

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 985 771**

51 Int. Cl.:

H04N 19/13 (2014.01)

H04N 19/167 (2014.01)

H04N 19/174 (2014.01)

H04N 19/196 (2014.01)

H04N 19/423 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **06.03.2013** **PCT/FR2013/050474**

87 Fecha y número de publicación internacional: **03.10.2013** **WO13144474**

96 Fecha de presentación y número de la solicitud europea: **06.03.2013** **E 13715270 (8)**

97 Fecha y número de publicación de la concesión europea: **29.05.2024** **EP 2832093**

54 Título: **Método y dispositivo de codificación y decodificación de imágenes**

30 Prioridad:

29.03.2012 FR 1252826

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
07.11.2024

73 Titular/es:

ORANGE (100.0%)
111, quai du Président Roosevelt
92130 Issy-les-Moulineaux, FR

72 Inventor/es:

HENRY, FELIX;
PATEUX, STÉPHANE y
CLARE, GORDON

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 985 771 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 y dispositivo de codificación y decodificación de imágenes

5 **Campo de la invención**

La presente invención hace referencia de manera general al campo del procesamiento de imágenes, y más exactamente a la codificación y decodificación de imágenes digitales y secuencias de imágenes digitales.

10 Así pues, la invención se puede aplicar en particular a la codificación de vídeo utilizada en los codificadores de vídeo actuales (MPEG, H.264, etc.) o futuros (ITU-T/VCEG (HEVC) o ISO/MPEG (HVC).

Antecedentes de la invención

15 El estándar HEVC, actualmente en desarrollo y descrito en el documento "Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 7th Meeting: Geneva, CH, 21-30 November, 2011" es similar al anterior estándar H.264, en el sentido de que utiliza una representación en bloques de la secuencia de vídeo. Sin embargo, el estándar HEVC difiere del estándar H.264 en que utiliza una subdivisión de la imagen en varias particiones de bloque llamadas "teselas, (traducción del inglés "tiles") según se describe en el documento "Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 6th Meeting: Torino, IT, 14-22July, 2011".

25 Más concretamente, una subdivisión de este tipo consiste en dividir cada imagen del vídeo mediante uno o varios límites verticales que se extienden desde la parte superior a la inferior de la imagen y uno o varios bordes horizontales que se extienden desde un lado (por ejemplo, el lado izquierdo) al otro (por ejemplo, el lado derecho) de la imagen. Los macrobloques situados entre dos límites verticales consecutivos y dos bordes horizontales consecutivos forman una tesela en la que los macrobloques vecinos se agrupan de este modo de dos en dos. En el interior de una tesela, los macrobloques se codifican y, a continuación, se decodifican según un orden lexicográfico, es decir, un escaneado línea por línea de los bloques, de tipo "barrido de trama", empezando por el bloque situado en la parte superior izquierda de la tesela y terminando por el bloque situado en la parte inferior derecha de la tesela. Las teselas se codifican y, a continuación, se decodifican en su orden lexicográfico. La **figura 1** muestra un ejemplo de subdivisión de una imagen IMG en un número predeterminado de teselas, por ejemplo, nueve teselas de T1 a T9. En el ejemplo:

- 35 • la tesela T1 comprende los macrobloques MB1 a MB9,
- la tesela T2 comprende los macrobloques MB10 a MB15,
- la tesela T3 comprende los macrobloques MB16 a MB24,
- 40 • la tesela T4 comprende los macrobloques MB25 a MB30,
- la tesela T5 comprende los macrobloques MB31 a MB34,
- la tesela T6 comprende los macrobloques MB35 a MB40,
- 45 • la tesela T7 comprende los macrobloques MB41 a MB43,
- la tesela T8 comprende los macrobloques MB44 a MB45,
- 50 • la tesela T9 comprende los macrobloques MB46 a MB48.

El conjunto de teselas T1, T4 y T7 está separado del conjunto de teselas T2, T5 y T8 por un límite vertical F1.

55 El conjunto de teselas T2, T5 y T8 está separado del conjunto de teselas T3, T6 y T9 por un límite vertical F2.

El conjunto de teselas T1, T2 y T3 está separado del conjunto de teselas T4, T5 y T6 por un límite horizontal F3.

El conjunto de teselas T4, T5 y T6 está separado del conjunto de teselas T7, T8 y T9 por un límite horizontal F4.

60 El número y el tamaño de las teselas se pueden establecer, imagen por imagen. Los macrobloques se numeran en el orden ("barrido de trama" en que se procesan en una tesela determinada, según se ha explicado anteriormente.

La ventaja de una subdivisión de este tipo en teselas es que no se permite la intra o interpredicción de un macrobloque de una tesela determinada, por ejemplo, el macrobloque MB10 de la tesela T2, con respecto a un macrobloque ya codificado y, a continuación, decodificado perteneciente a una tesela distinta de la tesela T2, por ejemplo, el

macrobloque MB3 de la tesela T1. En consecuencia, cada tesela se puede codificar y, a continuación, descodificar de forma independiente, lo que permite que estas teselas sean codificadas y/o decodificadas en paralelo en diferentes unidades de tratamiento, tratando cada unidad una tesela.

5 Al igual que en el estándar H.264, el estándar HEVC mencionado anteriormente propone que, para cada macrobloque considerado durante la codificación, se pueda codificar un macrobloque residual, correspondiente al macrobloque original menos una predicción. Un macrobloque residual de este tipo contiene coeficientes que se cuantifican después de cualquier transformación y, a continuación, se codifican mediante un codificador entrópico.

10 En este caso, nos interesa especialmente el codificador entrópico. El codificador entrópico codifica la información en el orden en que llega. Normalmente, se realiza un escaneado línea por línea de los bloques, de tipo "barrido de trama", empezando por el bloque situado en la parte superior izquierda de la imagen. Para cada bloque, los distintos elementos de información necesarios para representarlo (tipo de bloque, modo de predicción, coeficientes residuales, etc.) se envían de forma secuencial al codificador entrópico.

15 Ya conocemos un codificador aritmético eficaz de complejidad razonable, denominado "CABAC" (Context Adaptive Binary Arithmetic Coder en inglés), introducido en el estándar de compresión AVC (también conocido como ISO-MPEG4 parte 10 y UIT-T H.264).

20 Este codificador entrópico utiliza diferentes conceptos:

- la codificación de símbolos aritméticos basada en la utilización de una probabilidad de aparición del símbolo;
- 25 • la adaptación al contexto: en este caso se trata de adaptar la probabilidad de aparición de los símbolos que hay que codificar. Por una parte, se lleva a cabo un aprendizaje sobre la marcha. Por otra parte, en función del estado de la información codificada previamente, se utiliza un contexto específico para la codificación. Cada contexto corresponde a una probabilidad específica de que aparezca el símbolo. Por ejemplo, un contexto corresponde a un tipo de símbolo codificado (la representación de un coeficiente de un residuo, la señalización del modo de codificación, etc.) según una configuración dada, o a un estado de la vecindad (por ejemplo, el número de modos "intra" seleccionados en la vecindad, etc.) ;
- la binarización: los símbolos que hay que codificar se convierten en una secuencia de bits. A continuación, estos diferentes bits se envían de forma sucesiva al codificador entrópico binario.

35 De este modo, para cada contexto utilizado, este codificador entrópico utiliza un sistema de aprendizaje de probabilidades sobre la marcha con respecto a los símbolos previamente codificados para el contexto considerado. Este aprendizaje se basa en el orden de codificación de estos símbolos. Cuando comienza la codificación entrópica de un macrobloque actual, las probabilidades de aparición de símbolos utilizados son las obtenidas tras codificar un macrobloque previamente codificado y, a continuación, descodificado, que es el que precede inmediatamente al bloque actual de acuerdo con el escaneado de los bloques.

40 La desventaja de un codificador entrópico de este tipo es que la codificación y descodificación de un símbolo depende del estado de la probabilidad aprendida hasta ese momento, de modo que los símbolos sólo pueden descodificarse en el mismo orden en que fueron codificados.

En el futuro estándar HEVC, la codificación entrópica se realiza tesela a tesela en orden del "barrido de trama" de éstas últimas, en lugar de línea a línea.

50 De este modo, en el ejemplo mostrado en la **figura 1**, la codificación entrópica CABAC se aplica de forma sucesiva a los macrobloques MB1 a MB9 de la tesela T1, a los macrobloques MB10 a MB15 de la tesela T2, a los macrobloques MB16 a MB24 de la tesela T3, los macrobloques MB25 a MB30 de la tesela T4, los macrobloques MB31 a MB34 de la tesela T5, los macrobloques MB35 a MB40 de la tesela T6, los macrobloques MB41 a MB43 de la tesela T7, los macrobloques MB44 a MB45 de la tesela T8 y los macrobloques MB46 a MB48 de la tesela T9.

55 Se observa, sin embargo, que podría ser interesante descodificar una imagen que se ha codificado subdividiéndola previamente en teselas, conservando el orden del "barrido de trama" de las líneas de la imagen, tal como se propone en el estándar H.264, en lugar de utilizar el orden del "barrido de trama" de las teselas. Una disposición de este tipo tiene en efecto interés con el objetivo de suministrar líneas de macrobloques de píxeles adecuadas para su visualización en pantallas de visualización que generalmente funcionan según un barrido horizontal de las líneas de píxeles.

