

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 916 087**

51 Int. Cl.:

H04N 19/147 (2014.01)
H04N 19/115 (2014.01)
H04N 19/146 (2014.01)
H04N 19/176 (2014.01)
H04N 19/154 (2014.01)
H04N 19/103 (2014.01)
H04N 19/192 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **15.09.2016 PCT/GB2016/052883**
- 87 Fecha y número de publicación internacional: **20.04.2017 WO17064457**
- 96 Fecha de presentación y número de la solicitud europea: **15.09.2016 E 16770318 (0)**
- 97 Fecha y número de publicación de la concesión europea: **13.04.2022 EP 3363200**

54 Título: **Compresión de datos**

30 Prioridad:

16.10.2015 GB 201518290

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

28.06.2022

73 Titular/es:

**DIGITAL BARRIERS SERVICES LTD (100.0%)
121 Olympic Avenue, Milton Park, Milton
Abingdon, Oxon OX14 4SA, GB**

72 Inventor/es:

LAMBERT, ROBERT

74 Agente/Representante:

TOMAS GIL, Tesifonte Enrique

ES 2 916 087 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

Compresión de datos

5

[0001] La invención se refiere a un método y aparato para comprimir una matriz de entradas de datos en forma de señales electrónicas digitales.

10

[0002] El vídeo digital es una secuencia de imágenes visuales o fotogramas que transmiten la impresión de movimiento cuando se ven en orden temporal. Cada fotograma se codifica como un mapa de bits ortogonal compuesto de píxeles individuales, donde cada fila tiene un mismo número de píxeles que definen el ancho del fotograma y, de forma similar, cada columna tiene el mismo número de píxeles que definen la altura del fotograma. Cada píxel codifica el color donde se pueden representar variaciones de color más sutiles aumentando el número de bits usados para codificar cada píxel (a veces denominado profundidad de píxel). Por lo tanto, el número de bits requeridos para representar un único fotograma de vídeo es ancho multiplicado por alto multiplicado por profundidad de píxel.

15

20

[0003] Para proporcionar la impresión de movimiento, es necesario mostrar estos fotogramas en sucesión rápida a una velocidad de fotogramas constante, donde la velocidad de fotogramas se define como fotogramas por segundo (fps). Una velocidad de fotograma de vídeo 24 fps se considera generalmente el mínimo para transmitir la impresión de movimiento continuo.

25

[0004] La necesidad de compresión de vídeo se puede apreciar cuando se considera el volumen de datos requerido para transmitir o almacenar vídeo digital sin formato. Un fotograma de vídeo a todo color de alta definición (1920x1080) con una profundidad de píxel de 16 bits por píxel corresponde a 33 millones de bits o 4 Mbytes. Un vídeo de 1920x1080 con 25 fotogramas por segundo tiene 99 Mbytes de datos por segundo, lo que significa que un videoclip de alta definición de un minuto requiere aproximadamente 5,8 Gbytes de datos para transmitir o almacenar. Esto implicaría largos tiempos de descarga y podría exceder las asignaciones de datos para muchos clientes de teléfonos móviles.

30

35

[0005] Las técnicas de compresión usadas para el texto son sin pérdidas, donde la descompresión debe reproducir exactamente el mismo texto. La compresión de vídeo, por el contrario, explota la redundancia espacial y temporal dentro y entre los fotogramas de vídeo para reducir significativamente la cantidad de datos necesarios para transmitir o almacenar vídeo aproximando en lugar de reproducir los valores de píxeles originales. Por lo tanto, la compresión de vídeo es un proceso de compresión con pérdida en el que la calidad del vídeo descomprimido (que se aproxima bien al vídeo original) depende en la relación de compresión y la complejidad del vídeo.

40

[0006] La relación de compresión se expresa generalmente como $\left(\frac{\text{tamaño de fuente}}{\text{tamaño comprimido}}\right):1$, donde el tamaño se mide en bits. La mayoría de los algoritmos de compresión de vídeo pueden lograr fácilmente una relación de compresión de 100:1, lo que permite comprimir el videoclip de alta definición de 1 minuto hasta 59 Mbytes. Esta compresión significa que el videoclip se puede descargar en minutos en lugar de horas.

