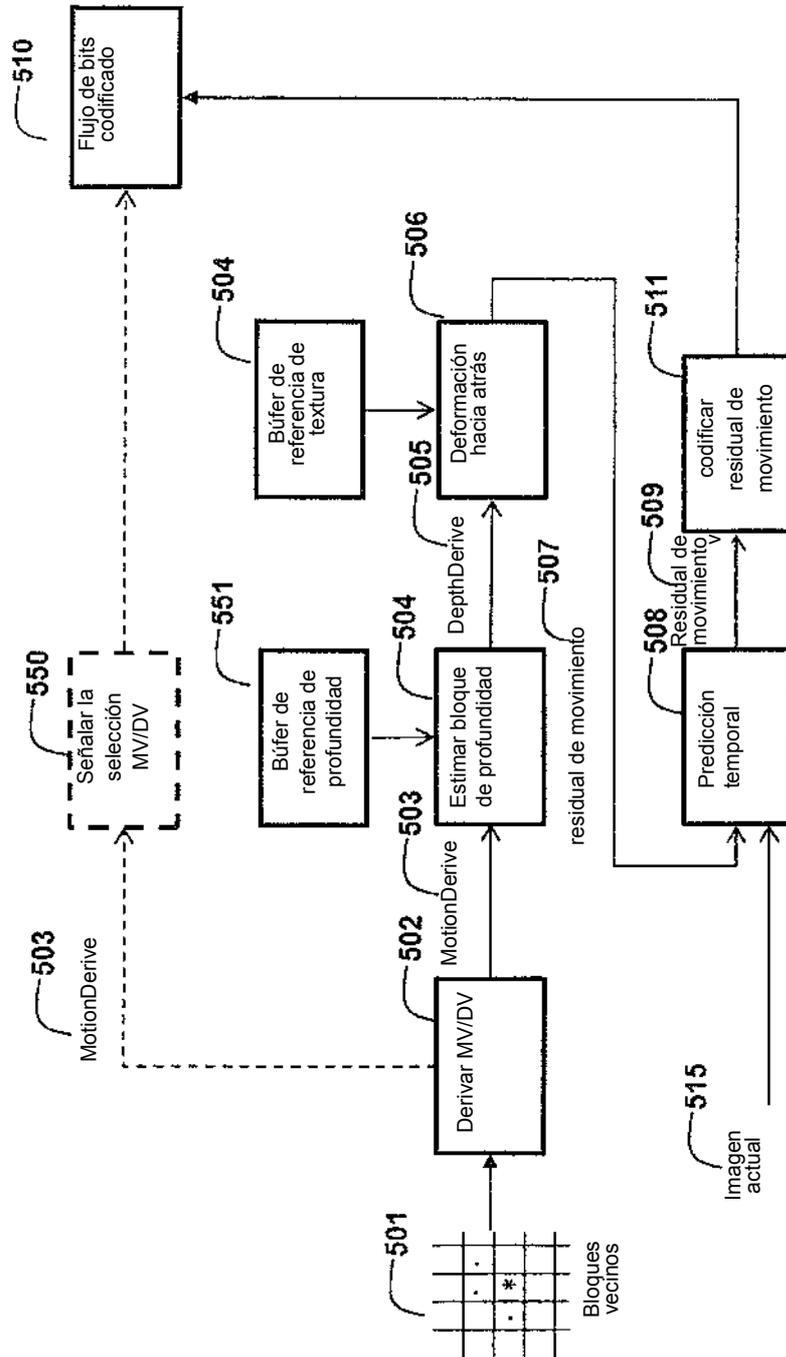
[Fig. 4]