60 Consideremos el caso de la imagen IMG subdividida en teselas según se muestra en la **figura 1**. Como se ha explicado anteriormente, en el estándar HEVC que se está desarrollando actualmente, los macrobloques se codifican en orden de barrido de trama de cada tesela T1 a T9. La codificación de cada uno de estas teselas genera el correspondiente flujo de bits SFX1, SFX2,..., SFX9, estando dispuesto cada subflujo en un flujo comprimido final FX, en el orden en

que se procesa la tesela asociado. Para simplificar, en la **figura 1** sólo se muestran los tres primeros subflujos SFX1, SFX2, SFX3. Acto seguido, cada subflujo se descodifica de forma sucesiva en el orden en que aparece en el flujo FX.

Sin embargo, por las razones mencionadas anteriormente, puede ser deseable descodificar la imagen IMG de acuerdo con el estándar H.264, es decir, línea de macrobloque por línea de macrobloque en la imagen IMG, en el siguiente orden: MB1, MB2, MB3, MB10, MB11, MB16, MB17, MB18, MB4, MB5, MB6, MB12, MB13, MB19, MB20, MB21 y así sucesivamente hasta el macrobloque MB48.

Como los macrobloques de la imagen IMG se descodifican en el orden definido por las teselas, para visualizar la primera línea de macrobloques MB1, MB2, MB3, MB10, MB11, MB16, MB17, MB18, es necesario descodificar todos los macrobloques MB1 a MB9 del subflujo SFX1 antes de poder empezar a descodificar los macrobloques MB10 y MB11 del subflujo SFX2. Del mismo modo, sería necesario descodificar todos los macrobloques MB10 a MB15 del subflujo SFX2 antes de poder empezar a descodificar los macrobloques MB16, MB17 y MB18 del subflujo SFX3. De este modo, únicamente para la descodificación de la primera línea de macrobloques de la imagen, una disposición de este tipo introduce un retraso significativo en la descodificación de dicha primera línea y requiere que todos los píxeles descodificados relativos a los macrobloques MB1 a MB9 de la tesela T1, a los macrobloques MB10 a MB15 de la tesela T2, a los macrobloques MB16 a MB24 de la tesela T3 se mantengan en memoria, lo que consume memoria. En consecuencia, es aconsejable:

- descodificar sólo los macrobloques MB1, MB2, MB3, MB10, MB11, MB16, MB17, MB18 para visualizar la primera línea de macrobloques,
- descodificar sólo los macrobloques MB4, MB5, MB6, MB12, MB13, MB19, MB20, MB21 para visualizar la segunda línea de macrobloques,
- y así sucesivamente para las siguientes líneas de macrobloques en la imagen IMG.

Consideremos el caso de la descodificación de la primera línea de macrobloques de la imagen IMG. Un puntero de flujo PO escanea el subflujo SFX1 para leer los datos necesarios para descodificar los macrobloques MB1, MB2, MB3. Teniendo en cuenta el hecho de que el descodificador entrópico CABAC comparte bits entre macrobloques consecutivos, cuando el puntero PO ha escaneado suficientes bits para poder descodificar los datos del macrobloque MB3, también habrá leído un cierto número de bits necesarios para descodificar al menos el siguiente macrobloque MB4, estos bits se denominan "bits compartidos". El puntero PO debe entonces "saltar" al inicio del siguiente subflujo SFX2 para descodificar los datos de los macrobloques MB10 y MB11. Un salto de este tipo es posible porque cada subflujo tiene una dirección de inicio que está escrita en una cabecera del flujo FX completo. Antes de realizar este salto, el descodificador debe:

- memorizar la posición del puntero PO en una memoria de punteros MP1, de forma que posteriormente sea posible escanear el subflujo SFX1 para continuar leyendo los bits correspondientes a los macrobloques MB4, MB5, MB6 de la segunda línea de la imagen IMG,
- memorizar los bits compartidos en una memoria de bits BM1.

El puntero PO comienza entonces a leer los bits del subflujo SFX2 con vistas a descodificar los macrobloques MB10 y MB11. De forma similar, a como se ha descrito anteriormente, el descodificador debe:

- memorizar la posición del puntero PO en una memoria de punteros MP2, de forma que el subflujo SFX2 pueda ser escaneado de nuevo más tarde para continuar leyendo los bits correspondientes a los macrobloques MB12 y MB13 de la segunda línea de la imagen IMG,
- memorizar los bits compartidos en una memoria de bits BM2.

Un procedimiento de este tipo continúa en cada paso del límite de la tesela durante la descodificación de la primera línea de la imagen IMG.

Cuando finaliza la descodificación de la primera línea de la imagen IMG, se realiza la descodificación de la segunda línea de esta imagen. El puntero PO se establece entonces en el valor almacenado previamente en la memoria MP1, y los bits compartidos contenidos en la memoria BM1 se transfieren a la memoria actual del descodificador CABAC. De este modo, puede comenzar la descodificación del macrobloque MB4. Este procedimiento también se utiliza para descodificar los macrobloques MB12 y MB13 del subflujo SFX2 y los macrobloques MB19, MB20 y MB21 del subflujo SFX3.

A continuación, las siguientes líneas de la imagen IMG se descodifican con el procedimiento que se acaba de describir.

El método de descodificación de acuerdo con este procedimiento es ineficaz y engorroso de ejecutar, dadas las múltiples operaciones que tiene que llevar a cabo y la gran cantidad de información que debe procesar y almacenar. El resultado es un método de descodificación que tarda demasiado en procesarse y consume demasiados recursos de memoria.

Objetivo y resumen de la invención

Uno de los objetivos de la invención es remediar las desventajas de la técnica anterior mencionadas anteriormente.

Para ello, un objetivo de la presente invención hace referencia a un método de descodificación de al menos una imagen, tal como se define en la reivindicación 1 anexa.

De acuerdo con un primer ejemplo, las variables de estado del módulo de descodificación entrópica son los dos límites de un intervalo que representa la probabilidad de aparición de un símbolo entre los símbolos del conjunto predeterminado de símbolos.

De acuerdo con un segundo ejemplo, las variables de estado del módulo de descodificación entrópica son un límite y la longitud de un intervalo que representa la probabilidad de aparición de un símbolo entre los símbolos del conjunto predeterminado de símbolos.

De acuerdo con un tercer ejemplo, las variables de estado del módulo de descodificación entrópica son las cadenas de símbolos contenidas en la tabla de conversión de un codificador entrópico LZW (Lempel-Ziv-Welch) bien conocido por el experto en la técnica, y descrito en la siguiente dirección de Internet el 29 de marzo de 2012: <http://en.wikipedia.org/wiki/Lempel%E2%80%93Ziv%E2%80%93Welch>.

Una disposición de este tipo permite descodificar el flujo previamente codificado de tal manera que los bloques de este flujo se descodifiquen unos tras otros en el orden del "barrido de trama" en el que se disponen en la imagen y no de un subflujo al siguiente en el orden del "barrido de trama" en el que se disponen por tesela.

Esta disposición es de particular interés cuando se desea mostrar la imagen línea por línea de bloques, en el orden del "barrido de trama" de los bloques de imagen, en particular como parte de la visualización de la imagen en una pantalla de visualización que generalmente funciona según un barrido horizontal de las líneas de píxeles.

Además, gracias a que el módulo de descodificación entrópica se reinicializa cada vez que un bloque descodificado recorre un límite entre la tesela que se está descodificando y la tesela que le sigue en orden de barrido de trama, no es necesario, con vistas a reanudar la descodificación de la segunda línea de bloques de dicha tesela, una vez descodificados los bloques de la primera línea de la tesela siguiente:

- memorizar previamente los bits que han sido compartidos durante la descodificación de la primera línea de bloques de dicha tesela (respectivamente durante la descodificación de la primera línea de bloques de dicha tesela siguiente), con vistas a ser utilizados para descodificar los bloques que siguen al último bloque de la primera línea de dicha tesela en el orden del "barrido de trama" (respectivamente para descodificar los bloques que siguen al último bloque de la primera línea de dicha tesela siguiente en el orden del "barrido de trama"),
- transferir los bits compartidos a la memoria actual del descodificador entrópico cuando se reanude la codificación de la segunda línea de bloques de la tesela (o cuando se reanude la codificación de la segunda línea de bloques de la siguiente tesela, respectivamente).

El resultado es un método de descodificación menos costoso de calcular y más rápido de ejecutar.

Al mismo tiempo, la invención hace referencia a un dispositivo para descodificar un flujo representativo de al menos una imagen codificada, tal como se define en la reivindicación 2 anexa.

La invención también hace referencia a un programa informático que comprende instrucciones para la ejecución de las etapas del método de descodificación anterior, cuando el programa es ejecutado por un ordenador.

Un programa de este tipo puede utilizar cualquier lenguaje de programación, y estar en forma de código fuente, código objeto, o código intermedio entre código fuente y código objeto, como por ejemplo en una forma parcialmente compilada, o en cualquier otra forma deseable.

Sin embargo, otro objetivo de la invención también hace referencia a un medio de grabación legible por ordenador que comprende instrucciones de programa informático tal como se mencionó anteriormente.

El medio de grabación puede ser cualquier entidad o dispositivo capaz de almacenar el programa. Por ejemplo, un soporte de este tipo puede incluir un medio de almacenamiento, como una ROM, por ejemplo un CD ROM o una ROM de circuito microelectrónico, o incluso un medio de grabación magnética, por ejemplo un disquete o un disco duro.