45

50

[0007] El vídeo se comprime para minimizar el almacenamiento, reducir los costes de transmisión o permitir la transmisión en vivo, y estos objetivos implican generalmente diferentes métodos. La compresión para minimizar el almacenamiento implica generalmente establecer un umbral de calidad aceptable para la compresión de cada fotograma, de manera que haya pocas o ninguna diferencia visible entre la fuente y el vídeo comprimido. El tamaño comprimido de cada fotograma de vídeo es irrelevante, lo que permite que el proceso de compresión de vídeo asigne más bits a aquellos segmentos del vídeo que tienen el detalle espacial más complejo o el movimiento rápido entre fotogramas. Dichas técnicas a veces se denominan "velocidad variable", donde el tamaño del vídeo comprimido dependerá del contenido del vídeo.

55

60

[0008] Cuando se trata de transmisión, es normal apuntar a un ancho de banda, definido como bits por segundo, en lugar de calidad. Esto garantiza que el vídeo comprimido tenga un tamaño predecible y, por lo tanto, se pueda controlar el uso de ancho de banda. La mayoría de los algoritmos de compresión emplean un control de velocidad que ajusta la calidad del objetivo en un fotograma por fotograma, de manera que se logre un ancho de banda objetivo. Para alcanzar el ancho de banda objetivo, es posible que el algoritmo de compresión tenga que "disminuir" fotogramas si la calidad es demasiado baja. Esto da como resultado una velocidad de fotograma variable, donde la velocidad de fotograma cae a medida que el movimiento capturado por el vídeo se vuelve más rápido o complejo.

65

[0009] La transmisión de baja latencia en tiempo real a una velocidad de fotograma constante solo se puede lograr mediante el uso de codificación de velocidad fija. A diferencia de los enfoques precedentes que comprimen cada fotograma de vídeo a una calidad objetivo que puede ser fijar o variable, la codificación de velocidad fija requiere que el proceso de compresión tenga como objetivo el número de bits usados para comprimir cada fotograma individual. Esto asegura que el tamaño comprimido de cada fotograma coincida con la capacidad real

de un canal de comunicaciones, donde el canal podrá transmitir datos hasta un ancho de banda máximo definido. La compensación es la calidad visual que variará según la complejidad de la secuencia de vídeo y la cantidad de movimiento de la escena.

5

[0010] En la WO97/16026 se describe una técnica para proporcionar una codificación de vídeo de velocidad fija, mediante la cual cada fotograma se puede aproximar como una colección de bloques de tamaño variable, donde los píxeles en cada bloque se aproximan a los valores de píxeles de fuentes equivalentes usando una técnica de codificación de imagen establecida. El algoritmo añade sucesivamente codificaciones comprimidas correspondientes a bloques individuales hasta que la codificación de fotograma resultante alcanza el tamaño de fotograma comprimido objetivo. En la WO97/16026, el algoritmo añade nuevas codificaciones comprimidas, que brindan la mayor mejora en la representación sin comprimir el fotograma de vídeo. En otras palabras, el algoritmo añade bloques, que ofrecen el error de reconstrucción más bajo, donde esta es una medida de la diferencia entre el fotograma de origen y el fotograma que se generaría a partir de una descodificación del fotograma comprimido.

10

15

[0011] Una limitación con el enfoque en la WO97/16026 es que el algoritmo se centra únicamente en mejorar el error de reconstrucción. Una dificultad es que a veces se logra el error de reconstrucción más bajo proporcionando codificaciones con un tamaño de datos relativamente grande. Por lo tanto, es posible que este algoritmo de compresión no siempre haga el uso más eficiente del ancho de banda disponible.