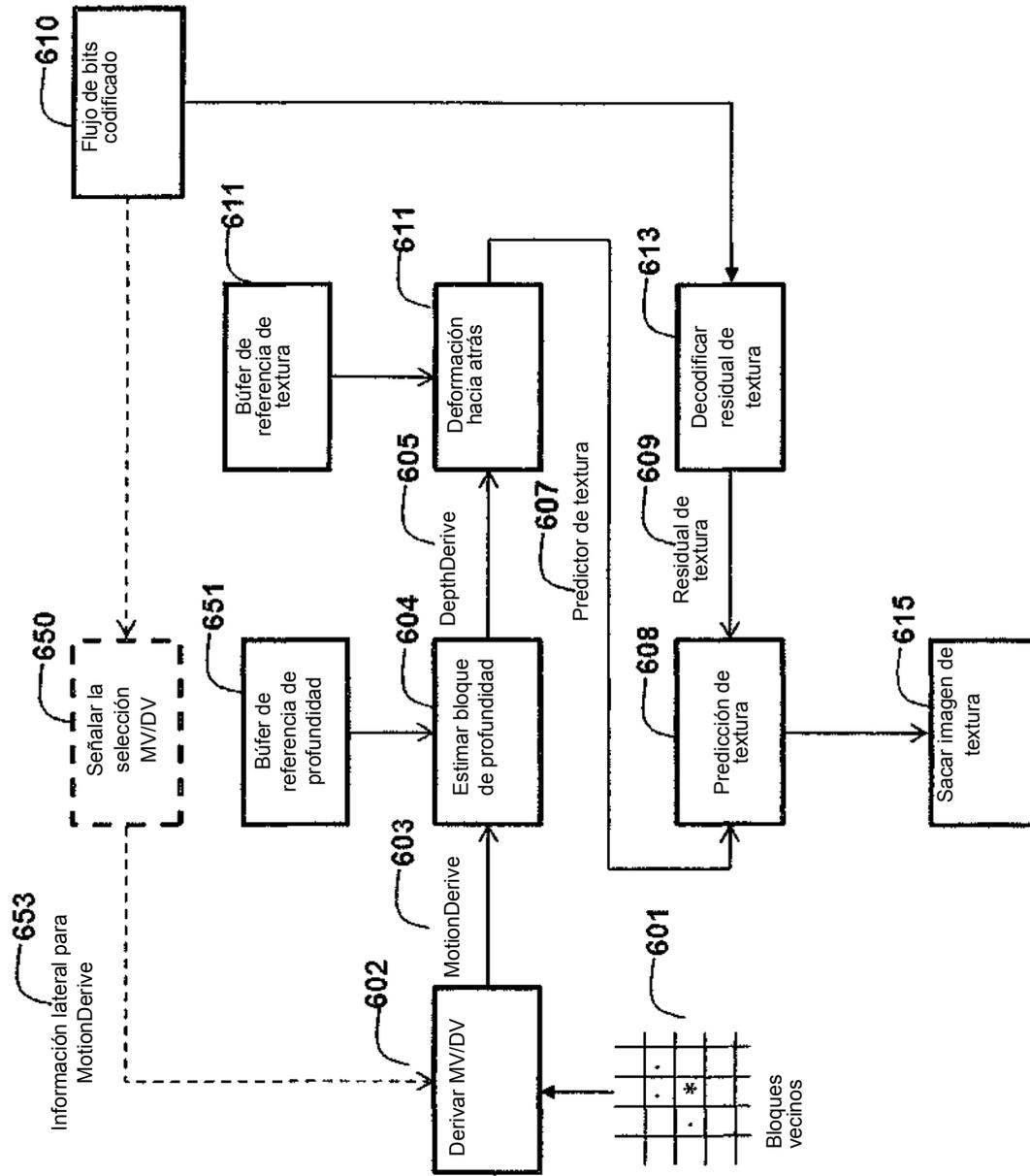
[Fig. 5A]



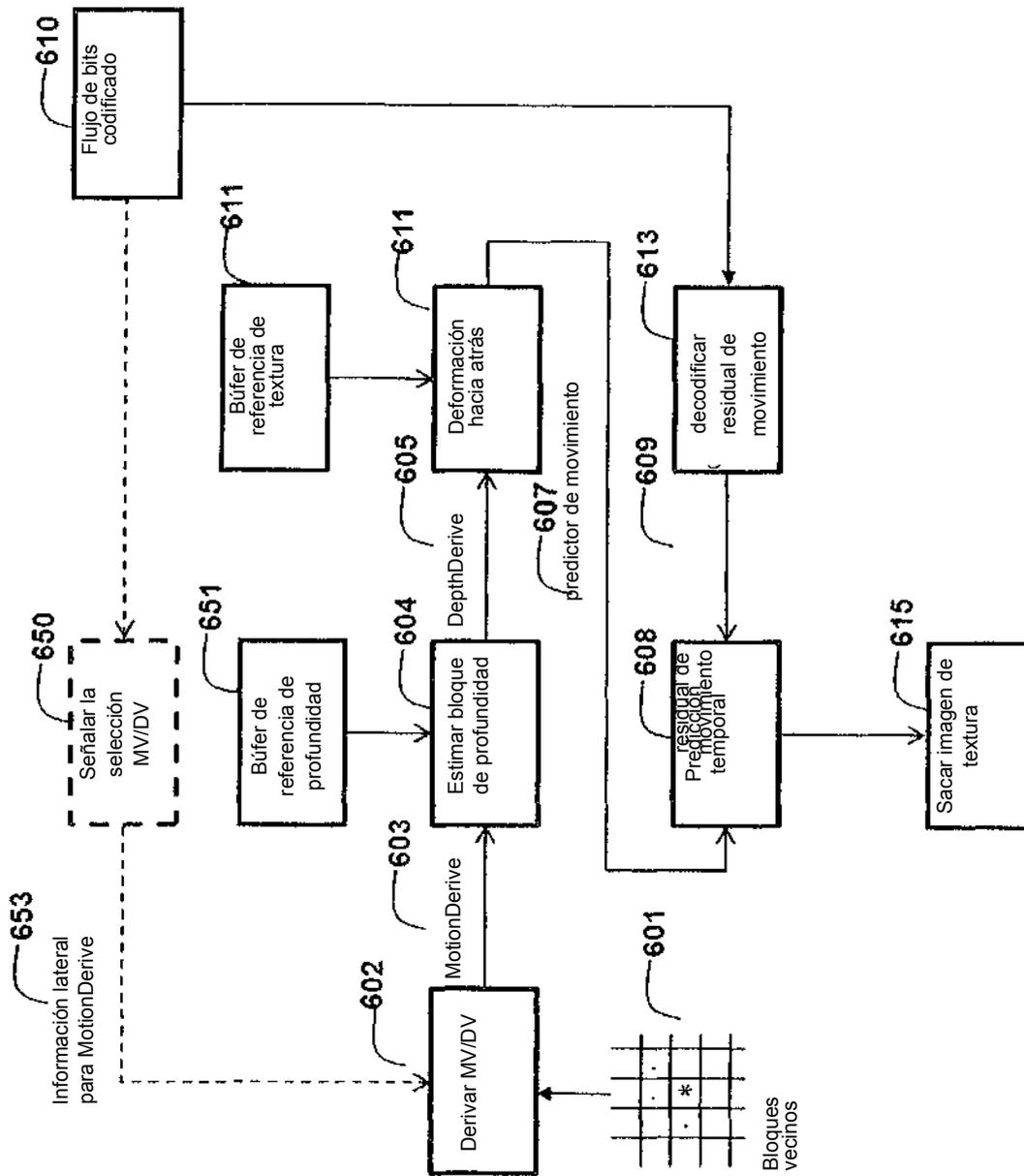