5 Por otra parte, un medio de grabación de este tipo puede ser un medio transmisible, como por ejemplo una señal eléctrica u óptica, que se puede dirigir por medio de un cable eléctrico u óptico, por radio o por otros medios. En particular, el programa de acuerdo con la invención se puede descargar de una red de tipo Internet.

10 Alternativamente, un medio de grabación de este tipo puede ser un circuito integrado en el que se incorpore el programa, estando el circuito adaptado para ejecutar el método en cuestión o para ser utilizado en la ejecución de éste último.

El dispositivo de decodificación y el programa informático mencionados anteriormente tienen al menos las mismas ventajas que las conferidas por el método de decodificación de acuerdo con la presente invención.

15 **Breve descripción de los dibujos**

Otras características y ventajas se desprenderán con la lectura de las formas de realización preferidas descritas con referencia a las figuras adjuntas, en las que los aspectos relativos a la codificación no forman parte de la invención y se presentan únicamente a título ilustrativo, y en las que:

- la Figura 1 muestra un esquema de codificación de imágenes de la técnica anterior,
- la figura 2A muestra las principales etapas del método de codificación de acuerdo con un ejemplo,
- la Figura 2B muestra en detalle el método de codificación utilizado en la Figura 2A,
- la figura 3 muestra un ejemplo de forma de realización de un dispositivo de codificación,
- la figura 4 muestra un esquema de codificación/decodificación de imágenes de acuerdo con una forma de realización preferida,
- la Figura 5A muestra las principales etapas del método de decodificación de acuerdo con la invención,
- la Figura 5B muestra en detalle la decodificación utilizada en el método de decodificación de la Figura 5A,
- la figura 6 muestra una forma de realización de un dispositivo de decodificación de acuerdo con la invención.

20 **Descripción detallada de un primer ejemplo de forma de realización de la parte de codificación**

A continuación se describirá un ejemplo de forma de realización, en el que el método de codificación se utiliza para codificar una secuencia de imágenes según un flujo de bits próximo al obtenido por codificación según el estándar H.264/MPEG-4 AVC. En este ejemplo de forma de realización, el método de codificación se implementa en software o hardware modificando un codificador que inicialmente cumple con el estándar H.264/MPEG-4 AVC. El método de codificación se representa en forma de un algoritmo que comprende las etapas C1 a C5, mostradas en la **figura 2A**.

Según el ejemplo de forma de realización, el método de codificación se implementa en un dispositivo de codificación CO, cuyo ejemplo de forma de realización se muestra en la **figura 3**.

Con referencia a la **figura 2A**, el primer etapa de codificación C1 es el troceado de una imagen IE de una secuencia de imágenes a codificar en varios bloques $B_1, B_2, \dots, B_i, \dots, B_K$, con $1 \leq i \leq K$, según se muestra en la **figura 4**. En el ejemplo mostrado, $K=40$. Un macrobloque tradicionalmente es un bloque que tiene un tamaño máximo predeterminado. Además, un macrobloque de este tipo se puede dividir en bloques más pequeños. En aras de la simplicidad, el término "bloque" se utilizará por tanto indistintamente para designar un bloque o un macrobloque. Dichos bloques $B_1, B_2, \dots, B_i, \dots, B_K$ pueden contener uno o más símbolos, formando parte dichos símbolos de un conjunto predeterminado de símbolos. En el ejemplo mostrado, dichos bloques tienen forma cuadrada y son todos del mismo tamaño. Dependiendo del tamaño de la imagen, que no es necesariamente múltiplo del tamaño de los bloques, los últimos bloques de la izquierda y los últimos bloques de la parte inferior pueden no ser cuadrados. En una forma de realización alternativa, los bloques pueden ser rectangulares y/o no estar alineados entre sí.

Un troceado de este tipo se lleva a cabo mediante un módulo PCO de partición que se muestra en la **figura 3** y que utiliza, por ejemplo, un algoritmo de partición conocido por se.

Con referencia a la **figura 2A**, la segunda etapa de codificación C2 es la reagrupación de los bloques vecinos mencionados anteriormente dos a dos en un número predeterminado N de particiones de la imagen denominadas teselas TU₁, TU₂,..., TU_j,..., TU_N, con $1 \leq j \leq N$ que están destinadas a codificarse de forma secuencial en orden del "barrido de trama", según se ha explicado anteriormente en la descripción de la técnica anterior. En el ejemplo mostrado en la **figura 4**, los bloques B₁ a B_K se agrupan en N=3 teselas TU₁, TU₂ TU₃. En aras de la simplicidad, la subdivisión en teselas se lleva a cabo a lo largo de dos límites únicamente verticales FR₁ y FR₂, y no horizontales. Huelga decir que el método de codificación también se aplica a la subdivisión de la imagen en límites verticales y horizontales o en límites horizontales únicamente, según las aplicaciones de vídeo previstas.

Los dos límites verticales FR₁ y FR₂ delimitan la imagen IE:

- la primera tesela TU₁ que contiene los bloques B₁ a B₁₅,
- la segunda tesela TU₂ que contiene los bloques B₁₆ a B₂₅
- la tercera tesela TU₃ que contiene los bloques B₂₆ a B₄₀.

Una reagrupación de este tipo se lleva a cabo mediante un módulo de cálculo GRCO que se muestra en la **figura 3**, mediante un algoritmo bien conocido per se.

Con referencia a la **figura 2A**, la tercera etapa de codificación C3 consiste en codificar cada una de las teselas de bloques TU₁, TU₂,..., TU_j,..., TU_N, según un orden de escaneado del "barrido de trama", codificándose los bloques de una tesela TU_j determinada según un orden de ejecución del "barrido de trama". En el ejemplo mostrado en la **figura 4**, las teselas TU₁, TU₂, TU₃ se codifican una tras otra, y los bloques de una tesela actual, por ejemplo la tesela TU₁, se codifican uno tras otro, en el orden del "barrido de trama" indicado por la flecha PRS. En aras de la claridad de la **figura 4**, la flecha PRS no se muestra en las teselas TU₂ y TU₃.

Una codificación de este tipo es secuencial, por ejemplo, y se lleva a cabo mediante una unidad de codificación UC que codifica de forma sucesiva en el orden del "barrido de trama" un número N de teselas a codificar, siendo N=3 en el ejemplo mostrado en la **figura 3**.

De una manera conocida per se, la unidad de codificación UC comprende:

- un módulo de codificación predictiva de un bloque actual con respecto a al menos un bloque previamente codificado y, a continuación, descodificado, denominado MCP;
- un módulo para la codificación entrópica de dicho bloque actual utilizando al menos una probabilidad de aparición de símbolo calculada para dicho bloque previamente codificado y, a continuación, descodificado, denominado MCE.

El módulo de codificación predictiva MCP es un módulo capaz de llevar a cabo la codificación predictiva del bloque actual utilizando técnicas de predicción convencionales, como el modo Intra y/o Inter.

El módulo de codificación entrópica MCE es, por su parte, del tipo CABAC, pero modificado de acuerdo con la presente invención, según se describirá más adelante en la descripción.

En particular, las modificaciones introducidas en el módulo de codificación entrópica MCE se basan en las conclusiones del documento: Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, y Ajay Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, pp. 560-576, julio de 2003. En este documento, se especifica que el codificador entrópico CABAC tiene la particularidad de asignar un número no entero de bits a cada símbolo de un alfabeto actual a codificar. En términos prácticos, el codificador CABAC espera hasta haber leído varios símbolos y, a continuación, asigna un número predeterminado de bits a este conjunto de símbolos leídos, que el codificador introduce en el flujo comprimido que se transmitirá al descodificador. Una disposición de este tipo permite agrupar los bits en varios símbolos y codificar un símbolo en un número fraccionario de bits, ya que este número refleja una información más próxima a la que transporta realmente un símbolo. Otros bits asociados a los símbolos leídos no se transmiten en el flujo comprimido, sino que se conservan a la espera de ser asignados a uno o más símbolos nuevos leídos por el codificador CABAC, lo que permite de nuevo compartir estos otros bits. De una manera conocida per se, el codificador entrópico procede a un "volcado" de estos bits no transmitidos en un momento determinado. En otras palabras, en ese momento determinado, el codificador extrae los bits aún no transmitidos y los escribe en el flujo comprimido con destino al descodificador. Un volcado de este tipo se produce, por ejemplo, en el momento en que se ha leído el último símbolo que se debe codificar, con el fin de garantizar que el flujo comprimido contiene todos los bits que permitirán al descodificador descodificar todos los símbolos del alfabeto. De forma más general, el momento en que se lleva a cabo el volcado se determina en función del rendimiento y la funcionalidad de un codificador/descodificador determinado.

Alternativamente, el módulo de codificación entrópica MCE podría ser un codificador Huffman conocido per se.

5 En el ejemplo mostrado en la **figura 4**, si la tesela actual a codificar es, por ejemplo, la tesela TU_1 , la unidad de codificación UC codifica los bloques B_1 , B_2 , B_3 de la primera línea de la tesela TU_1 , de izquierda a derecha. Cuando alcanza el último bloque B_3 de dicha primera línea, pasa al primer bloque B_4 de la segunda línea de la tesela TU_1 . Cuando alcanza el último bloque B_6 de dicha segunda línea, pasa al primer bloque B_7 de la tercera línea de la tesela TU_1 y así sucesivamente hasta que se haya codificado el último bloque B_{15} de la tesela TU_1 .