20

[0012] Otra limitación con el método en la WO97/16026 es que se pueden añadir los bloques más pequeños de tal manera que ocluyan parcialmente un parche previamente seleccionado. Aunque esto puede ofrecer un error de reconstrucción reducido, también puede ser problemático porque puede conducir a aproximaciones subóptimas.

25

[0013] La US 2005/100229 A1 y la US 2007/153897 A1 describen otros métodos de compresión. Estos dos documentos se centran en la optimización de bloques individuales.

30

[0014] Según un aspecto de la presente invención, se proporciona un método de compresión de una matriz de entradas de datos en forma de señales electrónicas digitales según la reivindicación 1.

[0015] De esta manera, se proporciona una técnica de compresión eficiente que equilibra los deseos contrapuestos de una representación sin comprimir de alta calidad de la matriz de datos original (es decir, valores de error de reconstrucción bajos) y un tamaño de datos bajo para la matriz de datos comprimidos. Esto se logra mediante la actualización iterativa de la representación comprimida mediante la adición de codificaciones comprimidas que ofrecen la mayor mejora relativa en el error de reconstrucción por aumento de unidad en el tamaño de los datos. Esto puede aumentar gradualmente el tamaño de los datos de la representación comprimida y disminuir gradualmente el error de reconstrucción hasta que se alcance un tamaño de datos máximo. Esta técnica puede permitir la creación de una representación comprimida con un error de reconstrucción óptima dentro de un presupuesto de datos específico.

35

40

[0016] Esta técnica es particularmente útil en la compresión de vídeo en vivo sobre canales de ancho de banda variable. La técnica puede permitir una compresión optimizada con un error de reconstrucción mínimo dentro de las limitaciones del ancho de banda de canal. Por lo tanto, el vídeo en vivo se puede transmitir con una latencia mínima al hacer el mejor uso del ancho de banda disponible.

45

[0017] También se superan las dificultades relacionadas con las oclusiones parciales. En el presente método, un parche anterior puede reemplazarse solo si está completamente ocluido por bloques más pequeños. Esto puede conducir a una mejor optimización de la codificación.

50

[0018] En la invención, la matriz de datos original se divide en bloques de uno o más tamaños, y se determina una pluralidad de posibles codificaciones comprimidas para cada bloque. De esta manera, se pueden proporcionar codificaciones comprimidas para divisiones de la matriz original. Se puede formar una representación comprimida completa de la matriz subyacente combinando codificaciones comprimidas para las diversas subdivisiones. En general, los bloques de mayor tamaño se pueden codificar con un tamaño de datos menor, pero con un error de reconstrucción mayor. Por lo tanto, es ventajoso proporcionar una pluralidad de niveles de división para la matriz de datos original con una granularidad creciente (es decir, los bloques con un tamaño reducido). De esta manera, se puede lograr una iteración efectiva en la que se reduce el error de reconstrucción a medida que aumenta el tamaño de los datos para la representación comprimida.

55

60

[0019] En la invención, se determina una pluralidad de posibles codificaciones comprimidas para cada bloque usando una pluralidad de técnicas de compresión. La técnica de compresión óptima para un bloque generalmente depende de las propiedades de los datos en el bloque. Al proporcionar una pluralidad de posibles técnicas de compresión, es posible seleccionar la técnica más eficaz para comprimir los datos subyacentes (es decir, la técnica que puede comprimir datos con el tamaño de datos y el error de reconstrucción óptimos). Para la codificación de vídeo, por ejemplo, la estimación de movimiento se puede usar como una técnica de compresión

65

5 para eliminar información temporalmente redundante entre secuencias de fotogramas de vídeo. La estimación de movimiento puede ser una técnica de compresión eficaz para fotogramas de vídeo en los que se produce algún movimiento, de manera que existen similitudes significativas entre fotogramas sucesivos. En otro ejemplo, la codificación de longitud de ejecución es eficaz para fotogramas de vídeo que tienen un gran número de entradas de datos "cero", intercaladas con valores distintos de cero. Al determinar las codificaciones comprimidas usando varias técnicas, el método tiene flexibilidad para seleccionar las codificaciones comprimidas más eficientes (es decir, la codificación comprimida que, cuando se descomprime y añade a la matriz de datos de referencia, proporciona la mayor mejora relativa en el error de reconstrucción por aumento de unidad en el tamaño de los datos).

