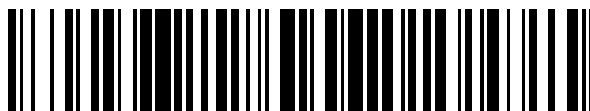


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 793 499**

51 Int. Cl.:

H04N 19/176

(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **16.09.2010** **PCT/JP2010/066621**

87 Fecha y número de publicación internacional: **28.04.2011** **WO11048909**

96 Fecha de presentación y número de la solicitud europea: **16.09.2010** **E 10768295 (7)**

97 Fecha y número de publicación de la concesión europea: **29.04.2020** **EP 2491716**

54 Título: **Método para decodificar un flujo de bits creado utilizando una transformación direccional**

30 Prioridad:

21.10.2009 US 603100

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

16.11.2020

73 Titular/es:

mitsubishi electric corporation (100.0%)
7-3 Marunouchi 2-Chome, Chiyoda-ku
Tokyo 100-8310, JP

72 Inventor/es:

COHEN, ROBERT, A.;
VETRO, ANTHONY y
SUN, HUIFANG

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 793 499 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 decodificar un flujo de bits creado utilizando una transformación direccional

Campo técnico

La invención se refiere, en general, a códecs de video, y, más particularmente, a transformaciones direccionales utilizadas durante la codificación y la decodificación de bloques de píxeles en fotogramas e imágenes de video.

Antecedentes de la técnica

Códecs

Un códec de video digital comprime y descomprime un video. Los códecs se pueden encontrar en equipos de transmisión, televisores, ordenadores personales, grabadoras y reproductores de video, satélites, así como dispositivos móviles y en línea. Los códecs dividen cada fotograma del video en bloques de píxeles y procesan los bloques uno a uno.

Durante la codificación, se eliminan las redundancias espaciales y temporales para reducir la velocidad de datos. La invención se refiere, particularmente, a las transformadas que se utilizan durante la codificación y decodificación de videos. La transformada más común es una transformada discreta de coseno (DCT – Discrete Cosine Transform, en inglés) tal como se especifica en los estándares MPEG y H.264/AVC. La DCT convierte las intensidades de píxeles en el dominio del espacio para transformar los coeficientes en el dominio de la frecuencia. Los coeficientes son cuantificados y codificados por entropía, para producir un flujo de bits comprimido. El flujo de bits puede ser almacenado en un medio (DVD) o comunicado directamente al decodificador. Durante la decodificación, las etapas se invierten. Después de la decodificación por entropía y la cuantización inversa, se aplica una transformada inversa para recuperar el video original.

En general, el número de decodificadores, por ejemplo, productos de consumo en todo el mundo, supera con creces el número de codificadores. Por lo tanto, para permitir la interoperabilidad, solo el flujo de bits y el proceso de decodificación están estandarizados. El proceso de codificación no se especifica en absoluto, en general, en un estándar.

En la publicación de CHUO-LING CHANG *ET AL* "Direction-adaptive partitioned block transform for image coding", Image Processing 2008, ICIP 2008, 15ª Conferencia Internacional del IEEE, Piscataway, NJ, EE.UU., páginas 145-148; se propone la transformada de bloque de dirección adaptativa para mejorar el rendimiento de la compresión. Como transformada de bloque, puede ser combinada directamente con intra e inter-predicción de bloques en los estándares de codificación de video.

Transformadas

La DCT incluye una DCT 1-D horizontal aplicada a cada fila de píxeles en el bloque y una DCT 1-D vertical aplicada a cada columna. Para bloques con características predominantemente horizontales o verticales, la DCT 2-D es eficiente. Sin embargo, la DCT 2-D no transforma eficientemente los bloques que contienen características que *no* son horizontal o vertical, es decir, características direccionales, donde direccional se refiere a orientaciones distintas a la horizontal y la vertical.

En general, existen dos métodos que implementan transformadas direccionales. El primer método aplica las DCT 2-D a lo largo de rutas predefinidas dentro del bloque. El segundo método aplica un filtro direccional, seguido de la DCT 2D. Típicamente, un filtro de ventilador divide el bloque en un conjunto de subbandas direccionales. Las transformadas son aplicadas posteriormente a cada subbanda. Las transformadas direccionales, como los contornos, son implementadas de esta manera. Los contornos transforman eficientemente los fotogramas que contienen regiones suaves separadas por límites curvos.

Se han utilizado transformadas direccionales para complementar la transformada DCT 2-D o similar a DCT existente para los métodos de codificación de video existentes, tales como H.264/AVC. Durante el proceso de codificación, el codificador H.264/AVC selecciona de entre un conjunto de transformadas, como la transformada 2-D convencional, y un conjunto de transformadas direccionales. La transformada única que produce el mejor rendimiento, en un sentido de velocidad/distorsión, es seleccionada para la codificación y la decodificación.

Después de la transformación, se pueden realizar mejoras en la codificación por entropía de los datos correspondientes, aprovechando las estadísticas de los datos direccionales. En el H.264/AVC, se utiliza un codificador aritmético binario adaptativo al contexto (CABAC – *Context Adaptive Binary Arithmetic Coder*, en inglés) o un codificador de longitud variable adaptativo al contexto (CAVLC – *Context Adaptive Variable Length Coder*, en inglés) para codificar por entropía diferentes tipos de datos. Los símbolos de entrada son asignados a palabras de código binario y son comprimidos mediante un codificador aritmético. Los contextos son utilizados para adaptar las estadísticas utilizadas por el codificador aritmético. Cada contexto almacena el símbolo más probable (0 o 1) y la probabilidad correspondiente.

El estándar H.264/AVC está diseñado para utilizar la DCT 2-D. Los métodos existentes pueden utilizar transformadas direccionales para extender el rendimiento de los codificadores H.264/AVC. Sin embargo, esos métodos aún generan y codifican las decisiones y los datos relacionados con la dirección utilizando el fotograma H.264/AVC convencional. Por lo tanto, existe la necesidad de representar eficientemente la información direccional, así como la necesidad de mejorar las eficiencias de codificación.

El propósito de la transformada es convertir un bloque de valores de píxeles variables en un bloque de coeficientes para los cuales la mayoría de los coeficientes son cero. En el caso de la DCT, un conjunto de píxeles de matriz se convierte en un conjunto de coeficientes DCT que representan datos de baja frecuencia a alta frecuencia en el bloque. La frecuencia más baja es el coeficiente DC, que está relacionado con el valor promedio de todos los píxeles convertidos mediante las transformadas. El siguiente coeficiente representa la magnitud de la onda cosenoidal de frecuencia más baja que está contenida en la señal. Los coeficientes posteriores corresponden a frecuencias crecientes. Si los datos son adecuados para la DCT, muchos de los coeficientes de frecuencia son cero y el decodificador no los necesita para reconstruir el video.

Un problema con las transformadas direccionales existentes, que utilizan un conjunto de transformadas 1-D paralelas, es que la longitud de cada transformada 1-D puede variar dependiendo de la posición de la transformada en el bloque. Por ejemplo, para transformar un bloque de 8x8 utilizando una transformada direccional orientada a 45 grados, la transformada 1-D a lo largo de la diagonal principal del bloque tiene ocho elementos, y la 1-D adyacente tiene siete elementos, y así sucesivamente para transformadas de uno o dos elementos, que son ineficientes. Una transformada de un elemento es, en el mejor de los casos, una escala de un valor de píxel, que mejora poco la eficiencia de la codificación. Por lo tanto, existe la necesidad de un método para transformar bloques utilizando estas transformadas de una manera que no sufra las ineficiencias mostradas por las pequeñas rutas de transformada, pero que aún mantengan las propiedades direccionales de la transformada original.

También existe la necesidad de un método para aplicar un segundo conjunto de transformadas a la salida del primer conjunto de transformadas de una manera que mejore aún más la eficiencia de codificación sin degradar el rendimiento de la forma en que lo haría una DCT 2-D cuando los datos no están correlacionados en la segunda dirección ortogonal.

Además, existe la necesidad de versiones divididas de esta transformada que sean adecuadas para codificar bloques residuales de predicción que se encuentran comúnmente en codificadores predictivos tales como el H.264/AVC.

Compendio de la invención

El problema se resuelve mediante el método para decodificar un flujo de bits según la reivindicación 1. En particular, el flujo de bits incluye una secuencia de fotogramas. Cada fotograma está dividido en bloques codificados. Para cada bloque, se determina un conjunto de rutas en un ángulo de transformada determinado a partir de un índice de transformada en el flujo de bits. Los coeficientes de transformada se obtienen del flujo de bits. Los coeficientes de transformada incluyen un coeficiente DC para cada ruta. Se aplica una transformada inversa a los coeficientes de transformada para producir un video decodificado.

Breve descripción de los dibujos

La figura 1A es un diagrama de bloques de un sistema de video;

la figura 1B es un diagrama de bloques de un decodificador;

la figura 1C es un diagrama de bloques de un codificador;

la figura 2 es un diagrama de bloques de un módulo de procesamiento direccional de subbloques y particiones;

la figura 3 es un diagrama de bloques de un tipo de transformada y un módulo de decisión de dirección;

la figura 4 es un diagrama de bloques del módulo de inferencia de dirección;

la figura 5 es un diagrama de bloques de un módulo de predicción direccional;

la figura 6 es un diagrama de bloques de un módulo de codificación de índice direccional;

la figura 7 es un esquema de un primer ejemplo del módulo de codificación direccional de índice;

la figura 8 es un esquema de un segundo ejemplo del módulo de codificación direccional de índice;

las figuras 9-10 son diagramas de flujo para un módulo de generación de contexto;

las figuras 11A-B son esquemas de rutas para transformadas direccionales primarias y secundarias;

las figuras 12A-F son diagramas de bloques para transformadas direccionales principales para bloques de píxeles de 8x8; y

la figura 13 es un diagrama de bloques de un conjunto de transformadas direccionales divididas.

Descripción de ejemplos

Códec

La figura 1A muestra un sistema de video. El sistema incluye un codificador 10 y un decodificador 20, en combinación, un códec 30. El códec puede ser implementado en procesadores que incluyen memorias e interfaces de entrada/salida, tal como es conocido en la técnica.

El codificador comprime un video de entrada 1 en un flujo de bits 15. El codificador aplica una codificación por transformación, cuantificación y entropía al video de entrada, tal como se describe en detalle a continuación. Para garantizar que el video de salida refleje con precisión el video de entrada, el decodificador 20 realiza las etapas inversas en un orden inverso. Además, el codificador incluye, típicamente, el equivalente del decodificador, para proporcionar una retroalimentación para el proceso de codificación. Debido a que todas las variables del codificador están fácilmente disponibles en el codificador, el decodificador en el codificador es relativamente simple. La invención se refiere, particularmente, a las transformadas direccionales inversas 25.

Tal como se describe a continuación, y se muestra en la figura 11B, las transformadas pueden incluir transformadas principales y secundarias. Durante la codificación, la transformada principal actúa sobre las intensidades de los píxeles para determinar los coeficientes de transformada, por ejemplo, los coeficientes CC y CA. La transformada secundaria actúa solo sobre los coeficientes DC 1160 para producir coeficientes de transformada secundaria 1170, para reducir aún más las redundancias de datos. Las transformadas inversas incluyen una transformada inversa secundaria 26 para reconstruir los coeficientes de CC, y las transformadas inversas principales 27 recuperan las intensidades de píxeles para el video decodificado.