10 Por supuesto, son posibles otros tipos de órdenes de ejecución distintos del descrito anteriormente. Dependiendo de la aplicación, la unidad de codificación podría procesar no una sucesión de líneas, como se ha explicado anteriormente, sino una sucesión de columnas. También es posible recorrer las filas o columnas en cualquier dirección.

15 Con referencia a la **figura 2A**, la cuarta etapa de codificación C4 es la producción de un número de N subflujos de bits $SF_1, SF_2, \dots, SF_j, \dots, SF_N$ que representan los bloques procesados comprimidos respectivamente por la unidad de codificación UC mencionada anteriormente, así como una versión descodificada de los bloques procesados de cada tesela TU_j . Los bloques procesados descodificados de cada una de dichas teselas, denominados $TUD_1, TUD_2, \dots, TUD_j, \dots, TUD_N$, se pueden reutilizar respectivamente por la unidad de codificación UC mostrada en la **figura 3**.

20 En referencia a la **figura 3**, el etapa de producir un subflujo de corriente SF_j es utilizada por un módulo de software de generación de flujos MGSF que está adaptado para producir flujos de datos, como bits, por ejemplo.

25 Con referencia a la **figura 2A**, la quinta etapa de codificación C5 consiste en construir un flujo global F a partir de los N subflujos $SF_1, SF_2, \dots, SF_j, \dots, SF_N$ mencionados anteriormente. De acuerdo con un ejemplo de forma de realización, los subflujos $SF_1, SF_2, \dots, SF_j, \dots, SF_N$ se yuxtaponen simplemente en orden de teselas como el mostrado en la **figura 4**, con información adicional destinada a indicar al descodificador la ubicación de cada subflujo en el flujo global F. Este último se transmite acto seguido a través de una red de comunicaciones (no mostrada) a un terminal remoto. Esto incluye el descodificador DO mostrado en la **figura 6**.

30 Una construcción de este tipo del flujo global F se utiliza en un módulo de construcción de flujo CF, como el mostrado en la **figura 3**.

35 Con referencia a la **figura 2B**, describiremos ahora las diferentes subetapas específicas de la invención, tal como se ejecutan durante la etapa de codificación de teselas C3 mencionada anteriormente, en la unidad de codificación UC.

En el curso de una etapa C31, la unidad de codificación UC selecciona como bloque actual el primer bloque a codificar de una tesela TU_j actual, como por ejemplo la primera tesela TU_1 de la imagen IE mostrada en la **figura 4**.

40 En el curso de una etapa C32, la unidad de codificación UC comprueba si el bloque actual es el primer bloque (situado en la parte superior izquierda) de una línea de la tesela TU_1 que se obtuvo al final del etapa C2 mencionada anteriormente.

45 Si éste es el caso, teniendo en cuenta el hecho de que el primer bloque B_1 es también el primer bloque de la imagen IE, el módulo de codificación entrópica MCE inicializa sus variables de estado en el curso de una etapa C33. De acuerdo con el ejemplo mostrado, que utiliza la codificación aritmética descrita anteriormente, esto implica inicializar un intervalo representativo de la probabilidad de aparición de un símbolo contenido en el conjunto predeterminado de símbolos. De una manera conocida per se, este intervalo se inicializa con dos límites L y H, inferior y superior respectivamente. El valor del límite inferior L se fija en 0, mientras que el valor del límite superior se fija en 1, que corresponde a la probabilidad de que aparezca un primer símbolo entre todos los símbolos del conjunto predeterminado de símbolos. Por lo tanto, el tamaño R de este intervalo se define en esta fase por $R=H-L=1$. El intervalo inicializado también se divide tradicionalmente en varios subintervalos predeterminados que son respectivamente representativos de las probabilidades de aparición de los símbolos en el conjunto predeterminado de símbolos.

55 Alternativamente, las variables internas inicializadas son uno de los dos límites del intervalo mencionado anteriormente y su longitud.

De acuerdo con otro ejemplo, si la codificación entrópica utilizada es la codificación LZW, se inicializa una tabla de conversión de cadenas de símbolos de forma que contenga todos los símbolos posibles una y sólo una vez.

60 En el curso de la etapa C34, que sigue a la etapa C33 mencionada anteriormente, se realiza la codificación del primer bloque actual B_1 de la primera línea de la tesela TU_1 mostrada en la **figura 4**. Una etapa C34 de este tipo comprende varias subetapas C341 a C348, que se describen a continuación.

En el curso de una primera subetapa C341, se realiza la codificación predictiva del bloque actual B_1 utilizando técnicas conocidas de intra y/o interpredicción, en el curso de las cuales el bloque B_1 se predice en relación con al menos un bloque previamente codificado y, a continuación, descodificado.

5 Ni que decir tiene que son posibles otros modos de intrapredicción como los propuestos en el estándar H.264.

El bloque actual B_1 también puede someterse a una codificación predictiva en modo inter, en el curso de la cual el bloque actual se predice en relación con un bloque procedente de una imagen previamente codificada y descodificada. Por supuesto, son posibles otros tipos de predicción. Entre las predicciones posibles para un bloque actual, la predicción óptima se elige de acuerdo con un criterio de tasa de distorsión bien conocido por los expertos en la técnica.

Dicha etapa de codificación predictiva mencionada anteriormente permite construir un bloque predicho B_{p1} que es una aproximación del bloque actual B_1 . La información relativa a esta codificación predictiva se escribirá posteriormente en el flujo F transmitido al descodificador DO. Dicha información incluye, en particular, el tipo de predicción (inter o intra) y, en su caso, el modo de intrapredicción, el tipo de partición de un bloque o macrobloque si éste se ha subdividido, el índice de imagen de referencia y el vector de desplazamiento utilizado en el modo de predicción inter. Esta información se comprime mediante el codificador CO mostrado en la **figura 3**.

En el curso de una subetapa posterior C342, se realiza la sustracción del bloque predicho B_{p1} del bloque actual B_1 para producir un bloque residual B_{r1} .

En el curso de una subetapa posterior C343, se realiza la transformación del bloque residual B_{r1} según una operación de transformación directa convencional como, por ejemplo, una transformación coseno discreta de tipo DCT, para producir un bloque transformado B_{t1} .

En el curso de una subetapa posterior C344, se realiza la cuantificación del bloque transformado B_{t1} según una operación de cuantificación convencional como, por ejemplo, la cuantificación escalar. Se obtiene entonces un bloque de coeficientes cuantificados B_{q1} .

En el curso de una subetapa C345 posterior, se realiza la codificación entrópica del bloque de coeficientes cuantificados B_{q1} . En la forma de realización preferida, se trata de una codificación entrópica CABAC. Una etapa de este tipo consiste en:

- leer el símbolo o símbolos del conjunto predeterminado de símbolos asociados a dicho bloque actual,
- asociar información digital, como bits, al símbolo o símbolos leídos.

En el ejemplo mencionado anteriormente, según el cual la codificación utilizada es una codificación LZW, la información digital correspondiente al código del símbolo de la tabla de conversión actual se asocia al símbolo que se va a codificar, y la tabla de conversión se actualiza según un método conocido per se.

En el curso de una subetapa posterior C346, se realiza la descuantificación del bloque B_{q1} según una operación de descuantificación convencional, que es la operación inversa de la cuantificación llevada a cabo en la etapa C344. Se obtiene entonces un bloque de coeficientes descuantificados BD_{q1} .

En el curso de una subetapa posterior C347, se realiza la transformación inversa del bloque de coeficientes descuantificados BD_{q1} , que es la operación inversa de la transformación directa llevada a cabo en la etapa C343 anterior. Se obtiene entonces un bloque residual descodificado BD_{r1} .

En el curso de una subetapa posterior C348, se realiza la construcción del bloque descodificado BD_1 añadiendo el bloque predicho B_{p1} al bloque residual descodificado BD_{r1} . Cabe señalar que este último bloque es el mismo que el bloque descodificado obtenido al final del método de descodificación de la imagen IE que se describirá más adelante en la descripción. El bloque descodificado BD_1 queda de este modo disponible para ser utilizado por la unidad de codificación UC.

Al final de la etapa de codificación C34 mencionada anteriormente, el módulo de codificación entrópica MCE, según se muestra en la **figura 3**, contiene todas las probabilidades actualizadas progresivamente a medida que se codifica el primer bloque. Estas probabilidades corresponden a los diferentes elementos sintácticos posibles y a los diferentes contextos de codificación asociados.

La etapa de codificación C34 también se lleva a cabo directamente si, al final de la etapa C32 mencionada anteriormente, el bloque actual no es el primer bloque de una línea de la tesela TU_1 .

En el curso de una etapa C35 posterior, la unidad de codificación UC comprueba si el bloque codificado/decodificado actual es el último bloque de una línea de la tesela TU_1 . Con referencia a la **figura 4**, una etapa de este tipo consiste

en comprobar si el lado derecho del bloque actual toca el borde del límite FR_1 que es adyacente al borde del límite FR_2 .