[Fig. 5B]



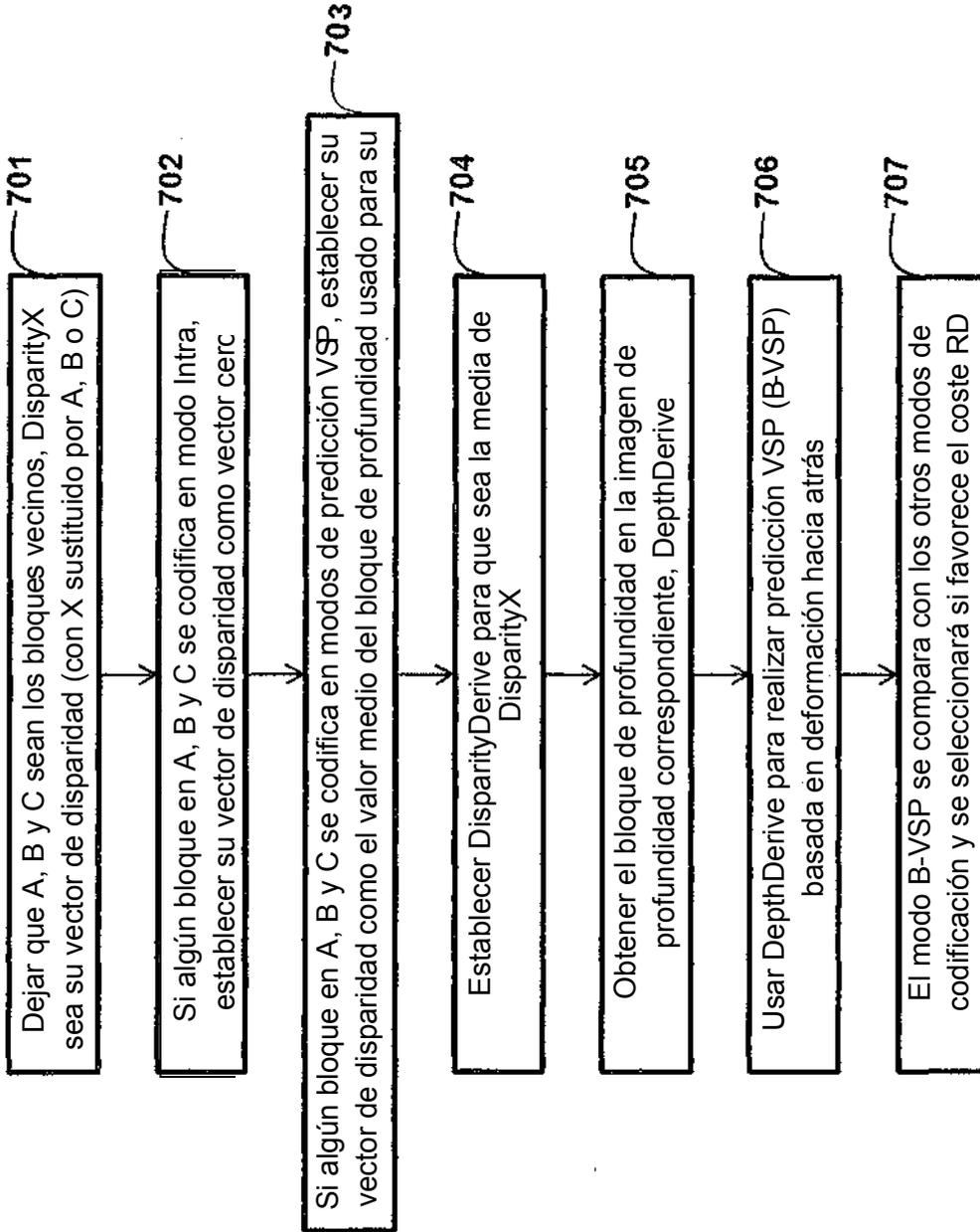