Para garantizar la interoperabilidad entre el codificador y el decodificador, los estándares de codificación de video generalmente especifican solo el flujo de bits y el proceso de decodificación. Sin embargo, se entiende que una descripción del proceso de codificación, tal como se detalla a continuación, es suficiente para deducir exactamente el proceso de decodificación inversa por parte de un experto en la técnica.

Decodificador

La figura 1B muestra partes relevantes de nuestro decodificador 20. El decodificador recibe el flujo de bits 15 de codificación, y la información 160. El flujo de bits es presentado a un decodificador por entropía CABAC 191, que produce coeficientes de transformada cuantificados 192 de acuerdo con la información. Para el primer bloque, la información puede ser un contexto inicial. Posteriormente, la información es relacionada con bloques previamente procesados (decodificados).

Los coeficientes son cuantificados inversamente 24 y transformados inversamente 25, de manera que los bloques decodificados formen la salida o el video decodificado 2. La transformada puede ser una transformada discreta inversa de coseno (IDCT – *Inverse Discrete Cosine Transform*, en inglés). Las transformadas pueden incluir una transformada discreta inversa de coseno 2D y un conjunto de transformadas direccionales inversas. Las transformadas inversas secundarias también pueden ser aplicadas tal como se describe a continuación con mayor detalle.

La información 160 es presentada al módulo de generación de contexto (CGM – *Context Generation Module*, en inglés) del decodificador, que reenvía los contextos seleccionados 921-922 al decodificador CABAC. Los indicadores de transformada predichos (PTI – *Predicted Transform Indicators*, en inglés) 501 de los bloques 160 decodificados previamente son presentados a un módulo de decodificación direccional de índice (DIDM – *Directional Index Decoding Module*, en inglés) 601, que genera un indicador de transformada 602 para la transformada inversa 25. La transformada inversa puede utilizar cualquiera de las transformadas inversas, por ejemplo, DCT inversas horizontal 1D y vertical 1D (IDCT 2D) 41, un conjunto de transformadas direccionales inversas 42 y cualquier otra transformada inversa 43 conocida.

Se observa que los estándares de codificación de video actuales solo utilizan una única transformada especificada previamente, por lo que no se necesita un índice para diferentes transformadas. Además, los estándares actuales no consideran la información secundaria relacionada con bloques previamente decodificados durante la transformación inversa.

Codificador

La figura 1C muestra los detalles relevantes del codificador 10. El codificador utiliza transformadas direccionales. Las etapas del método que se muestran pueden ser realizadas en un procesador de un codificador. El procesador incluye interfaces de memoria y entrada/salida, tal como se conoce en la técnica.

La entrada al codificador es un bloque 101 de un fotograma de un video a codificar. Tal como se define en el presente documento, los bloques incluyen macrobloques, subbloques y particiones de bloque, en general, una matriz de píxeles. En la mayoría de las aplicaciones de codificación, las operaciones son realizadas preferiblemente sobre macrobloques

y subbloques. El bloque puede contener datos de video originales, residuos de una predicción espacial o de movimiento compensado de datos de video u otros datos a transformar relacionados con la textura. El bloque puede ser dividido en subbloques mediante un módulo de procesamiento direccional de partición de subbloques (SPDPM – *Sub-block Partition Directional Processing Module*, en inglés) 200. En el presente documento, los subbloques son procesados uno a uno como “bloques”.

Cada bloque es transformado utilizando transformadas seleccionadas de entre una transformada discreta de coseno bidimensional (DCT 2-D) 120 convencional, un conjunto de transformadas direccionales 130 u otras transformadas, en general, transformadas 125. La salida de la transformada es medida por un módulo de decisión de tipo de transformada y de dirección (TTDDM – *Transform Type and Direction Decision Module*, en inglés) 300. El TTDDM utiliza una métrica, tal como un coste de velocidad/distorsión, para determinar cuál de las transformadas proporciona el mejor rendimiento. El coste de la velocidad/distorsión es una suma de una velocidad de codificación y un escalar multiplicado por la distorsión. El tipo de transformada y la dirección tienen un coste mínimo seleccionado para la transformada. El rendimiento puede ser, pero sin estar limitado a, una medida de la eficiencia de la codificación. La idea es que la transformada que tiene el mejor rendimiento sea seleccionada para la codificación, y la transformada seleccionada es indicada al decodificador como en un índice 16 en el flujo de bits.

El TTDDM también puede recibir información de un módulo de inferencia de dirección (DIM – *Direction Inference Module*, en inglés) 400. La entrada al DIM es una recogida de datos 160 que indican las transformadas y direcciones utilizadas para bloques adyacentes procesados previamente. La salida del DIM es un valor o conjunto de valores correspondientes a los datos 160, tales como las direcciones 431 preferidas. El TTDDM utiliza esta información para tomar la decisión sobre qué transformadas y direcciones utilizar para codificar el bloque 101. El TTDDM puede reenviar, asimismo, un indicador de partición final (FPI – *Final Partitioning Indicator*, en inglés) 141 al SPDPM como guía para la partición. El módulo TTDDM produce el bloque transformado 102 y un indicador de transformada seleccionada (STI – *Selected Transform Indicator*, en inglés) 145 que representa la transformada y dirección seleccionadas.

A continuación, el bloque transformado 102 puede ser codificado 150 de manera apropiada utilizando codificación por entropía, para producir un bloque de salida codificado 17.

El módulo de predicción de dirección (DPM – *Direction Prediction Module*, en inglés) 500 también recibe información del DIM e información relacionada con los bloques 160 procesados previamente. El DPM utiliza esta información para generar un indicador de transformada predicho (PTI) 501. El PTI es introducido en un módulo de codificación direccional de índice (DIEM – *Directional Index Encoding Module*, en inglés) 600, junto con el STI 145. El DIEM convierte la representación en una palabra de código 603 binaria para codificar mediante un codificador aritmético binario adaptativo al contexto (CABAC) 190.

Los contextos utilizados por el CABAC están determinados por un módulo de generación de contexto (CGM) 900. La entrada al CGM es información sobre las transformadas y direcciones utilizadas por bloques adyacentes previamente codificados desde el DIM, o información ya codificada desde el bloque actual. El CGM produce contextos para que el CABAC codifique el índice direccional binario. El CABAC genera un índice de transformada 16 codificado.

Módulo de procesamiento direccional de subbloques y particiones

La figura 2 muestra detalles del SPDPM 200. Los píxeles en el bloque de entrada 101 pueden representar información relacionada con el video, tal como datos de fotogramas de video, residuos de predicción con compensación de movimiento y residuos de predicción espacial. El SPDPM divide el bloque en particiones 210, en general, conjuntos de píxeles. Las transformadas convencionales o direccionales 125 son aplicadas a las particiones. El indicador de partición 141 final producido por el TTDDM indica qué particiones utilizar para el mejor rendimiento.

Tipo de transformada y módulo de decisión de dirección

La figura 3 muestra el TTDDM 300 para seleccionar la mejor transformada y dirección a utilizar para transformar el bloque 210. Un selector de transformada 310 elige cuál de los tipos de transformada disponibles será enviado al módulo de medición 320, que determina una métrica 321, tal como la velocidad/distorsión (R/D – *Rate/Distortion*, en inglés), que se utiliza para seleccionar la transformada.

El selector de transformada puede verse influenciado por el DIM 400. El DIM, por ejemplo, puede examinar bloques adyacentes para determinar qué direcciones tienen más probabilidades de comportarse bien para el bloque actual. La medición puede estar limitada a un subconjunto de direcciones disponibles, reduciendo de este modo el tiempo de procesamiento. Después de que estas mediciones son utilizadas para determinar la mejor dirección o transformada, se envían el indicador de transformada 145 seleccionado y el bloque transformado 102 correspondiente. Si el TTDDM está actuando sobre una selección de particiones, entonces el indicador de partición 141 final que produce el mejor rendimiento también enviado al SPDPM.

Módulo de inferencia de dirección

La figura 4 muestra el DIM 400. El módulo de selección de bloque utiliza los bloques procesados previamente y la información 160 lateral para determinar las posibles direcciones de transformada 411 para el bloque actual. Las posibles direcciones de transformada son utilizadas para determinar un conjunto de direcciones 431 preferidas. Posteriormente, el DPM utiliza esto para reducir el número de bits necesarios para representar esta información, lo que da como resultado una mayor eficiencia en el codificador y el decodificador.

Un módulo de selección de bloque (BSM – *Block Selection Module*, en inglés) 410 realiza una selección de entre los bloques 160 en base a criterios, tales como la distancia de los bloques seleccionados al bloque actual. El módulo de decisión de fiabilidad (RDM – *Reliability Decision Module*, en inglés) 420 estima la fiabilidad de los bloques seleccionados. El módulo RDM puede utilizar información de la textura, la posición y otros datos del bloque 412. Un factor de fiabilidad 421 de cada uno de los bloques seleccionados, y la dirección de la transformada 411 correspondiente son proporcionados al módulo de determinación de dirección preferencial (PDDM – *Preferential Direction Determination Module*, en inglés) donde las direcciones 431 preferidas son identificadas.

Módulo de predicción direccional

La figura 5 muestra el DPM 500 para determinar el indicador de transformada 501 previsto para el DIEM 600 y el CGM. Un primer predictor de etapa 510 selecciona candidatos 515 de las direcciones 431 preferidas. Un segundo predictor de etapa 420 utiliza estos candidatos y la información 160 lateral codificada para seleccionar el indicador de transformada 501 preferido.

Para codificar residuos de textura transformados, el indicador 145 de transformada seleccionado puede ser correlacionado con un predictor de textura, tal como un modo de intra-predicción utilizado en H.264/AVC. Por lo tanto, la información secundaria proporcionada al DPM puede incluir, por ejemplo, el modo intra-predicción para seleccionar el indicador 501.

Módulo de codificación direccional de índice

La figura 6 muestra el DIEM 600. La entrada incluye el indicador de transformada 145 seleccionado y el indicador de transformada 501 previsto, que son asignados 605-606 a una representación significativa de las direcciones. Se pueden utilizar diferentes asignaciones 605-606 para los indicadores de transformada seleccionados y los predichos. Una diferencia entre las dos direcciones se determina 610 como una diferencia de indicadores de transformada 612. Debido a que la predicción es una aproximación razonable de la dirección de transformada seleccionada, las pequeñas diferencias de ángulo deberían dar como resultado palabras de código similares que pueden ser codificadas de manera efectiva. La diferencia es binarizada 620 a la palabra de código 603, que es codificada por el CABAC 190 como el índice de transformada 16 codificado. Se comprende que se puede utilizar cualquier codificador por entropía adaptativo al contexto, así como la codificación de longitud variable (VLC - *Variable Length Coding*, en inglés). El cálculo de la diferencia puede ser omitido 611 tal como se describe a continuación.

La figura 7 muestra esquemáticamente un primer ejemplo del DIEM 600. Por ejemplo, hay ocho posibles direcciones de transformada 701 y las predicciones 702 correspondientes. La dirección de transformada es seleccionada mediante el indicador de transformada 145 seleccionado, y la predicción es seleccionada mediante el PTI 501. Los indicadores de transformada son asignados a un código gris, en el que las direcciones adyacentes difieren solo en un bit. Las palabras de código para las direcciones seleccionadas y predichas son comparadas bit a bit mediante una operación O-exclusiva (XOR) 610 para obtener la diferencia 611. Para un predictor preciso, esto produce un flujo de bits con ceros en su mayoría para una baja entropía. Debido a que las asignaciones de indicadores 605-606 utilizan representaciones binarias, la binarización 620 no se utiliza.