Si el bloque actual que se ha codificado/decodificado es el último bloque de una línea de la tesela TU_1 , es decir, con referencia a la **figura 4**, si se verifica que el bloque actual B_3 tiene su lado derecho discurrendo a lo largo del borde B_1T_1 de la tesela TU_1 que forma una parte del límite FR_1 con el borde B_1T_2 de la segunda tesela TU_2 , la unidad de codificación UC de la **figura 3** lleva a cabo en el curso de una etapa C36 siguiente, un volcado según se ha descrito anteriormente en la descripción. Para ello, la unidad de codificación UC transmite al módulo de generación de subflujos MGSF todos los bits que estaban asociados al símbolo o símbolos leídos durante la codificación de cada bloque de la línea de la tesela TU_1 considerada, de forma que el módulo MGSF escribe dicha totalidad de bits en el subflujo de datos SF_1 que se muestra en la **figura 4** y que contiene un flujo de bits representativo de los bloques codificados de dicha línea. Un volcado de este tipo se simboliza en la **figura 4** mediante un triángulo al final de cada línea de tesela.

En el curso de una etapa C37, la unidad de codificación UC lleva a cabo una etapa idéntica a la etapa C33 mencionada anteriormente, es decir, inicializa de nuevo el intervalo representativo de la probabilidad de aparición de un símbolo contenido en el conjunto predeterminado de símbolos.

El interés de llevar a cabo las etapas C36 y C37 en este nivel de codificación es que cuando se codifica el siguiente bloque procesado por la unidad de codificación UC, el codificador CO se encuentra en un estado inicializado. De este modo, según se describirá más adelante, es posible que una unidad de descodificación descodifique el flujo comprimido F directamente desde este punto, ya que sólo necesita estar en el estado inicializado.

En el curso de una etapa posterior C38, la unidad de codificación UC comprueba si el bloque actual que se acaba de codificar/descodificar es el último bloque de la imagen IE.

El etapa C38 también se lleva a cabo en el caso de que el bloque actual que se ha procesado al final de la etapa C34 mencionada anteriormente no sea el último bloque de una línea de la tesela actual que sigue a la etapa de prueba C35.

Si el bloque actual es el último bloque de la imagen IE, se finaliza el método de codificación en el curso de la etapa C39.

Si éste no es el caso, se realiza la selección del siguiente bloque a codificar de acuerdo con el orden de escaneado del "barrido de trama" en la tesela TU_1 y se repiten las etapas C31 y siguientes. En el ejemplo mostrado en la **figura 4**, se trata del bloque B_2 .

El método de codificación descrito anteriormente se lleva a cabo para todas las líneas que hay que codificar en una tesela determinada.

Descripción detallada de una forma de realización de la parte de codificación

A continuación, se describirá una forma de realización del método de descodificación de acuerdo con la invención, en el que el método de descodificación se implementa en software o hardware modificando un descodificador que inicialmente cumple con el estándar H.264/MPEG-4 AVC.

El método de descodificación de acuerdo con la invención se representa en forma de un algoritmo que comprende las etapas D1 a D4, mostradas en la **figura 5A**.

De acuerdo con la forma de realización de la invención, el método de descodificación de acuerdo con la invención se implementa en un dispositivo de descodificación DO mostrado en la **figura 6**.

Con referencia a la **figura 5A**, el primer etapa de descodificación D1 es la identificación en dicho flujo F de los N subflujos $SF_1, SF_2, \dots, SF_j, \dots, SF_N$ respectivamente que contienen las N teselas $TU_1, TU_2, \dots, TU_j, \dots, TU_N$ de los bloques $B_1, B_2, \dots, B_i, \dots, B_K$, con $1 \leq i \leq K$ que han sido previamente codificadas, según lo mostrado en la **figura 4**. Para ello, cada subflujo $SF_1, SF_2, \dots, SF_j, \dots, SF_N$ del flujo F se asocia a un indicador destinado a permitir al descodificador DO determinar la ubicación de cada subflujo en el flujo F.

En el ejemplo mostrado en la **figura 4**, los bloques son cuadrados y todos del mismo tamaño. Dependiendo del tamaño de la imagen, que no es necesariamente múltiplo del tamaño de los bloques, los últimos bloques de la izquierda y los últimos bloques de la parte inferior pueden no ser cuadrados. En una forma de realización alternativa, los bloques pueden ser rectangulares y/o no estar alineados entre sí.

Cada bloque o macrobloque puede a su vez dividirse en subbloques, que son a su vez subdivisibles.

Una identificación de subflujos de este tipo se lleva a cabo mediante un módulo de extracción de flujos EXDO, según se muestra en la **figura 6**.

Con referencia a la **figura 5A**, la segunda etapa de decodificación D2 es la decodificación de cada una de dichas teselas $TU_1, TU_2, \dots, TU_j, \dots, TU_N$, entregando respectivamente N teselas de bloque decodificadas $TUD_1, TUD_2, \dots, TUD_j, \dots, TUD_N$

Una única unidad de decodificación UD lleva a cabo la decodificación las teselas de bloques, según se muestra en la **figura 6**.

Según se muestra con más detalle en la **figura 6**, la unidad de codificación UD comprende:

- un módulo de decodificación entrópica de dicho bloque actual mediante el aprendizaje de al menos una probabilidad de aparición de símbolo calculada para al menos un bloque decodificado previamente, denominado MDE,
- un módulo de decodificación predictiva de un bloque actual con respecto a dicho bloque decodificado previamente, denominado MDP.

El módulo de decodificación predictiva MDP es capaz de llevar a cabo la decodificación predictiva del bloque actual de acuerdo con las técnicas de predicción convencionales, tales como el modo Intra y/o Inter.

El módulo de decodificación entrópica MDE es, por su parte, del tipo CABAC, pero modificado de acuerdo con la presente invención, según se describirá más adelante.

Alternativamente, el módulo de decodificación entrópica MDE podría ser un decodificador Huffman conocido per se.

En el ejemplo mostrado en la **figura 4**, si la tesela actual a decodificar es, por ejemplo, la tesela TU_1 , la unidad UD decodifica los bloques B_1, B_2, B_3 de la primera línea de la tesela TU_1 , de izquierda a derecha. Cuando alcanza el último bloque B_3 de dicha primera línea, pasa al primer bloque B_4 de la segunda línea de la tesela TU_1 . Cuando alcanza el último bloque B_6 de dicha segunda línea, pasa al primer bloque B_7 de la tercera línea de la tesela TU_1 y así sucesivamente hasta que se haya decodificado el último bloque B_{15} de la tesela TU_1 .

Por supuesto, son posibles otros tipos de órdenes de ejecución distintos del descrito anteriormente. Dependiendo de la aplicación, la unidad de decodificación UD podría procesar no una sucesión de líneas, como se ha explicado anteriormente, sino una sucesión de columnas. También es posible recorrer las filas o columnas en cualquier dirección.

Con referencia a la **figura 5A**, la tercera etapa de decodificación D3 es la reconstrucción de una imagen decodificada ID a partir de cada tesela decodificada $TUD_1, TUD_2, \dots, TUD_j, \dots, TUD_N$ obtenida en la etapa de decodificación D2. Más concretamente, los bloques decodificados de cada tesela decodificada se transmiten a una unidad URI de reconstrucción de imagen, según se muestra en la **figura 6**. En el curso de esta etapa D3, la unidad URI escribe los bloques decodificados en una imagen decodificada a medida que estos bloques están disponibles.

En el curso de una cuarta etapa de decodificación D4 mostrada en la **figura 5A**, la unidad URI mostrada en la **figura 6** emite una imagen ID totalmente decodificada.

Con referencia a la **figura 5B**, describiremos ahora las diferentes subetapas específicas de la invención, según se llevan a cabo durante la etapa de decodificación D2 mencionada anteriormente, en la unidad de decodificación UD.

En el curso de una etapa D21, la unidad de decodificación UD selecciona como bloque actual en el subflujo correspondiente el primer bloque a decodificar de una tesela actual TU_j , según, por ejemplo, la primera tesela TU_1 de la imagen IE mostrada en la **figura 4**.

En el curso de una etapa D22, la unidad de decodificación UD comprueba si el bloque actual es el primer bloque (situado en la parte superior izquierda) de una línea de la tesela TU_1 , en este caso el primer bloque B_1 del subflujo SF_1 .

Si éste es el caso, teniendo en cuenta el hecho de que el primer bloque B_1 es también el primer bloque de la imagen IE, el módulo de decodificación entrópica MDE realiza, en el curso de una etapa D23, una inicialización de sus variables de estado. De acuerdo con el ejemplo mostrado, se trata una inicialización de un intervalo representativo de la probabilidad de aparición de un símbolo contenido en el conjunto predeterminado de símbolos.

Alternativamente, si la decodificación entrópica utilizada es la decodificación LZW, se inicializa una tabla de conversión de cadenas de símbolos de forma que contenga todos los símbolos posibles una y sólo una vez. Como la etapa D23 es idéntica a la etapa de codificación C33 mencionada anteriormente, no se describirá más adelante.

En el curso de una etapa D24 que sigue a la etapa D23 mencionada anteriormente, se realiza la decodificación del primer bloque actual B_1 de la primera línea de la tesela TU_1 mostrada en la **figura 4**. Una etapa D24 de este tipo comprende varias subetapas D241 a D246, que se describen a continuación.