[Fig. 6A]



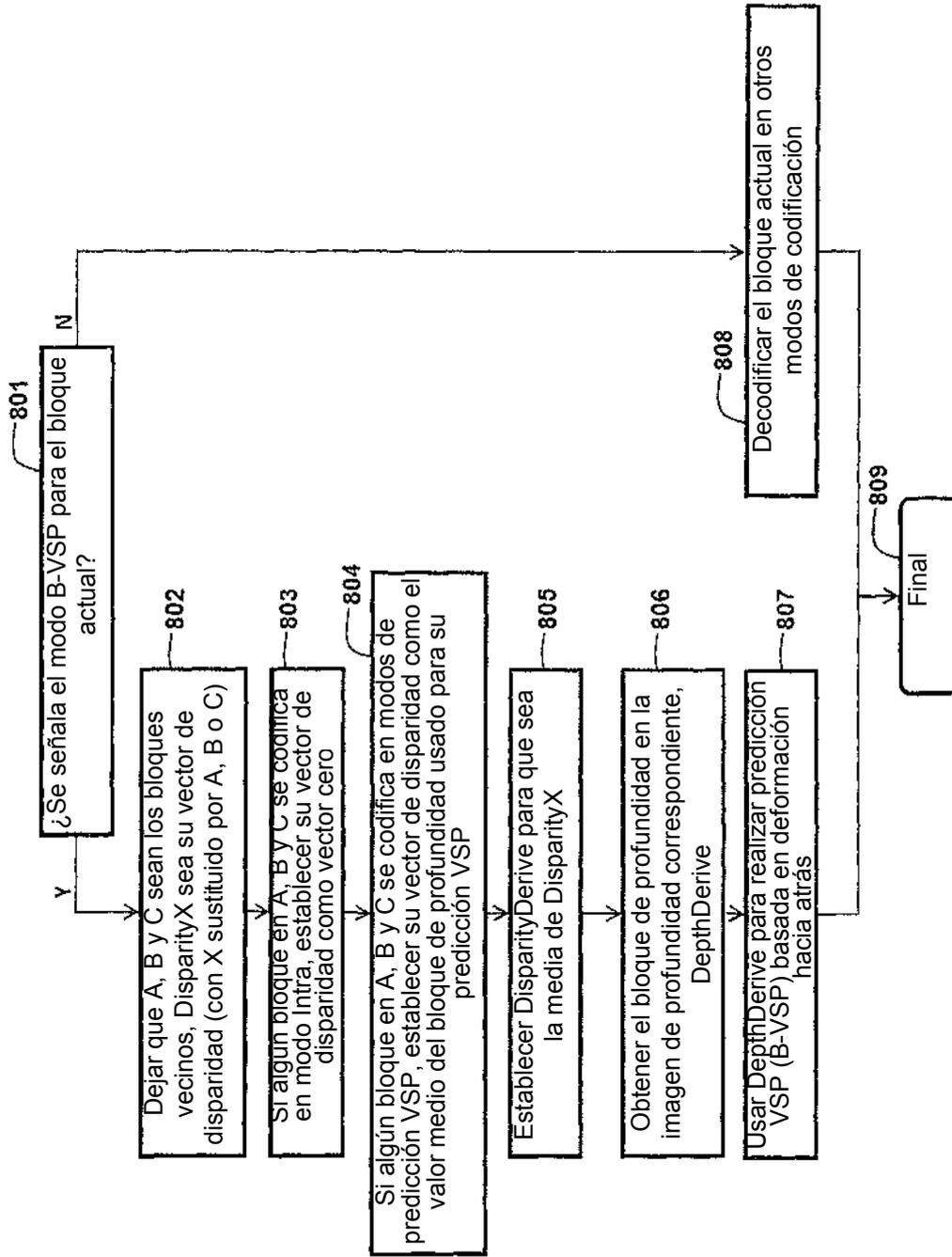
[Fig. 6B]