La figura 8 muestra un segundo ejemplo del DIEM. En este ejemplo, las direcciones están representadas por una secuencia de números uniformemente continua. La diferencia 610 es

$$\Delta = (I_S - I_P + N) \bmod N,$$

donde I_S e I_P son los índices asignados de los indicadores de dirección seleccionados y predichos, respectivamente, y N es el número de dirección posible, por ejemplo, ocho. Debido a que las pequeñas diferencias son más probables, la binarización 620 codifica diferencias cercanas a cero (0, 1, $N-1$, 2, $N-2$, ...) con menos bits. El cálculo de la diferencia puede ser omitido 611 y el indicador de transformada asignado es reenviado directamente al módulo de binarización 620. En este caso, el módulo de generación de contexto 900 utiliza el indicador de transformada previsto para seleccionar un contexto apropiado.

Módulo de generación de contexto

Las figuras 9-10 muestran ejemplos del CGM 900. El CGM selecciona los contextos 921-922 para el CABAC 190. También se pueden seleccionar más de dos contextos. Para determinar los contextos, el CGM puede utilizar la información 160 de bloque procesada preferida, el PTI 501 y las direcciones 431 preferidas. Los contextos A y B

- distinguen direcciones predichas precisas y predicciones inexactas. La figura 9 muestra cómo se utilizan las direcciones 431 preferidas para determinar los contextos. Se determina una diferencia máxima ϕ 910 y se compara 920 con un umbral predeterminado T . Si la diferencia es menor que el umbral, entonces la predicción es precisa en el contexto A 921; de lo contrario, el contexto B 922 es inexacto. Por ejemplo, si se utiliza el DIEM, los bits proporcionados a CABAC son en su mayoría ceros, y el contexto A es seleccionado para ajustarse a esta probabilidad. La selección de contexto del CGM 900 también puede considerar otros factores, tales como la posición del bit, para decidir entre más de dos contextos.
- Los ejemplos que se muestran en la figura 10 suponen que el DIEM omite 611 el cálculo de la diferencia 610. El indicador de transformada 501 predicho y un índice de posición i 1001 que representan qué bit del índice 603 será codificado, son entradas. El PTI 501 es asignado 1010 a una palabra de código binaria con el mismo indicador de asignación 605 utilizado en el DIEM. Debido a que ambas palabras de código deberían ser iguales, el bit más probable del CABAC debería ser el mismo que el bit actual CW $[j]$ de 1030. De este modo, si la comparación 1030 indica que el bit actual es 1, se selecciona el contexto A 921, que prefiere 1, de lo contrario se selecciona el contexto B 922, donde 0 es el bit preferido.
- Transformadas direccionales principal y secundaria
- La figura 11A muestra la transformada direccional según las realizaciones de nuestra invención. Se comprende que, durante la decodificación, la transformada es una transformada *inversa*. La transformada (o su inversa) actúan sobre un bloque 1100 de $M \times N$ píxeles. Para el propósito de la transformada, los valores 1101 asociados con los píxeles pueden ser intensidades o coeficientes de transformada, dependiendo de la etapa de codificación o decodificación.
- La transformada incluye un conjunto de transformadas 1-D $\{T_0, T_1, \dots, T_{N-1}\}$ 1102, donde N es el número total de transformadas 1-D aplicadas al bloque. La longitud l de la transformada T_l indica el número de píxeles sobre los que actúa la transformada 1-D. Por lo tanto, las transformadas $\{T_0, T_1, \dots, T_{N-1}\}$ en el conjunto tienen longitudes correspondientes $\{l_0, l_1, \dots, l_{N-1}\}$.
- Cada transformada se aplica a los píxeles a lo largo de una ruta 1102 en el bloque. Una ruta incluye, en general, un conjunto de píxeles contiguos o adyacentes.
- Tal como se muestra en la figura 11B, si los valores de los píxeles en la ruta son coeficientes, entonces el primer coeficiente para cada ruta es el coeficiente DC, seguido de los coeficientes AC. Todos los primeros coeficientes 1150 son recogidos en un conjunto 1170 de coeficientes secundarios de transformada (STC – Secondary Transform Coefficients, en inglés). El primer coeficiente secundario en el conjunto es un coeficiente secundario DC (SDC – Secondary DC, en inglés).
- Durante la decodificación, la transformada direccional secundaria 26 puede ser aplicada al conjunto de coeficientes de transformada secundarios para reconstruir el coeficiente DC de cada ruta. Durante la codificación, el primer coeficiente DC o de cada ruta es descartado después de que se forma el conjunto 1170 de coeficientes de transformada secundarios.
- Cada ruta está orientada, con respecto a la dirección vertical 1103, en un ángulo θ 1105 de transformada, que está determinado por el índice de transformada direccional. Tal como se describió anteriormente, el índice de transformada, que es determinado durante la codificación, forma parte del flujo de bits a decodificar.
- Las rutas para un bloque y transformada en particular están orientados, en general, en la misma dirección. Las rutas se generan de la siguiente manera.
- Una ruta mínima de longitud de transformada L_{min} 1110 se especifica para el bloque. La ruta de transformada habitualmente comienza en un píxel inicial 1120 ubicado en un borde (o esquina) del bloque 1100. Tal como se indicó anteriormente, si los valores para los píxeles son coeficientes, el valor para el píxel inicial es el coeficiente DC.
- La ruta continúa a lo largo del ángulo θ hasta un píxel final 1121 en otro borde. La longitud de la ruta, en píxeles, es m . Si el píxel inicial está en un borde o esquina, entonces la longitud $m = 1$.
- Si $m \geq L_{min}$, entonces la ruta se considera completa. Si el bloque todavía contiene píxeles que no están en una ruta, se inicia una nueva ruta. La nueva ruta puede comenzar en cualquier píxel no transformado en el bloque. Habitualmente, la siguiente ruta comienza en un píxel adyacente o cerca del inicio de la ruta anterior, o, la ruta puede comenzar en una esquina opuesta del bloque para que la distribución de las longitudes de las rutas dentro del bloque sea sustancialmente simétrica. El proceso continúa con la etapa 2 hasta que se transformen todos los píxeles.
- Si $m < L_{min}$, entonces la ruta es demasiado corta y el proceso continúa mediante la inclusión de un píxel adyacente a un píxel procesado previamente. Si hay más de un píxel adyacente, se utilizan otras rutas en el bloque para determinar la ruta actual. Si hay un píxel sin procesar disponible, el píxel forma parte de la ruta y la longitud m de la ruta se incrementa, y el proceso continúa con la etapa 2 en la dirección $(180 - \theta)$ hasta llegar al borde del bloque. Por lo tanto, la dirección θ es establecida en $(180 - \theta)$, de manera efectiva, un giro en U, antes de continuar con la etapa 2.

Las figuras 12A-C muestran transformadas direccionales para $L_{min} = 3$ y ángulos de 45, 30 y 90 grados, respectivamente. Las figuras 12D-F muestran transformadas direccionales para $L_{min} = 5$.

La transformada inversa 26 actúa a lo largo de las mismas rutas que se describieron anteriormente, excepto por que se utilizan coeficientes de transformada inversa. Por ejemplo, si se utiliza una DCT 1D para cada ruta, la transformada inversa utilizaría la DCT inversa 1-D (IDCT).

Transformadas direccionales secundarias

La figura 11B muestra la transformada direccional secundaria 26. Después de que se haya aplicado la transformada direccional principal, la transformada secundaria puede ser aplicada adaptativamente a cada bloque a lo largo de una ruta que contiene los coeficientes iniciales de cada ruta de la transformada inicial. Estos coeficientes corresponden, habitualmente, a los coeficientes DC de una transformada 1-D, tal como la DCT.

La transformada secundaria inversa actúa a lo largo de la misma ruta descrita anteriormente, excepto por que se utilizan coeficientes de transformada secundaria inversa. La transformada direccional secundaria inversa es ejecutada antes de la transformada direccional inversa durante la decodificación.

La transformada secundaria reduce aún más la redundancia en los componentes DC de los coeficientes de transformada direccional. Alternativamente, se puede utilizar el componente DC de una transformada direccional para predecir el componente DC de otra transformada direccional.

Transformadas direccionales divididas

La figura 13 muestra una transformada direccional dividida. Un bloque de $M \times N$ (8×8) 1300 de píxeles está dividido en dos particiones de bloque A y B aproximadamente a lo largo de una línea 1310 con un ángulo de partición φ con respecto a la vertical. El ángulo de partición es perpendicular al ángulo θ 1105 de transformada.

A continuación, las transformadas direccionales son generadas utilizando las etapas descritas anteriormente, con la restricción de que la partición A del bloque tiene un conjunto de rutas orientadas en un ángulo θ_A y la partición B tiene un conjunto si las rutas están orientadas en un ángulo θ_B . El ángulo principal θ de la transformada direccional se considera igual a θ_A . Puesto que una ruta está generada dentro de la partición, la línea 1310 se aproxima al borde de la partición. De este modo, cada píxel en la transformada direccional está en la partición A o B. Los ángulos θ_A y θ_B pueden ser diferentes.

En un ejemplo, la transformada direccional secundaria es aplicada a ambas particiones B. Para invertir el proceso, se aplica la transformada inversa secundaria y, a continuación, se aplican las transformadas direccionales inversas independientemente a las particiones A y B. En otro ejemplo, la transformada secundaria se aplica de manera independiente a cada partición.

Es decir, ambas transformadas secundarias inversas pueden ser aplicadas antes que las transformadas principales inversas, o la transformada secundaria inversa y la transformada principal inversa pueden ser aplicadas a las particiones de manera independiente. Esta decisión se toma de manera adaptativa, por cada bloque.

Orden de escalado y cuantificación

Después de que se han completado las transformadas principal y secundaria, los coeficientes resultantes son escalados, ordenados y cuantificados.

El escalado de los coeficientes de transformada depende de la longitud m de la ruta de cada transformada direccional 1-D, o de una ubicación del coeficiente en el bloque. Una transformada 1-D de longitud m tiene un factor de escalado S_m . De este modo, todos los coeficientes en la ruta de longitud m son escalados, mediante S_m . Habitualmente, el factor de escalado es seleccionado para que las magnitudes de los coeficientes DC sean las mismas cuando se transforman valores de píxeles idénticos. Si una transformada con longitud $m = 4$ transforma cuatro píxeles, cada uno con valor v , y una transformada con longitud $m = 5$ transforma cinco píxeles, cada uno con valor v , entonces el factor de escalado S_m es seleccionado de modo que ambas transformadas generen los primeros coeficientes (DC) con el mismo valor.

También son posibles métodos de escala alternativos. Transformadas más cortas pueden recibir factores de escala más pequeños o más grandes en base a la longitud m , o la dirección θ . El escalado también se puede hacer parte de la transformación en sí, para simplificar la implementación de este proceso.

Los coeficientes escalados están dispuestos en un orden de escaneo. En un ejemplo, el conjunto de transformadas $\{T_0, T_1, \dots, T_{N-1}\}$ es escaneado de manera independiente y en orden. En cada transformada, el primer coeficiente DC se escanea primero, seguido de los siguientes coeficientes AC en el mismo orden que la ruta para esa transformada.