En el curso de una primera subetapa D241, se realiza la descodificación entrópica de los elementos sintácticos vinculados al bloque actual leyendo el flujo F mediante un puntero de flujo P. Una etapa de este tipo consiste principalmente en:

- leer los bits contenidos al principio del subflujo SF_1 asociado al primer bloque codificado B_1 ,
- reconstruir los símbolos a partir de los bits leídos.

En la variante mencionada anteriormente en la que la descodificación utilizada es una descodificación LZW, se lee la información digital correspondiente al código del símbolo en la tabla de conversión actual, se reconstruye el símbolo a partir del código leído y se lleva a cabo una actualización de la tabla de conversión mediante un método conocido per se.

Más concretamente, los elementos de sintaxis vinculados al bloque actual son descodificados por el módulo MDE de descodificación entrópica CABAC, según se muestra en la **figura 6**. Este último descodifica una primera parte del subflujo de bits SF_1 del fichero comprimido para producir los elementos sintácticos y, al mismo tiempo, actualiza sus probabilidades para que, cuando descodifique un símbolo, las probabilidades de aparición de este símbolo sean idénticas a las obtenidas al codificar este mismo símbolo durante la etapa de codificación entrópica C345 mencionada anteriormente.

En el curso de una subetapa D242 posterior, se realiza la descodificación predictiva del bloque actual B_1 utilizando técnicas conocidas de intra y/o interpredicción, durante las cuales el bloque B_1 se predice con respecto a al menos un bloque descodificado previamente.

Ni que decir tiene que son posibles otros modos de intrapredicción como los propuestos en el estándar H.264.

En el curso de esta etapa, se realiza la descodificación predictiva utilizando los elementos de sintaxis descodificados en la etapa precedente, incluyendo en particular el tipo de predicción (inter o intra) y, si procede, el modo de intrapredicción, el tipo de partición de un bloque o macrobloque si este último se ha subdividido, el índice de la imagen de referencia y el vector de desplazamiento utilizado en el modo de interpredicción.

Dicha etapa de descodificación predictiva mencionada anteriormente permite construir un bloque predicho Bp_1 .

En el curso de una subetapa posterior D243, se realiza la construcción de un bloque residual cuantificado Bq_1 utilizando los elementos sintácticos descodificados anteriormente.

En el curso de una subetapa D244 posterior, se realiza la descuantificación del bloque residual cuantificado Bq_1 según una operación de descuantificación convencional que es la inversa de la operación de cuantificación llevada a cabo en la etapa C344 mencionada anteriormente, para producir un bloque descodificado descuantificado BDt_1 .

En el curso de una subetapa D245 posterior, se realiza la transformación inversa del bloque BDt_1 descuantificado, que es la operación inversa de la transformación directa llevada a cabo en la etapa C343 anterior. Se obtiene entonces un bloque residual descodificado BDr_1 .

En el curso de una subetapa posterior D246, se realiza la construcción del bloque descodificado BD_1 añadiendo el bloque predicho Bp_1 al bloque residual descodificado BDr_1 . El bloque descodificado BD_1 queda de este modo disponible para ser utilizado por la unidad decodificadora UD.

Al final de la etapa de descodificación D246 mencionada anteriormente, el módulo de descodificación entrópica MDE, según se muestra en la **figura 6**, contiene todas las probabilidades según se actualizan de forma progresiva a medida que se descodifica el primer bloque. Estas probabilidades corresponden a los diferentes elementos sintácticos posibles y a los diferentes contextos de descodificación asociados.

La etapa de descodificación D24 también se ejecuta directamente si, al final de la etapa D22 mencionada anteriormente, el bloque actual no es el primer bloque de una línea de la tesela TU_1 .

En el curso de una etapa posterior D25, la unidad de descodificación UD comprueba si el bloque descodificado actual se identifica como el último bloque de una línea de la tesela TU_1 .

Si el bloque actual descodificado es el último bloque de una línea de la tesela TU_1 , por ejemplo, el bloque B_3 mostrado en la **figura 4**, la unidad de descodificación UD de la **figura 6** lleva a cabo, en el curso de una etapa D26, una etapa idéntica a la etapa D23 mencionada anteriormente, es decir, inicializa de nuevo el intervalo que representa la probabilidad de aparición de un símbolo contenido en el conjunto predeterminado de símbolos.

En el curso de una etapa D27 posterior, la unidad de decodificación UD comprueba si el bloque actual que se acaba de decodificar es el último bloque de la imagen IE, es decir, el último bloque del último subflujo, el subflujo SF₃ en el ejemplo mostrado en la **figura 4**.

5 Una etapa D27 de este tipo también se ejecuta en el caso de que el bloque actual que se ha decodificado al final de la etapa D24 mencionada anteriormente no sea el último bloque de una línea de la tesela actual que sigue a la etapa de prueba D25.

10 Si el bloque decodificado actual es efectivamente el último bloque decodificado de la imagen IE, en el curso de una etapa D28, se termina el método de decodificación.

Si éste no es el caso, se realiza la selección del siguiente bloque a decodificar de acuerdo con el orden de escaneado del "barrido de trama" en la tesela TU₁ y se repiten las etapas D21 y siguientes. En el ejemplo mostrado en la **figura 4**, se trata del bloque B₂ contenido en el subflujo SF₁.

15 El interés de la etapa de inicialización D26 mencionada anteriormente reside en el hecho de que la unidad de decodificación UD se encuentra en un estado inicializado cada vez que el puntero P del flujo F de la **figura 4** se encuentra en una parte de subflujo SF₁, SF₂ o SF₃ que corresponde a los datos de un primer bloque codificado de una línea de tesela asociada al subflujo considerado, con vistas a decodificar esta línea.

20 Una disposición de este tipo es especialmente adecuada para los casos en los que se desea decodificar línea de bloques por línea de bloques la imagen IE, en particular como parte de la visualización de la imagen en una pantalla de visualización que generalmente funciona según un barrido horizontal de líneas de píxeles.

25 De este modo, la unidad de decodificación UD podría desplazar el puntero P hasta el inicio del subflujo SF₂ para comenzar a leer la primera parte del subflujo SF₂ correspondiente, respectivamente, a la primera línea de bloques codificados B₁₆ a B₁₇ de la segunda tesela TU₂ y decodificar los bloques de esta línea de acuerdo con el método de decodificación anterior. Una vez llevada a cabo la decodificación de estos bloques, la unidad de decodificación UD se encontraría entonces en un estado inicializado con vistas a procesar la siguiente línea de bloques B₁₈ a B₁₉. De este modo, la unidad de decodificación UD podría desplazar el puntero P hasta el inicio del subflujo SF₃ para comenzar a leer la primera parte del subflujo SF₃ correspondiente, respectivamente, a la primera línea de bloques codificados B₂₆ a B₂₈ de la tercera tesela TU₃ y decodificar los bloques de esta línea de acuerdo con el método de decodificación anterior. Una vez llevada a cabo la decodificación de estos bloques, la unidad de decodificación UD se encontraría entonces en un estado inicializado con vistas a procesar la siguiente línea de bloques B₂₉ a B₃₁.