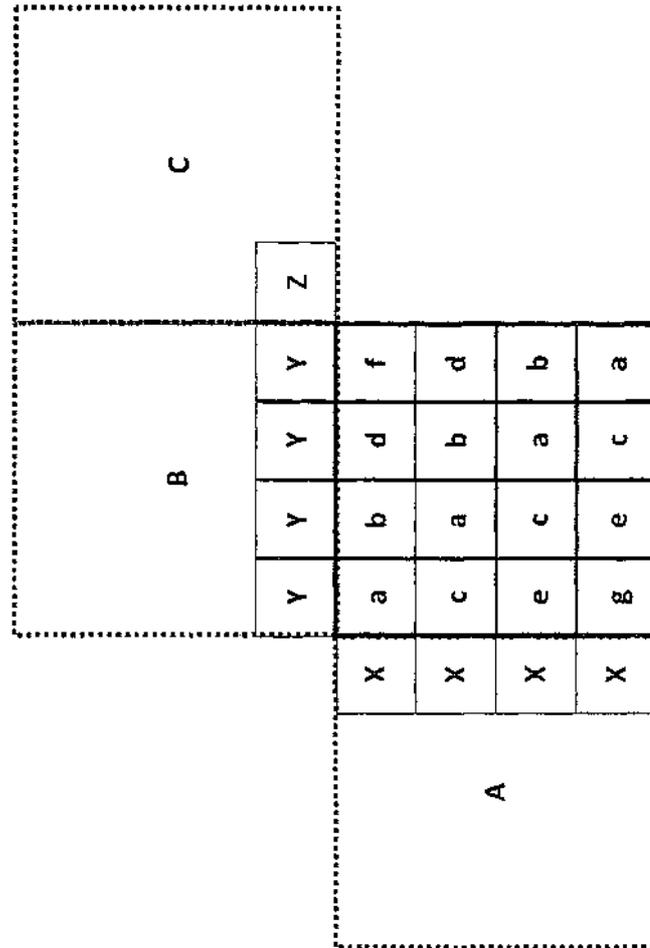
[Fig. 7]