En otro ejemplo, se escanean todos los primeros coeficientes DC de cada transformada, seguidos de todos los segundos coeficientes de cada transformada, y así sucesivamente, hasta la última transformada. En el segundo ejemplo, el orden en que se escanean las transformadas puede variar. Por ejemplo, las transformadas pueden ser escaneadas en orden de índice, es decir, el primer escaneo utiliza los coeficientes DC del conjunto de transformadas

$\{T_0, T_1, \dots, T_{N-1}\}$ en el orden $\{0, 1, \dots, N-1\}$. Alternativamente, las transformadas se pueden escanear en orden de longitud $\{l_0, l_1, \dots, l_{N-1}\}$. Los coeficientes pueden ser escaneados en base a su posición relativa en el bloque. Por ejemplo, todos los coeficientes a lo largo de un borde del bloque pueden ser escaneados en primer lugar, seguidos de los coeficientes que están desplazados del borde.

REIVINDICACIONES

1. Un método para decodificar un flujo de bits (15), en donde el flujo de bits corresponde a un video codificado (1), en donde el video codificado (1) incluye una secuencia de fotogramas, en donde cada fotograma está dividido en bloques codificados, para cada bloque codificado caracterizado por que comprende las etapas de:
 - 5 determinar un conjunto de rutas en el bloque, en donde cada ruta está en un ángulo de transformada (1105) determinado a partir de un índice de transformada en el flujo de bits (15) y tiene una longitud de ruta m , estando especificada una ruta mínima de longitud de transformada $L_{min} > 2$ para el bloque;
 - obtener coeficientes de transformada (192) del flujo de bits (15), en donde los coeficientes de transformada incluyen un vector unidimensional de coeficientes DC transformados, en donde la longitud del vector varía según el ángulo de transformada, y una pluralidad de coeficientes AC, cada uno asociado con una ruta particular dentro del bloque,
 - 10 aplicar una transformada inversa secundaria solo al vector unidimensional de coeficientes DC transformados, para producir una serie de coeficientes DC, cada uno correspondiente a la ruta particular dentro del bloque; y
 - aplicar una transformada inversa primaria (25) a los coeficientes de transformada en cada ruta completa, que comprende el coeficiente DC y los coeficientes AC asociados, asociados con dicha ruta, para producir un video decodificado (2), en el que las etapas son realizadas en un decodificador (20),
 - 15 en donde la ruta es generada comenzando en un píxel inicial no transformado en el bloque e incluyendo repetidamente píxeles adyacentes en una dirección correspondiente a dicho ángulo de transformada, y realizando un giro en U cuando se alcanza el borde del bloque hasta que la ruta es reconstruida y considerada completa cuando la longitud m de la ruta, correspondiente al número de píxeles en la ruta, es igual o mayor que L_{min} , $m \geq L_{min}$, y cuando la ruta termina en un borde del bloque, lo que garantiza que el número de coeficientes que son transformados simultáneamente en el decodificador está por encima de un mínimo.
2. El método de la reivindicación 1, en donde cada ruta tiene un primer coeficiente (1150) para formar un conjunto de primeros coeficientes, y en donde la obtención comprende, además:
 - predecir el componente DC para cada ruta a partir del conjunto de primeros coeficientes (1150).
- 25 3. El método de la reivindicación 1, en donde la transformada inversa principal (25) está basada en una transformada discreta inversa de coseno.
4. El método de la reivindicación 1, que comprende, además:
 - dividir el bloque en un conjunto de particiones según un ángulo de partición.
- 30 5. El método de la reivindicación 4, en donde el ángulo de partición es perpendicular al ángulo de transformada (1105).
6. El método de la reivindicación 1, en donde los bloques incluyen macrobloques, subbloques, particiones de bloque o una matriz de píxeles.
7. El método de la reivindicación 1, que comprende, además:
 - cuantificar inversamente los coeficientes de transformada según un parámetro de cuantificación; y
 - 35 escalar los coeficientes de transformada según una ubicación de los coeficientes de transformada en el bloque.
8. El método de la reivindicación 1, que comprende, además:
 - cuantificar inversamente los coeficientes de transformada según un parámetro de cuantificación; y
 - escalar los coeficientes de transformada según las longitudes.

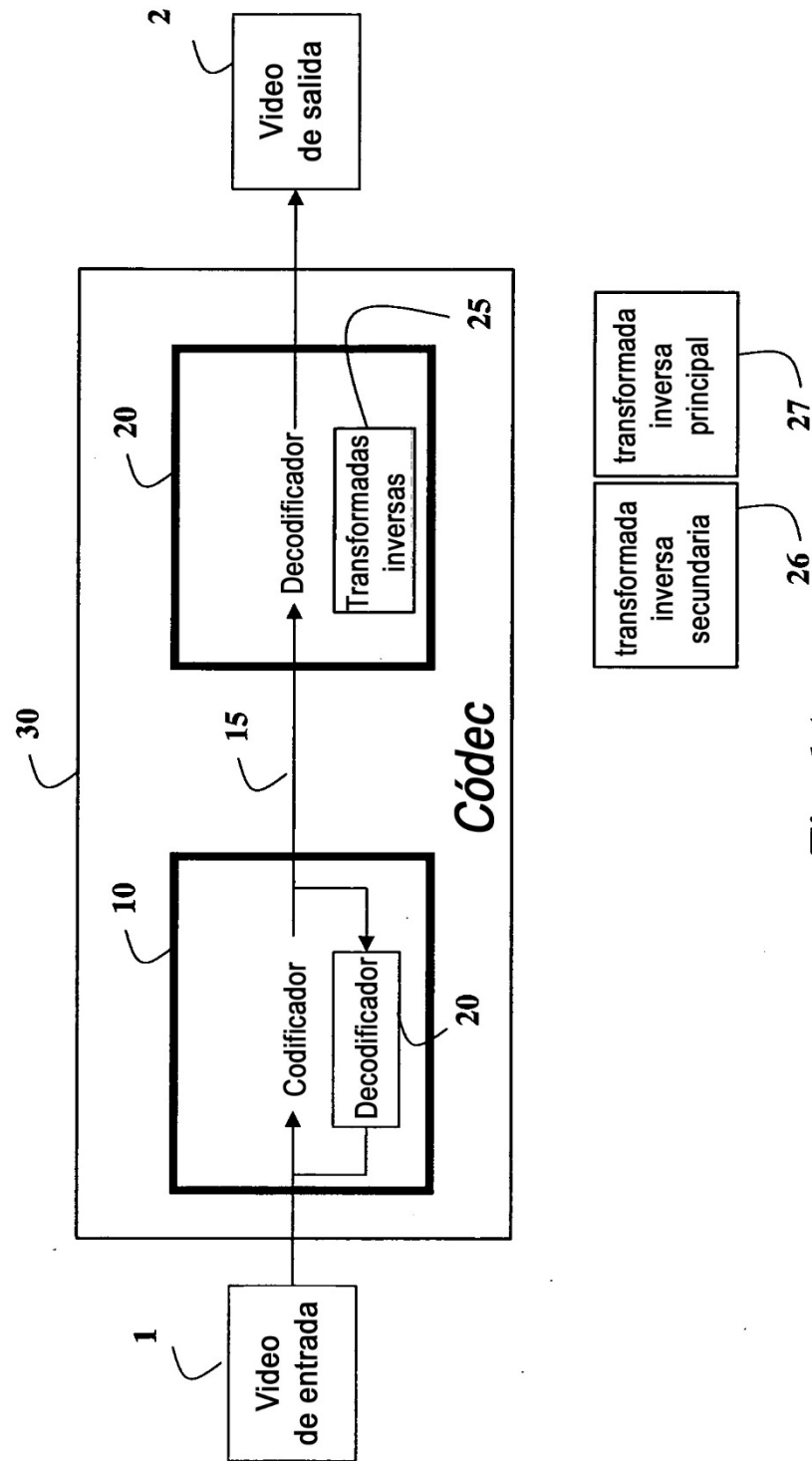


Fig. 1A

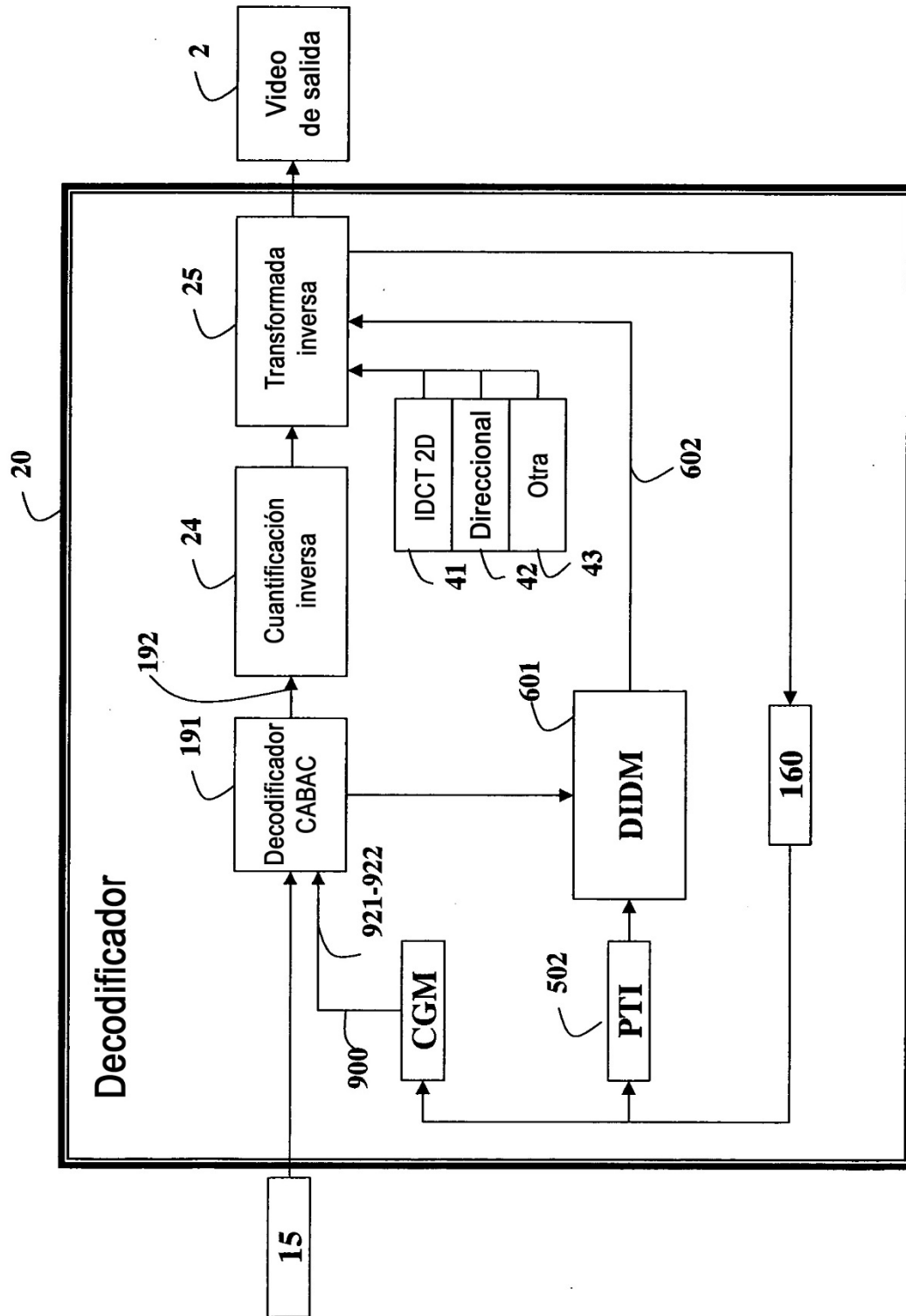
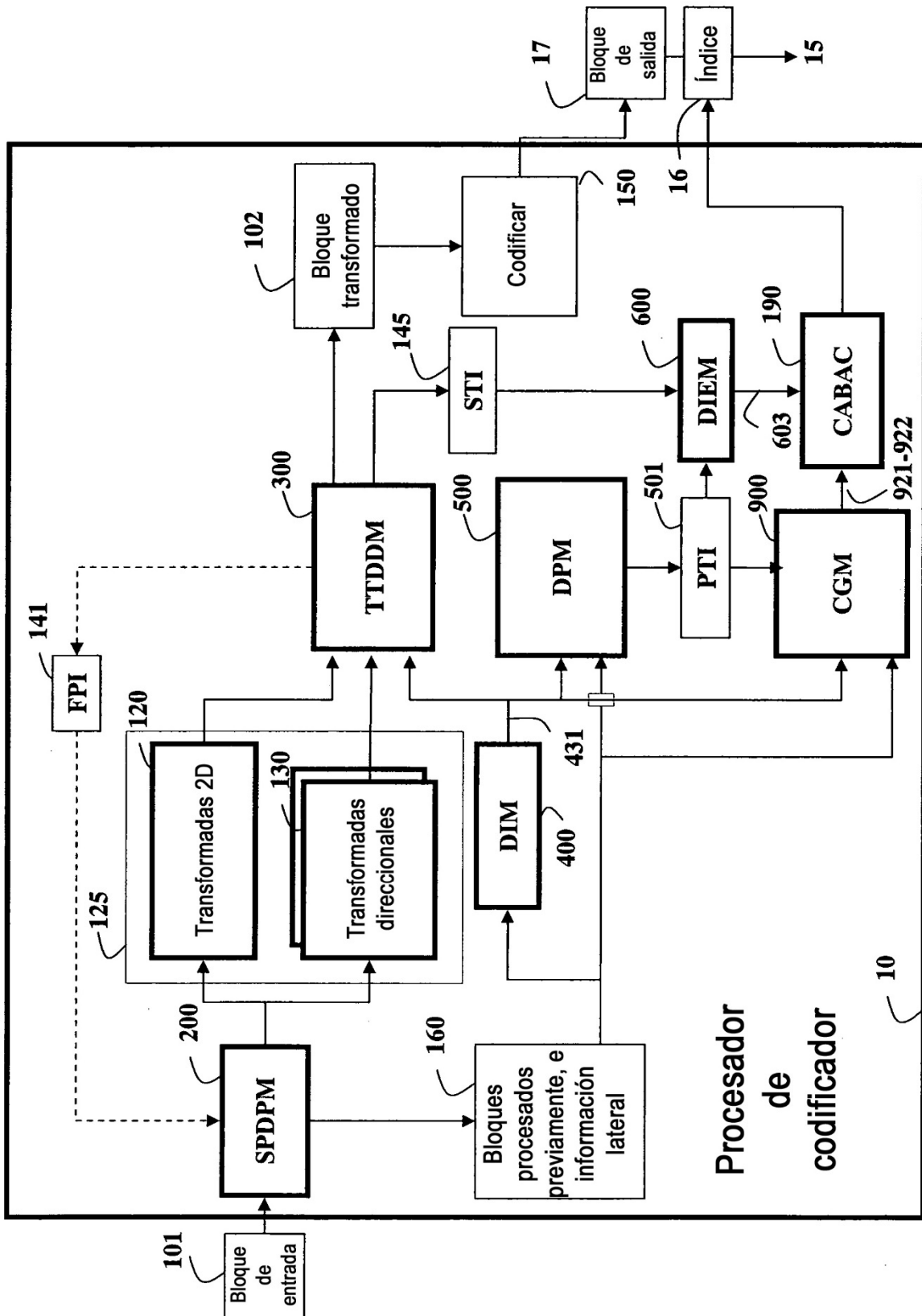