10 [0020] Los ejemplos no limitantes de posibles técnicas de compresión para la codificación de vídeo incluyen estimación de movimiento, codificación de transformación, cuantificación de vector y codificación residual.

15 [0021] El paso de generar la lista de la pluralidad de codificaciones comprimidas puede comprender el paso de clasificar las posibles codificaciones para cada bloque en términos de tamaño de datos creciente y excluir de la lista cualquiera de las posibles codificaciones comprimidas para cualquier bloque dado que no produzca una disminución en el error de reconstrucción para ese bloque para un aumento en el tamaño de los datos.

20 [0022] El paso de seleccionar una de la pluralidad de codificaciones comprimidas puede comprender seleccionar una segunda codificación con respecto a un bloque para el cual se seleccionó previamente una primera codificación en una iteración anterior.

25 [0023] El paso de actualizar la codificación comprimida puede comprender usar la segunda codificación además de la primera codificación.

[0024] El paso de actualizar la codificación comprimida puede comprender usar la segunda codificación en lugar de la primera codificación.

30 [0025] La matriz de datos original se puede dividir primero en una pluralidad de macrobloques y luego en bloques dentro de cada macrobloque.

[0026] La lista de la pluralidad de codificaciones comprimidas puede incluir codificaciones para bloques

35 [0027] en cada macrobloque, así como para cada macrobloque completo.

[0028] El paso de generar una lista de la pluralidad de codificaciones comprimidas puede comprender determinar una pluralidad de codificaciones comprimidas de macrobloques para cada macrobloque, donde cada codificación de macrobloques comprende un conjunto de codificaciones comprimidas para bloques en el macrobloque respectivo y el paso de seleccionar una de la pluralidad de codificaciones impresas comprende seleccionar una de las codificaciones comprimidas de macrobloque.

40 [0029] Según otro aspecto de la invención, se proporciona un codificador de vídeo configurado para enviar datos de vídeo a través de un canal según la reivindicación 12.

45 [0030] Según otro aspecto, se proporciona un medio de almacenamiento legible por ordenador no transitorio que tiene un programa informático almacenado en él, según la reivindicación 13.

50 [0031] Las características del método pueden proporcionarse como características del aparato correspondiente y viceversa.

[0032] Ahora se describirán formas de realización de la presente invención, solo a modo de ejemplo, con referencia a los dibujos adjuntos, en los que:

55 La figura 1 es una vista esquemática de un sistema de transmisión en una forma de realización de la presente invención;

60 La figura 2 es un diagrama que muestra una posible subdivisión jerárquica de un macrobloque en una forma de realización de la presente invención;

La figura 3 es un diagrama que muestra posibles combinaciones de regiones para usar en la creación de una codificación de macrobloques; y

65 La figura 4 es un diagrama de flujo que muestra los pasos del método para usar en una forma de realización de la presente invención.

[0033] La figura 1 es una vista esquemática de un sistema de transmisión que incluye un codificador de vídeo 2 y un servidor 4 con los respectivos transmisores celulares 6, 8. El codificador 2 puede funcionar para enviar datos de vídeo, datos críticos en el tiempo y datos no críticos en el tiempo al servidor 4 a través del enlace celular. El servidor 4 puede funcionar para enviar datos críticos en el tiempo y datos no críticos en el tiempo al codificador 2 a través del enlace celular.

[0034] El servidor 4 está conectado a los dispositivos de usuario 22 a través de una red convencional 24, como internet. Se pueden usar varias conexiones alternativas entre el servidor 4 y los dispositivos de usuario 22, incluidos los canales de bajo ancho de banda. El servidor 4 está configurado para transmitir vídeo y/o otros datos recibidos desde el codificador 2 a los dispositivos de usuario 22 a través de internet. En general, se proporciona una conexión de mayor capacidad entre el servidor 4 y los dispositivos de usuario 22 que la que es posible en el enlace celular entre el servidor 4 y el codificador 2.