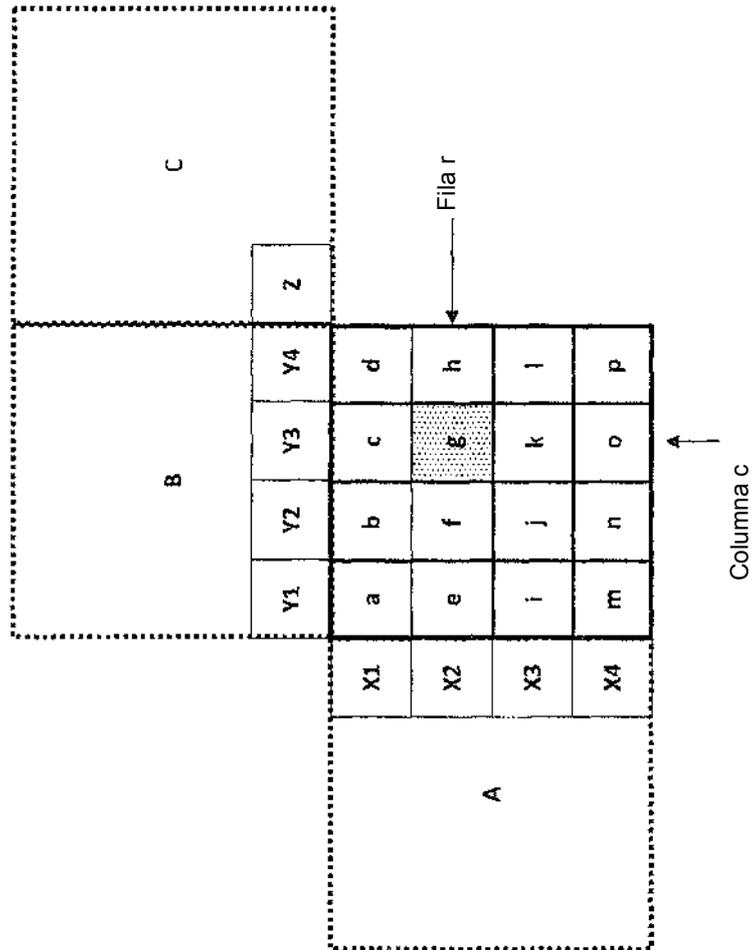
[Fig. 8]



[Fig. 9]



[Fig. 10]



[Fig. 11]