Fig. 1B



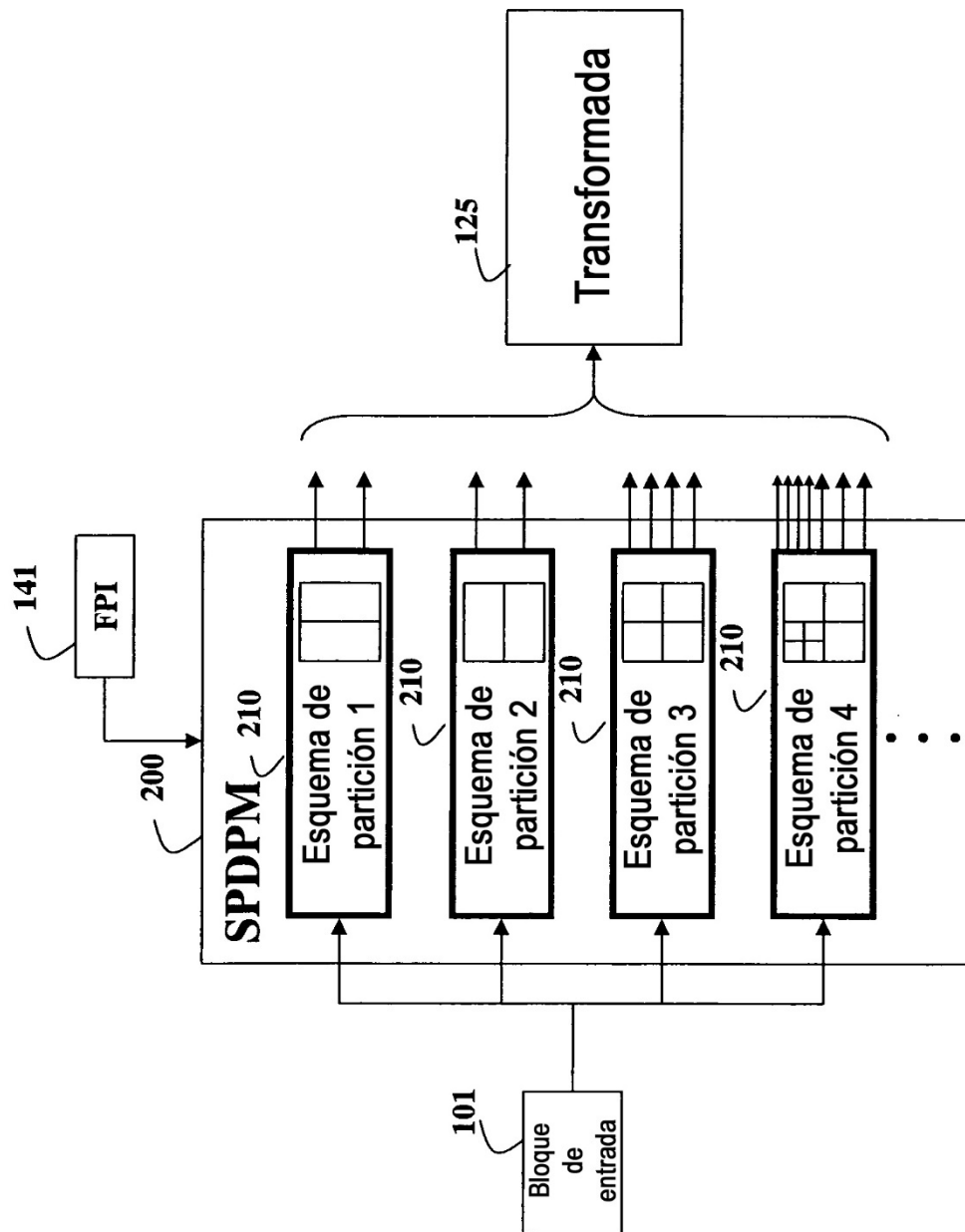


Fig. 2
200

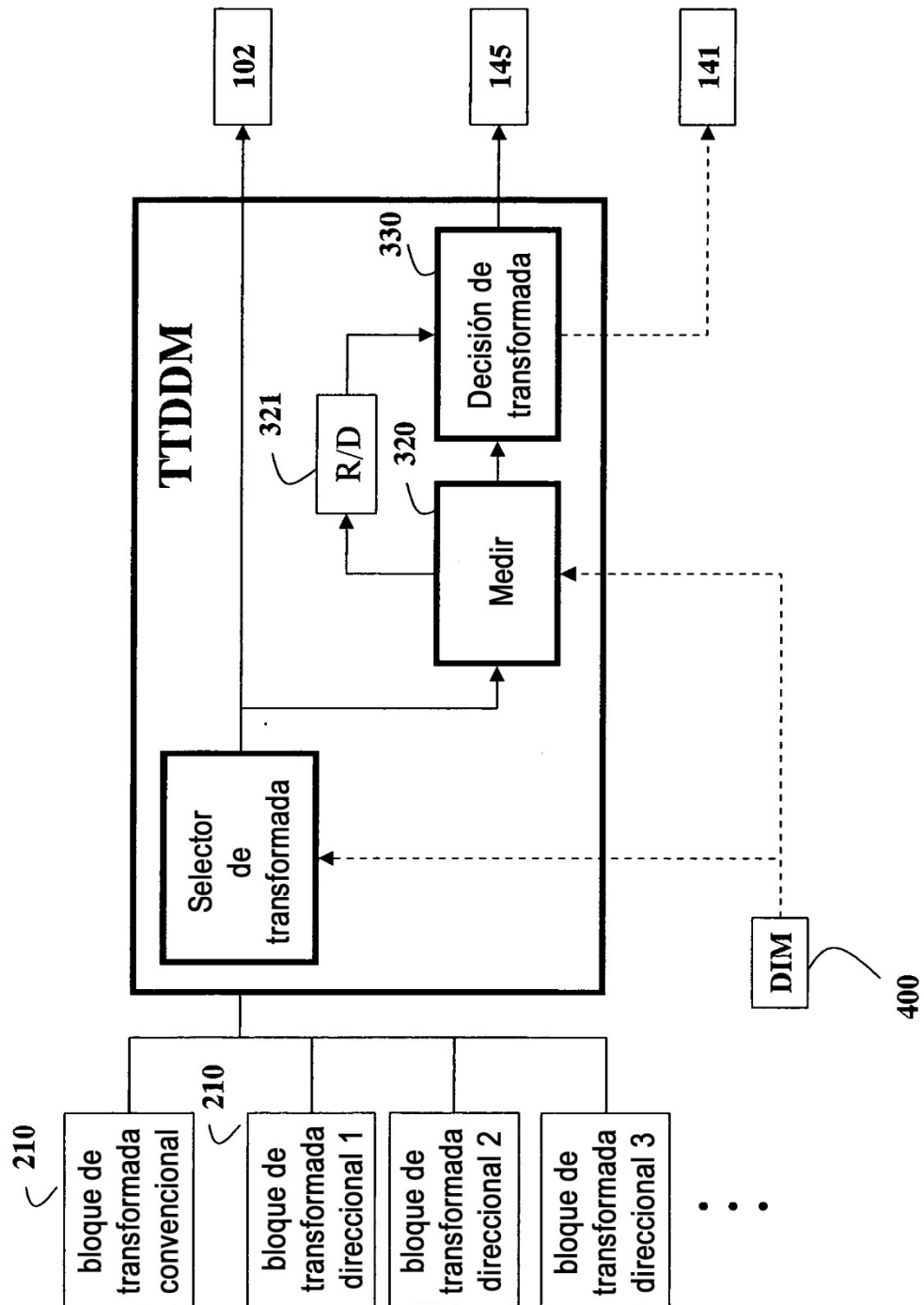


Fig. 3
300

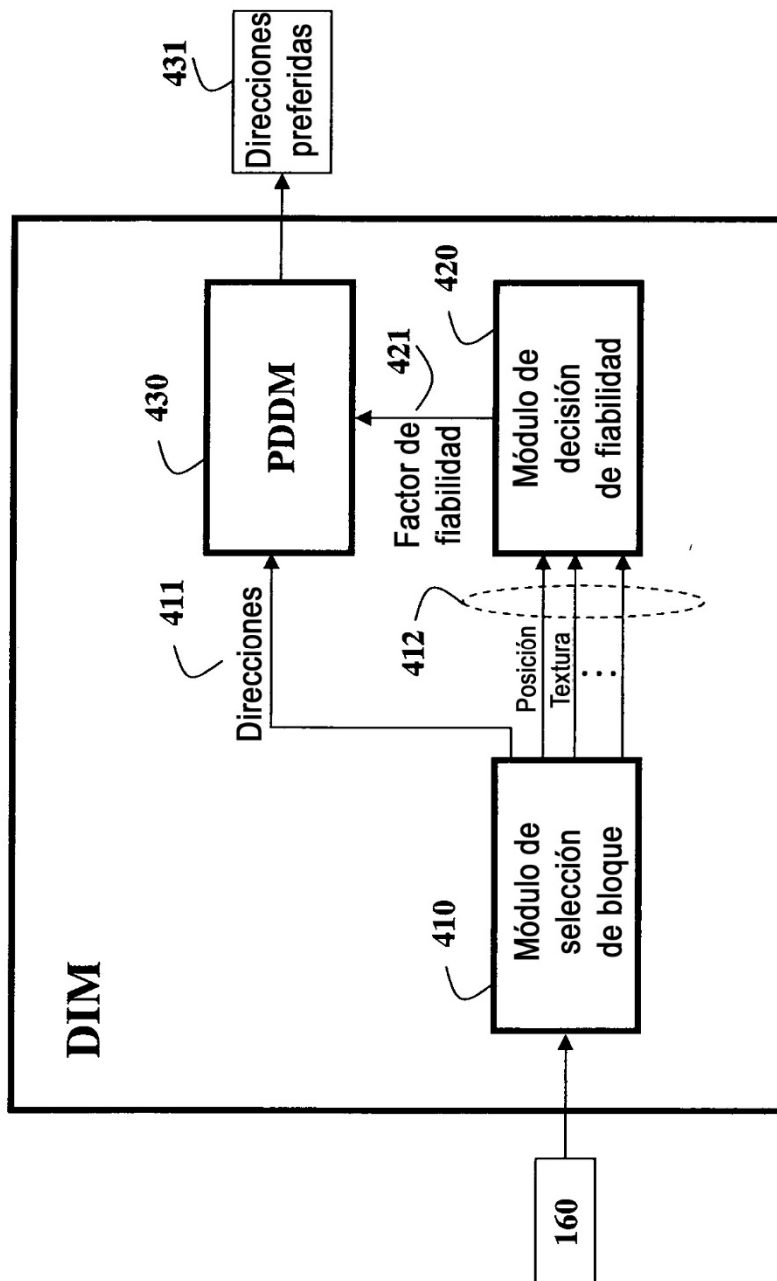


Fig. 4
400

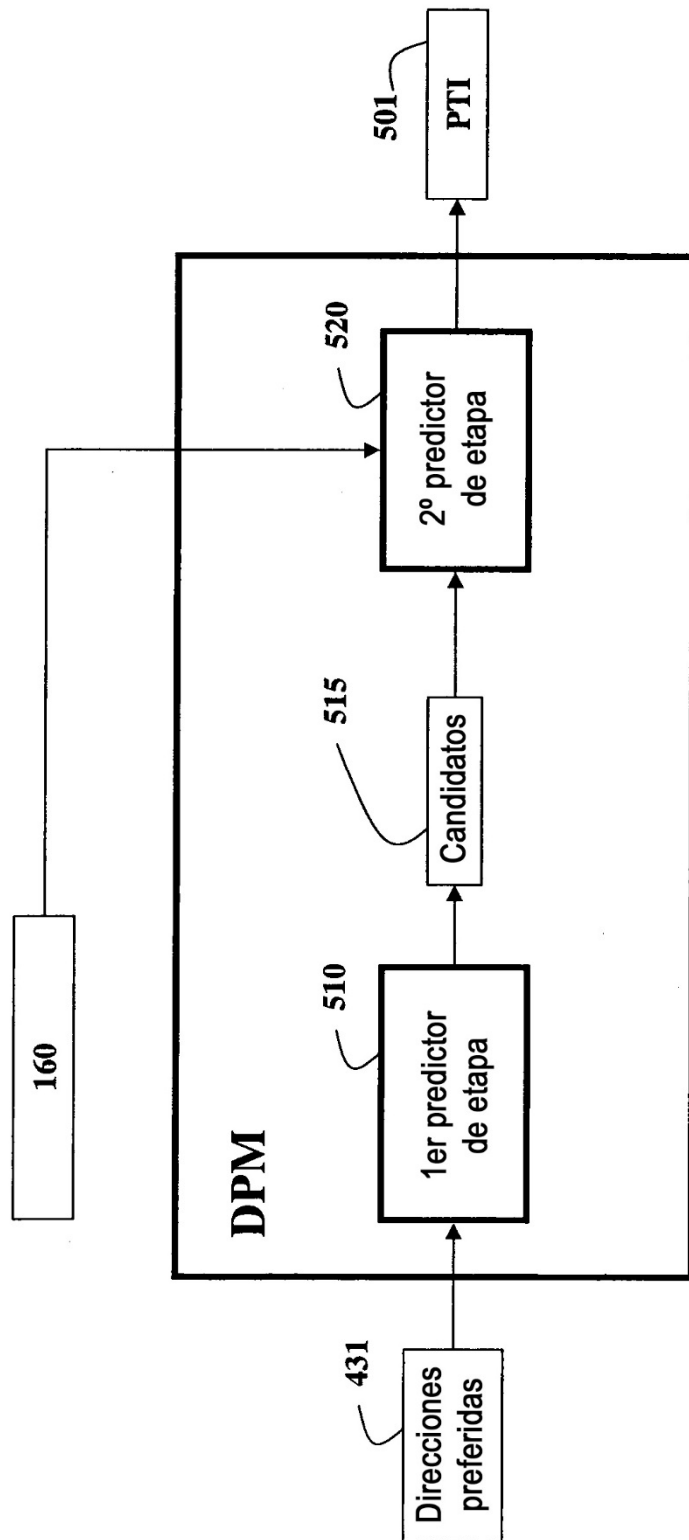


Fig. 5
500