35 Un procedimiento de este tipo se repetiría entonces para cada línea de bloques de la imagen IE.

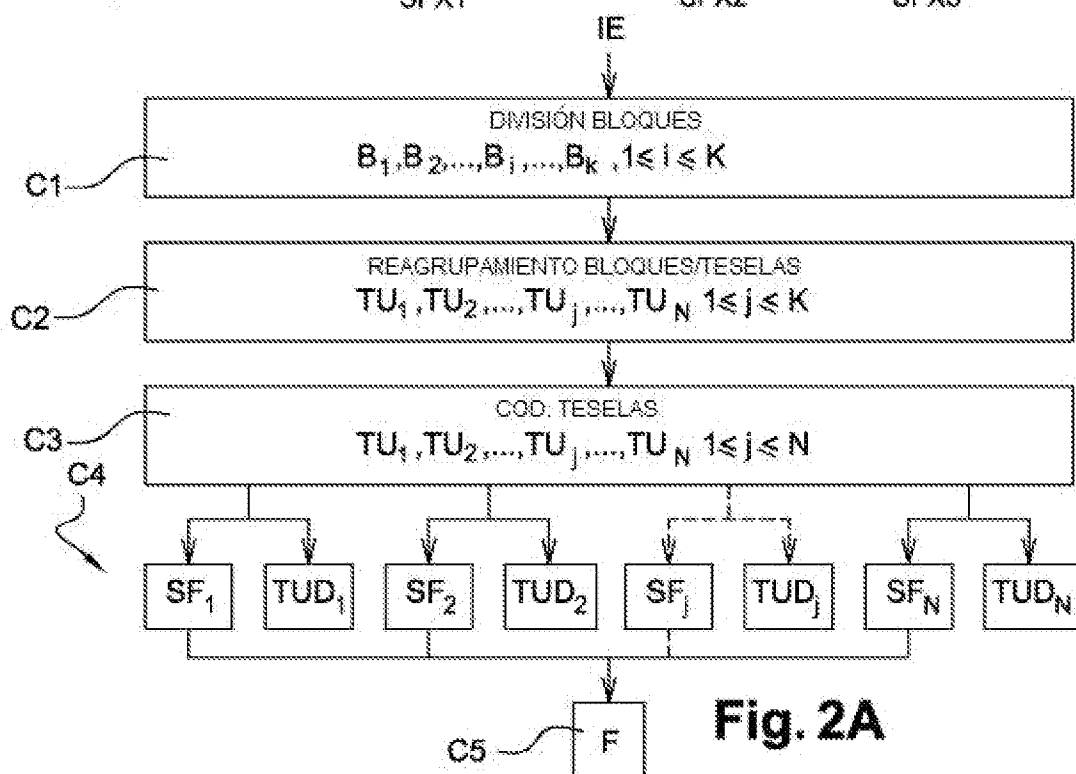
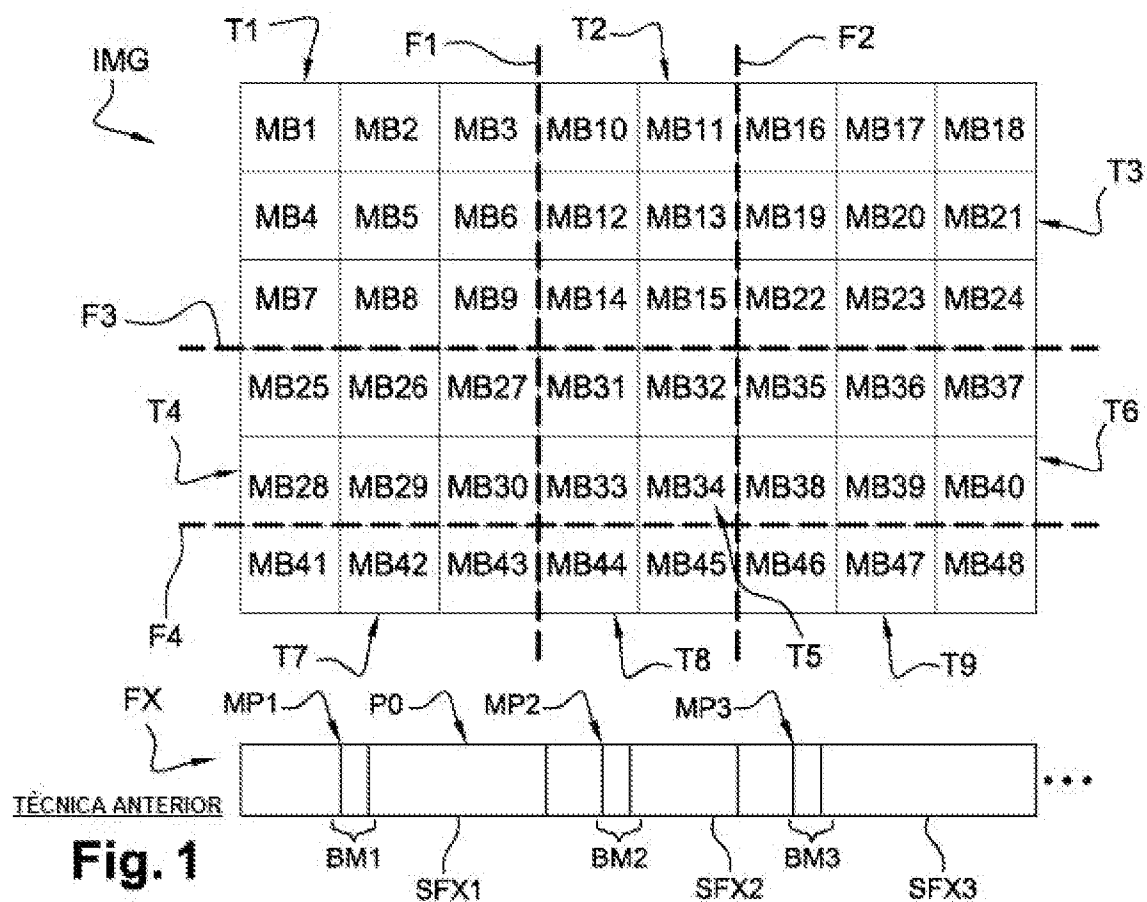
Por otra parte, teniendo en cuenta el hecho de que la unidad de decodificación se reinicializa cada vez que un bloque decodificado recorre un límite entre la tesela que se está decodificando y la tesela que le sigue en orden del "barrido de trama", no es necesario, con vistas a reanudar la decodificación de la segunda línea de bloques de dicha tesela, una vez decodificados los bloques de la primera línea de la tesela siguiente:

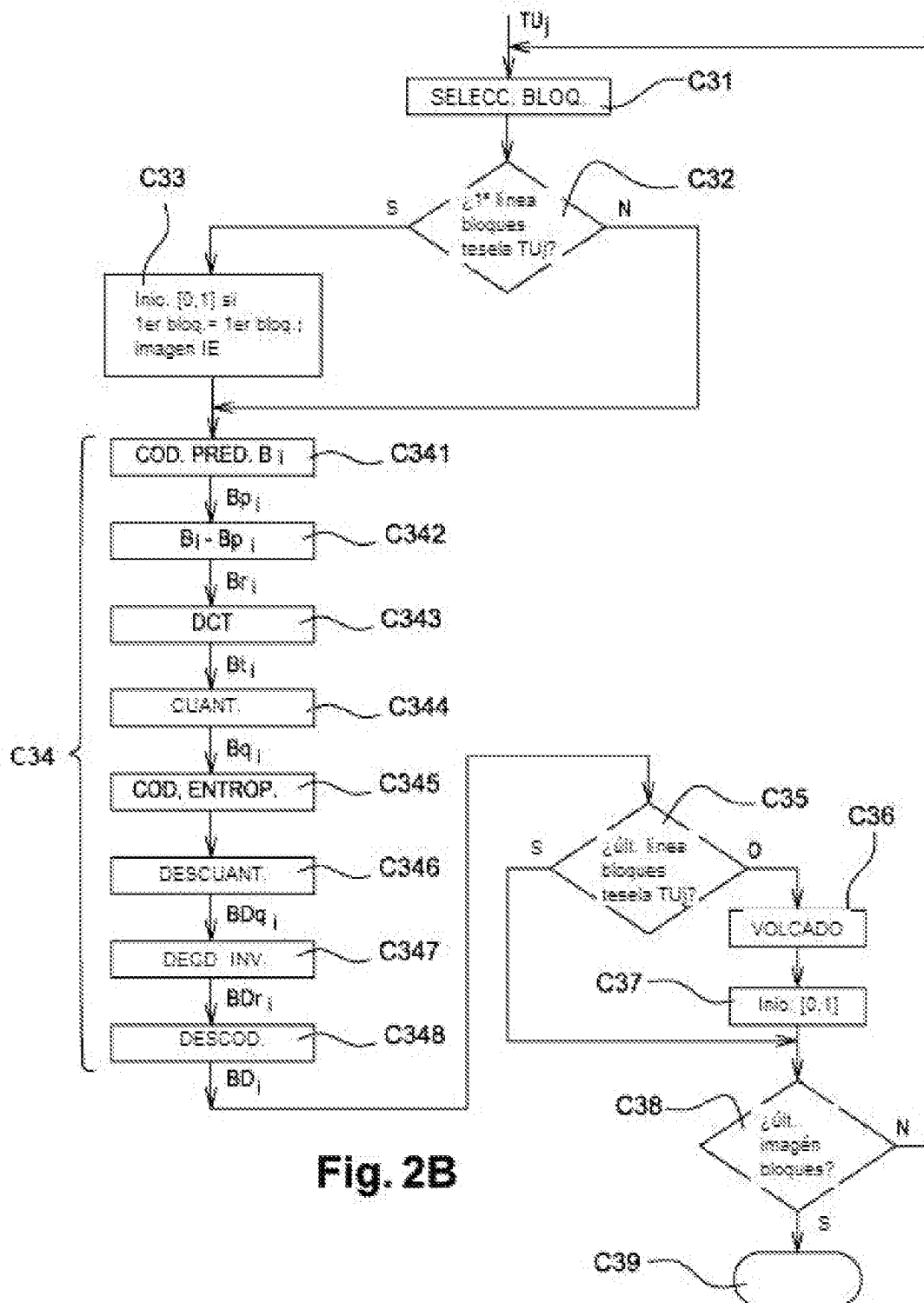
45 • memorizar previamente los bits que han sido compartidos en el curso de la decodificación de la primera línea de bloques de dicha tesela, con vistas a ser utilizados para decodificar los bloques que siguen al último bloque de la primera línea de dicha tesela en orden de barrido de trama,