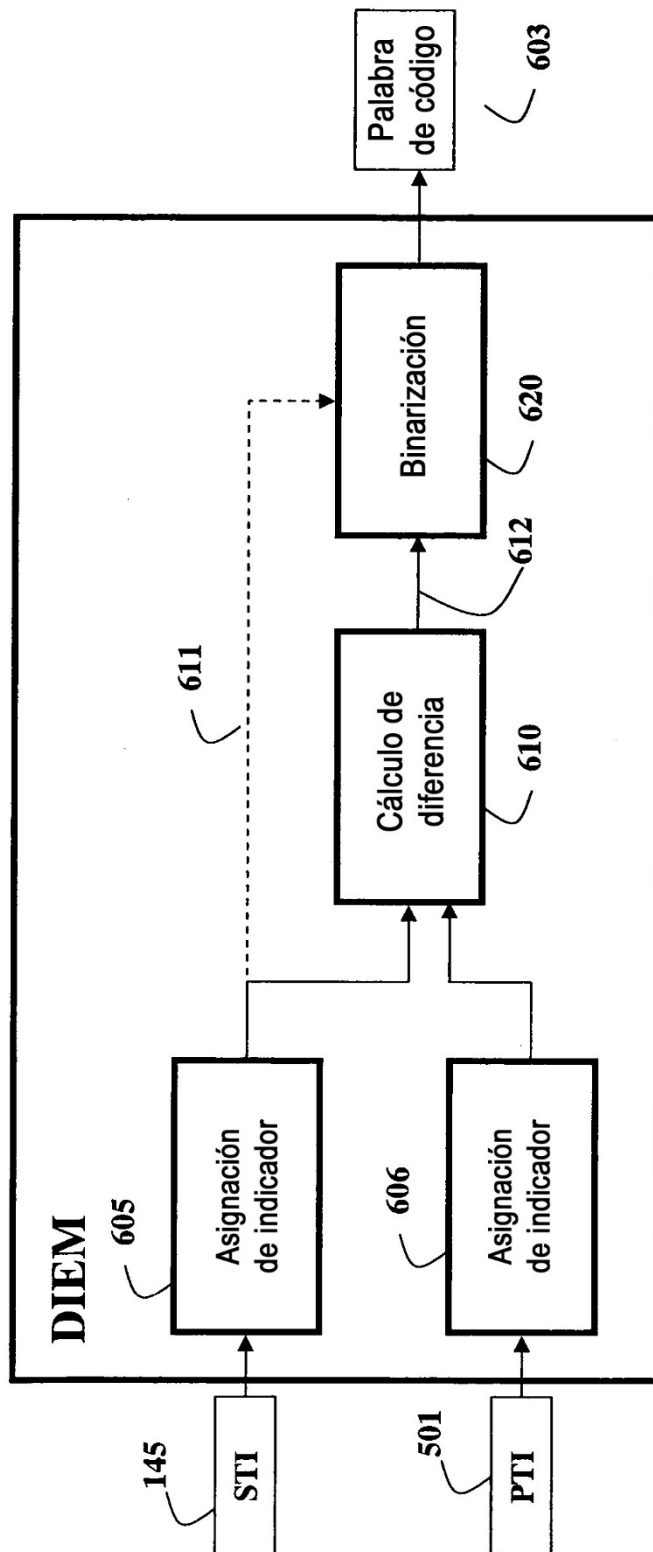


Fig. 6
600

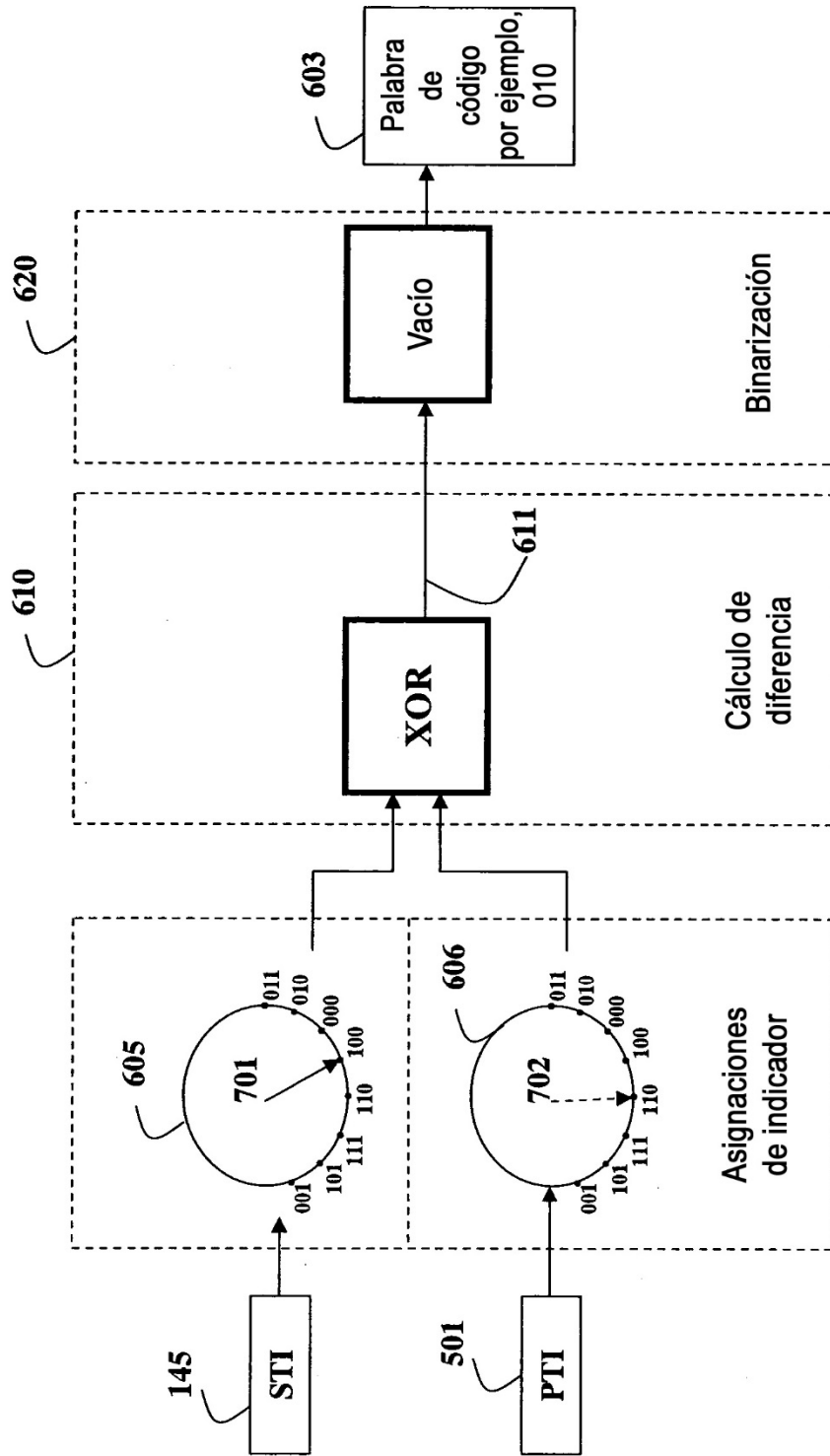


Fig. 7
700

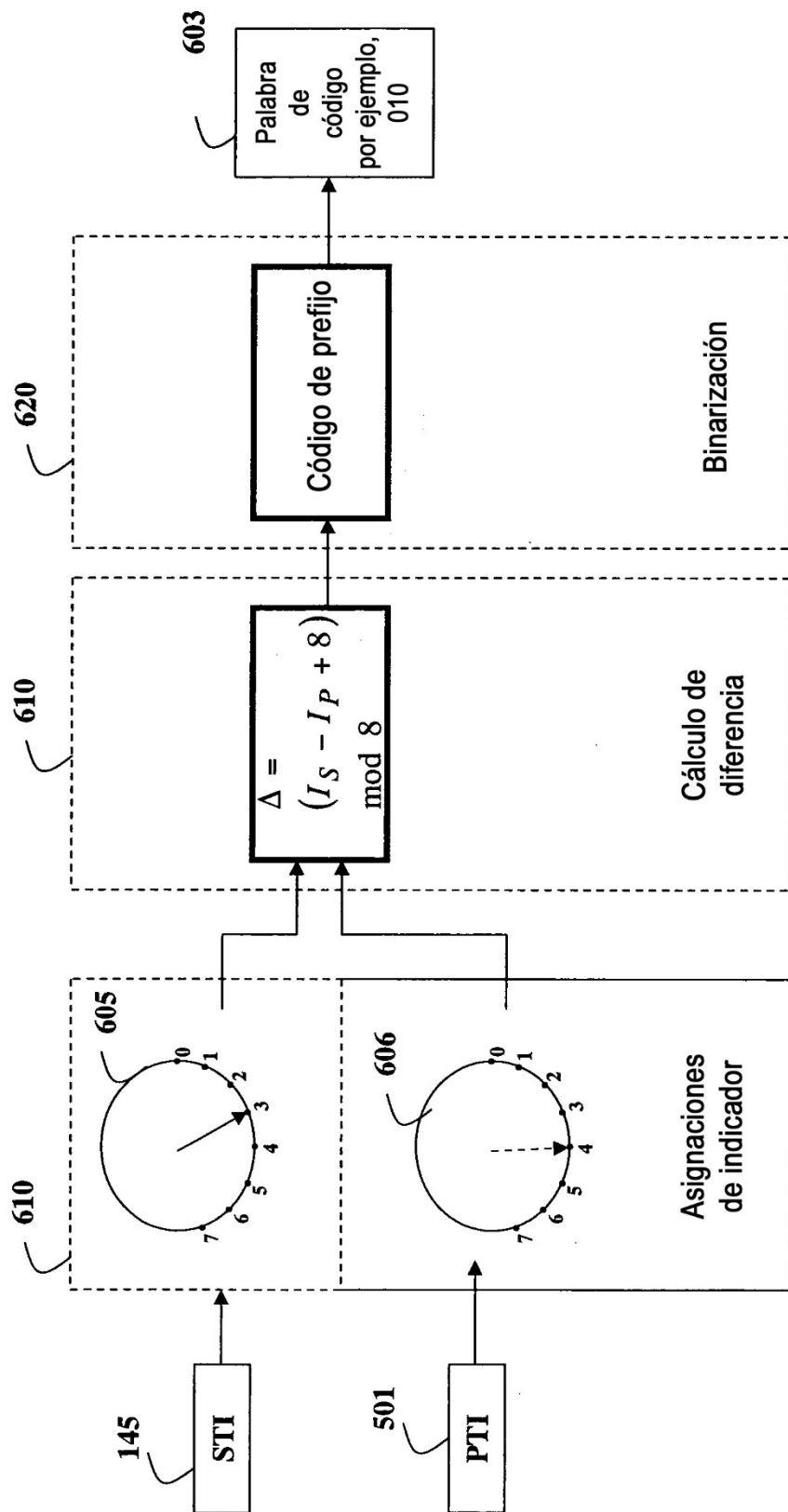


Fig. 8
800

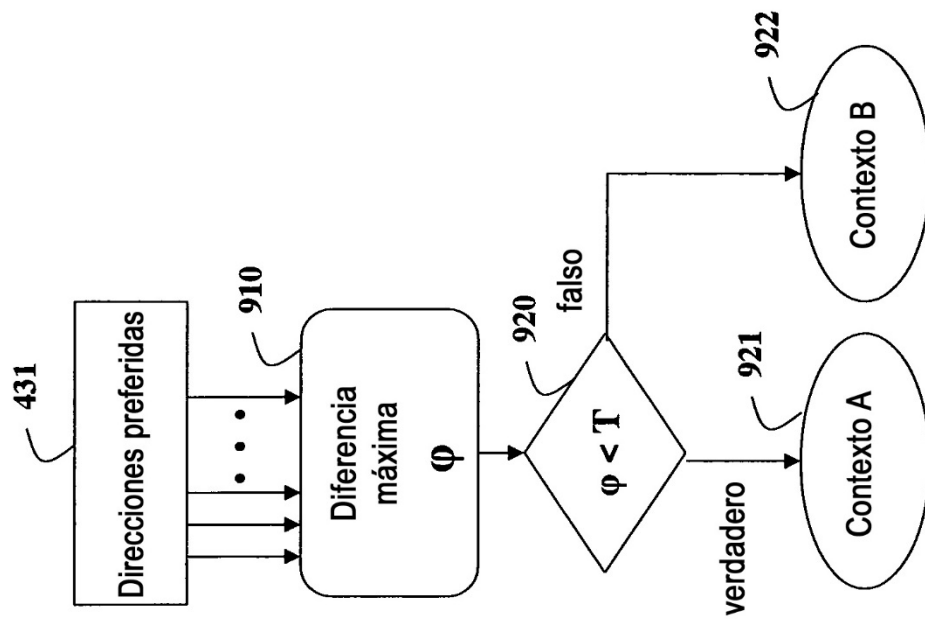


Fig. 9

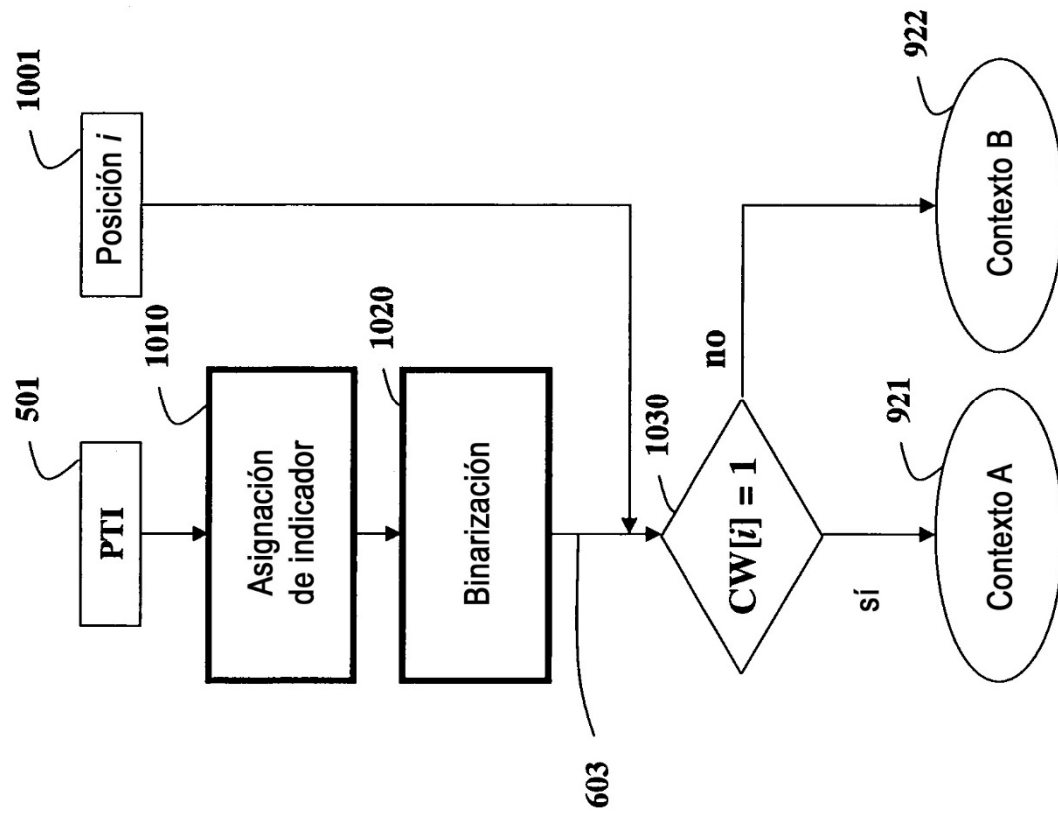


Fig. 10

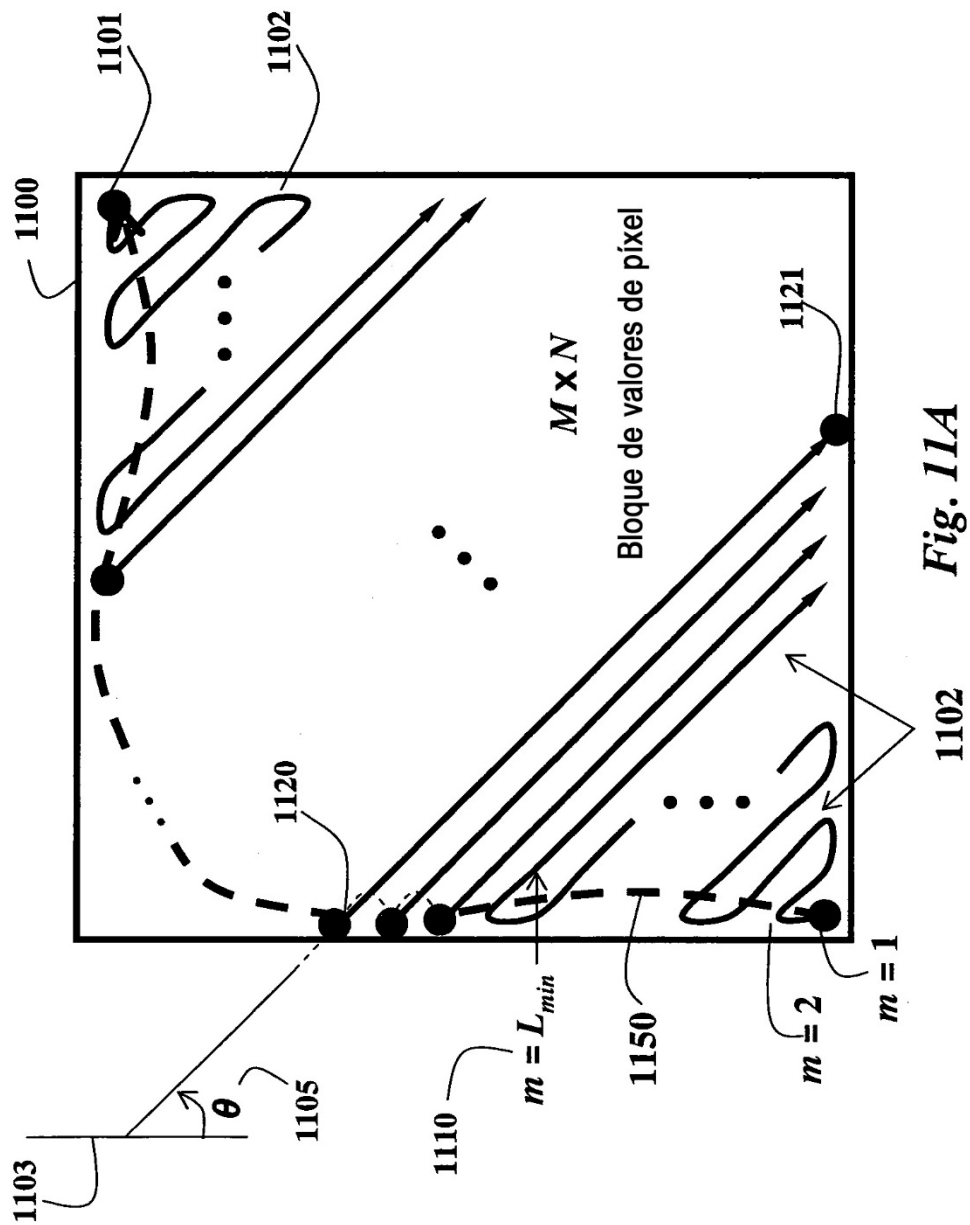


Fig. 11A

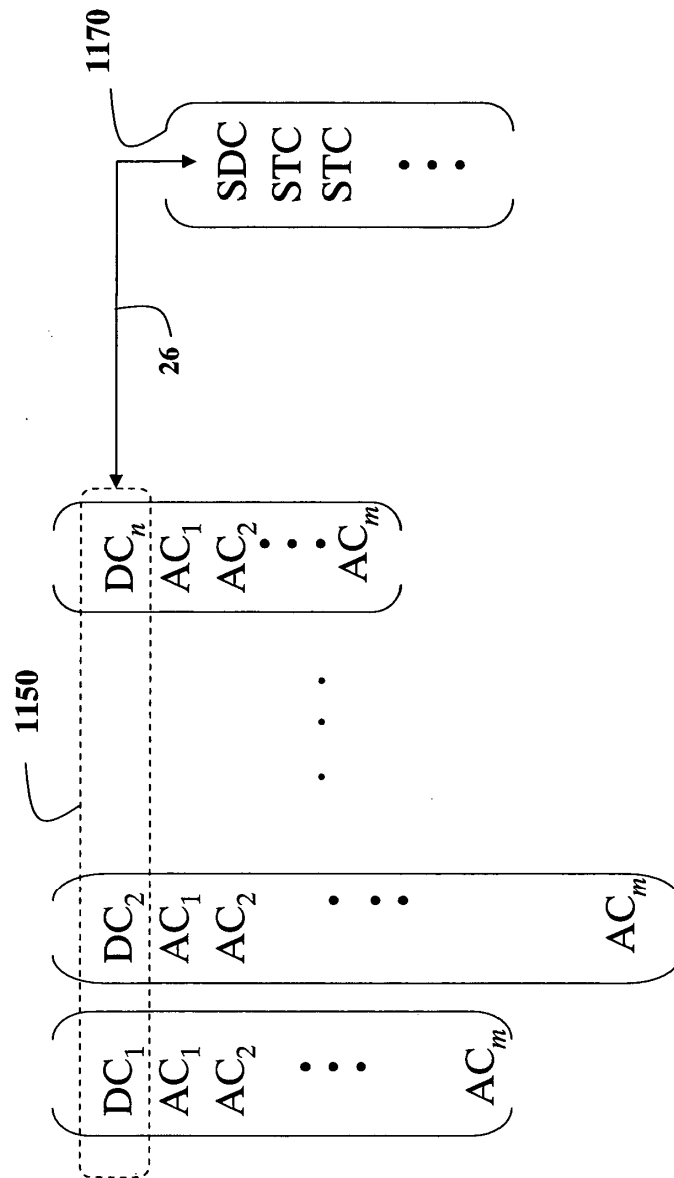


Fig. 11B

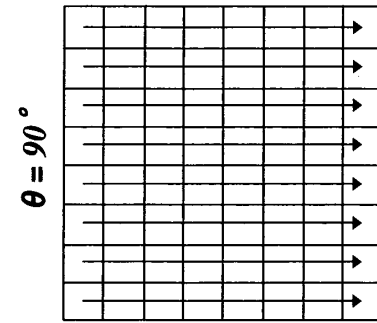


Fig 12C

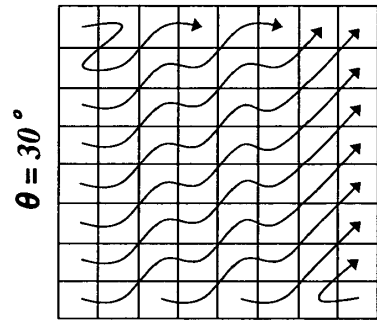


Fig 12B