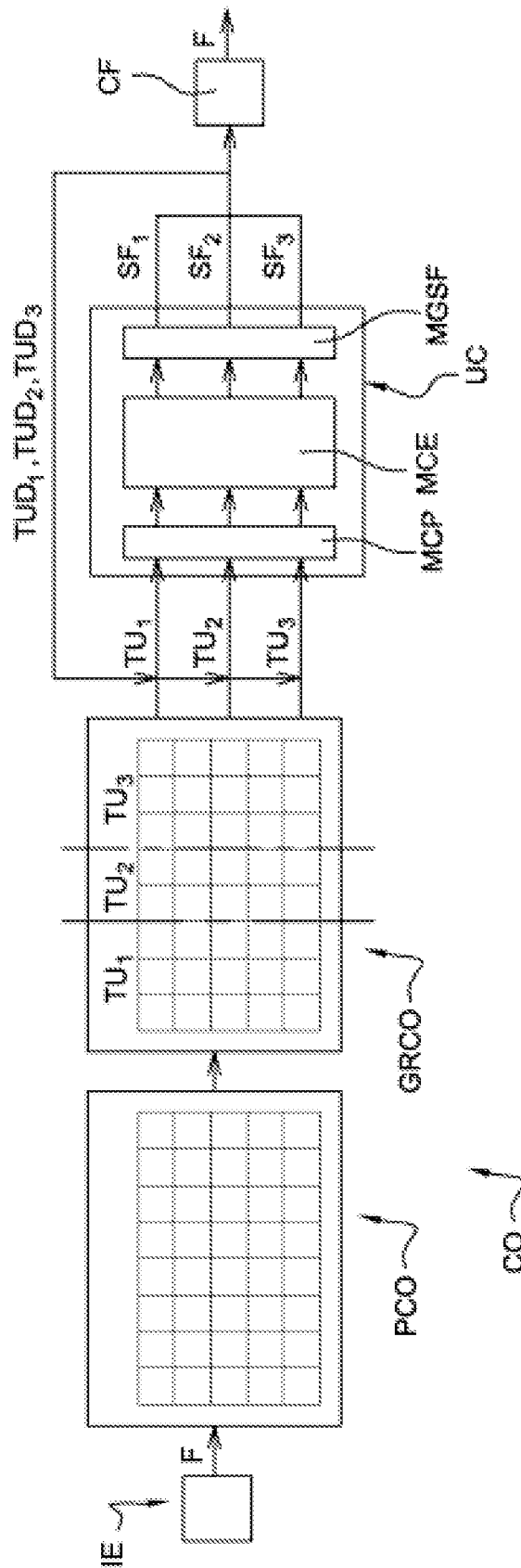
• transferir los bits compartidos a la memoria actual del decodificador entrópico cuando se reanude la decodificación de la segunda línea de bloques en la tesela.

REIVINDICACIONES

1. Método de descodificación de un flujo (F) representativo de al menos una imagen codificada consistente en al menos dos teselas, estando constituida una tesela de varios bloques de al menos dos líneas de la imagen a descodificar ($TU_1, TU_2, \dots, TU_i, \dots, TU_N$) que son adyacentes por pares y han sido agrupados y, a continuación, codificados consecutivamente, pudiendo dichos bloques contener símbolos pertenecientes a un conjunto predeterminado de símbolos, estando dichas al menos dos teselas separadas por un límite vertical, comprendiendo las etapas de:
 - identificación (D1), en dicho flujo, de al menos dos subflujos de datos ($SF_1, SF_2, \dots, SF_i, \dots, SF_N$) correspondientes respectivamente a dichas al menos dos teselas, estando asociada una tesela a un subflujo,
 - descodificación de dichos al menos dos teselas de bloques por medio de un módulo de descodificación entrópica, leyendo, en dichos subflujos identificados, información digital asociada a los símbolos de cada bloque de la tesela correspondiente, comprendiendo dicha etapa de descodificación, para el primer bloque a descodificar de la imagen, una subetapa (D23) de inicialización de variables de estado de dicho módulo de descodificación entrópica, dicho método de descodificación **se caracteriza por que** comprende las siguientes etapas durante la descodificación de los bloques de una línea de la imagen, en cada límite vertical:
 - en el caso (D25, O) en el que el bloque actual de dicha línea es un bloque previamente descodificado e identificado como el último bloque de una línea de una de dichas teselas comprende, además:
 - una etapa (D26) de reinicialización de las variables de estado de dicho módulo de codificación entrópica, y
 - una etapa de memorización en una memoria de una posición del puntero de lectura de dicho subflujo asociado a dicha tesela;
 - en el caso (D23, O) en el que el bloque actual de dicha línea es un bloque identificado como el primer bloque de una línea de una de dichas teselas, comprende además una etapa (D241) de restitución de una posición del puntero de lectura de dicho subflujo asociado con dicha tesela desde dicha memoria.
2. Dispositivo (DO) de descodificación de un flujo (F) representativo de al menos una imagen codificada que consiste en al menos dos teselas, estando constituida una tesela de varios bloques de al menos dos líneas de la imagen a descodificar ($TU_1, TU_2, \dots, TU_i, \dots, TU_N$) que son adyacentes por pares y han sido agrupados y, a continuación, codificados consecutivamente, pudiendo dichos bloques contener símbolos pertenecientes a un conjunto predeterminado de símbolos, estando dichas al menos dos teselas separadas por un límite vertical, que comprende:
 - medios (EXDO) para identificar en dicho flujo un número predeterminado de al menos dos subflujos de datos ($SF_1, SF_2, \dots, SF_i, \dots, SF_N$) correspondientes respectivamente a dichas al menos dos teselas, estando asociada una tesela a un subflujo,
 - medios (UD) para descodificar dichos al menos dos teselas de bloques, comprendiendo dichos medios de descodificación un módulo de descodificación entrópica capaz de leer, en dichos subflujos identificados, información digital asociada a los símbolos de cada bloque de dicha tesela correspondiente a dicho al menos un subflujo identificado, comprendiendo dichos medios de descodificación, para el primer bloque a descodificar de la imagen, submedios de inicialización de variables de estado de dicho módulo de descodificación entrópica, dicho dispositivo de descodificación **se caracteriza por que** durante la descodificación de los bloques de una línea de la imagen, en cada límite vertical:
 - en el caso de que el bloque actual de dicha línea sea un bloque previamente descodificado e identificado como el último bloque de una línea de una de dichos teselas:
 - dichos submedios de inicialización se activan para reinicializar las variables de estado de dicho módulo de descodificación entrópica, y
 - una posición del puntero para leer dicho subflujo asociado a dicha tesela se memoriza en una memoria;
 - en el caso de que el bloque actual de dicha línea sea un bloque identificado como el primer bloque de una línea de uno de dichas teselas, se restituye desde dicha memoria una posición del puntero de lectura de dicho subflujo asociado a dicha tesela.
3. Programa informático que comprende instrucciones para aplicar el método de descodificación de acuerdo con la reivindicación 1, cuando se ejecuta en un ordenador.
4. Medio de grabación legible por ordenador en el que se graba un programa informático que comprende instrucciones para la ejecución de las etapas del método de descodificación de acuerdo con la reivindicación 1, cuando dicho programa se ejecuta en un ordenador.







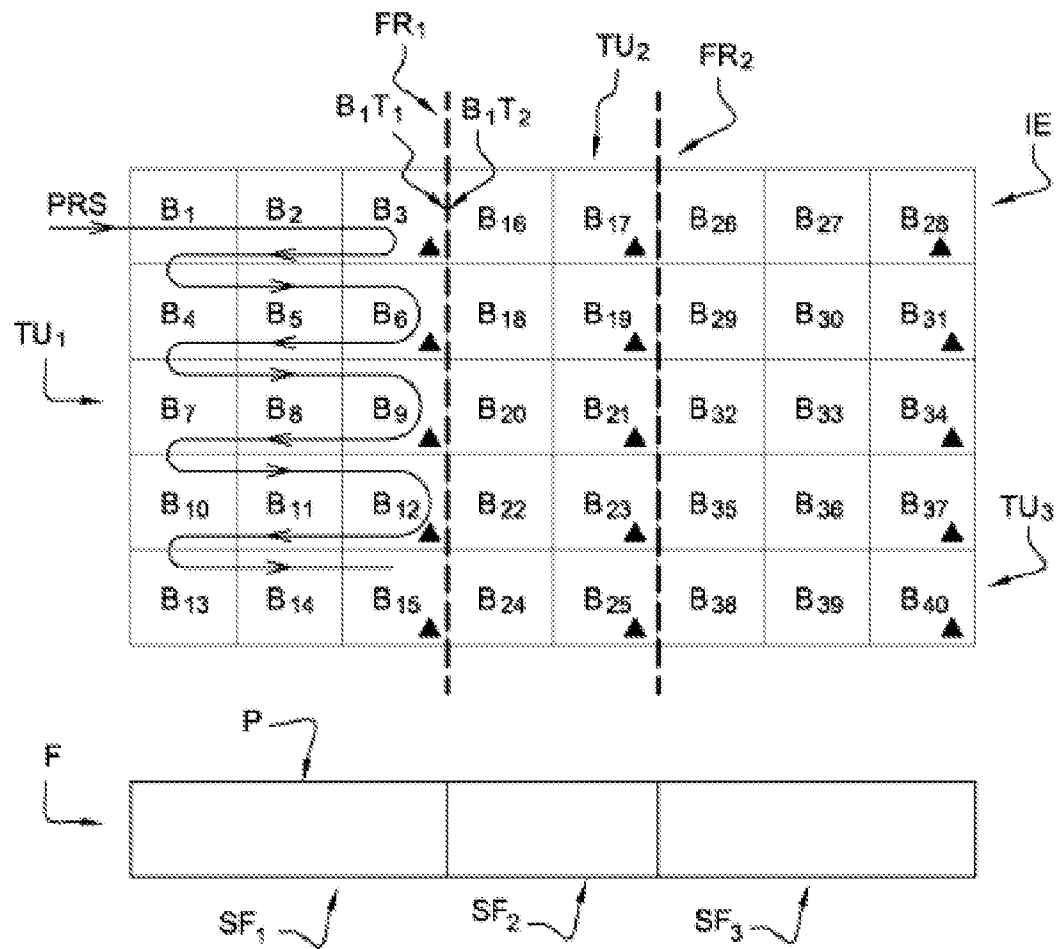


Fig. 4