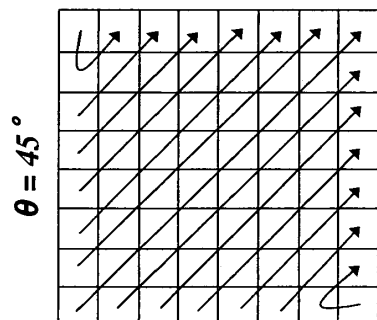


Fig 12A

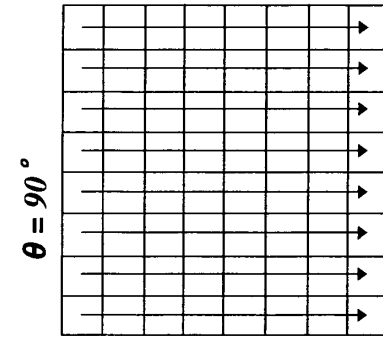


Fig 12F

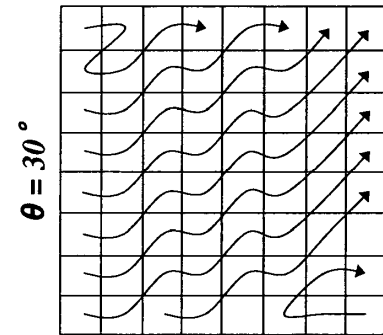


Fig 12E

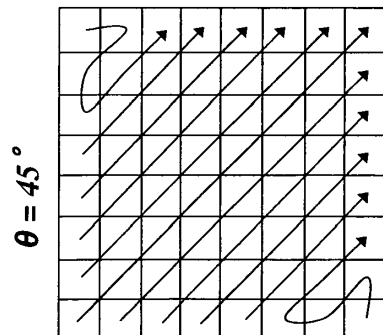


Fig 12D

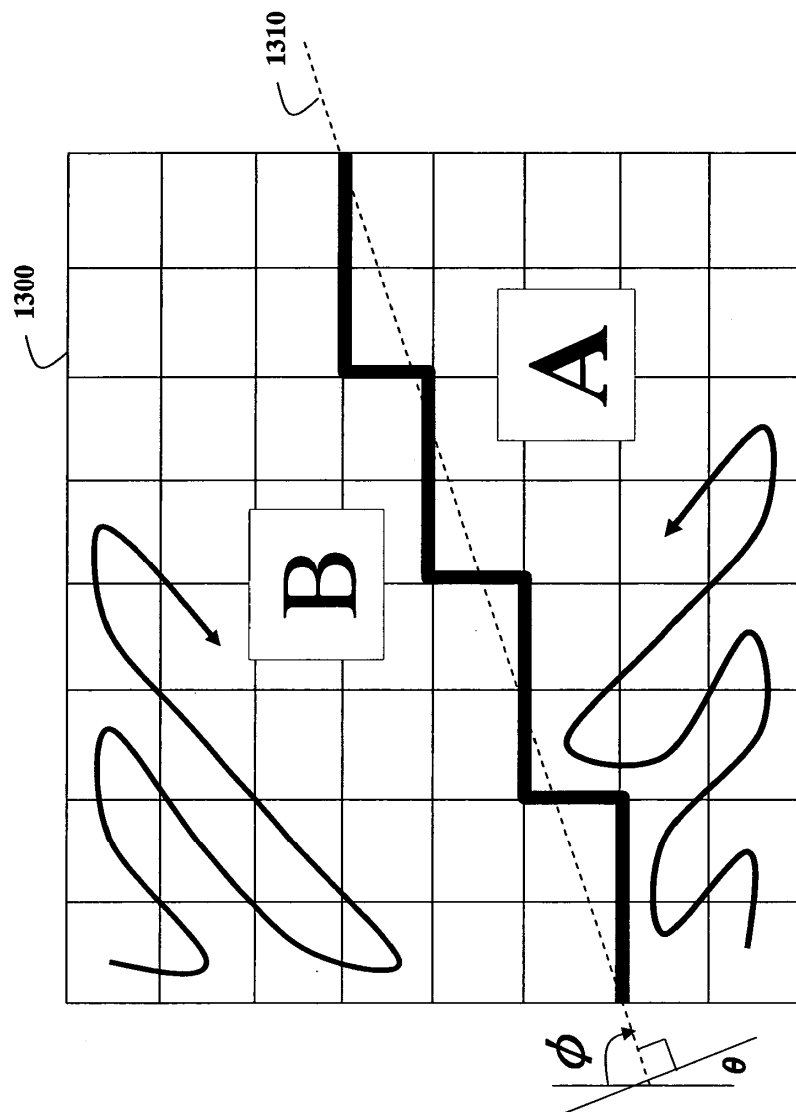


Fig. 13