[0035] La cámara de vídeo 10 está configurada para capturar vídeo de una escena y proporcionarlo a un compresor 16 en el codificador 2. El compresor 16 puede funcionar para comprimir fotogramas de vídeo usando un factor de compresión ajustable para producir fotogramas de vídeo comprimidos con un tamaño de datos predeterminado. Un multiplexor 18 recibe vídeo comprimido del compresor de vídeo 16 y también recibe datos críticos en el tiempo y no críticos en el tiempo de la CPU 14. Estos datos se multiplexan en un único canal y se transmiten al servidor 4 a través del enlace celular. El compresor 16 se puede implementar como *hardware* dedicado o como un programa informático que se ejecuta en un procesador.

[0036] El codificador incluye un monitor de ancho de banda 20, que está configurado para controlar el comportamiento de los paquetes de datos que han sido transmitidos en el canal. El monitor de ancho de banda 20 también puede funcionar para dar instrucciones al compresor de vídeo 16 para que cambie el factor de compresión ajustable en respuesta a condiciones cambiantes en el canal. Las técnicas para calcular el factor de compresión ajustable se describen en la solicitud de patente en trámite n.º: GB1502434.2. Para cada fotograma de vídeo, se calcula un factor de compresión ajustable para que la representación comprimida del fotograma de vídeo tenga un tamaño de datos predecible.

[0037] En funcionamiento, el compresor 16 recibe un fotograma de vídeo sin comprimir de la cámara 10. El compresor 16 divide el fotograma recibido en una pluralidad de macrobloques, que se usan como unidad base para la codificación. El fotograma de vídeo está dividido en macrobloques que no se superponen, donde cada macrobloque codifica una matriz única de elementos de píxeles. Los macrobloques pueden tener cualquier tamaño y forma, pero normalmente son cuadrados.

[0038] Generalmente, el error de reconstrucción que se puede lograr disminuye a medida que se reduce el tamaño de los macrobloques. Sin embargo, el tamaño de los datos de la representación comprimida del fotograma de vídeo aumenta generalmente a medida que los macrobloques disminuyen en tamaño. Si los macrobloques fueran el único nivel de división en el fotograma, entonces habría una compensación entre el tamaño elegido del macrobloque, el tamaño de los datos de la representación comprimida y la calidad del vídeo resultante o el error de reconstrucción. Por esta razón, la partición de macrobloques se utiliza para apoyar una compensación entre alta compresión y calidad de reconstrucción que se puede igualar a la complejidad del detalle de píxeles representado por cada macrobloque.

[0039] Un ejemplo de partición de macrobloques se muestra en la figura 2. En este ejemplo, el nivel 4 representa el tamaño completo del macrobloque. Los otros niveles se crean usando divisiones horizontales sucesivas o verticales de los bloques de mayor nivel, y cada uno de los bloques de nivel 0 proporciona la aproximación de $\frac{N}{4} \times \frac{M}{4}$ píxeles del macrobloque original de $N \times M$ píxeles. Cabe señalar que en este ejemplo hay cuatro niveles de división, pero cualquier número podría usarse potencialmente cuando la representación fraccional de píxeles sea viable.

[0040] La partición real de cada macrobloque usado para la compresión de vídeo normalmente debería basarse en la complejidad del detalle de píxeles que se aproxima cuando se puede usar una codificación comprimida de cualquier combinación de bloques para representar los píxeles subyacentes. La figura 3 muestra cuatro ejemplos de particiones de macrobloques válidas. En cada uno de estos ejemplos, se calcula una codificación comprimida para cada uno de los bloques particionados, y estos se añaden para crear una codificación de macrobloques (MBC), que es una representación comprimida para el macrobloque. Un buen algoritmo de compresión usa bloques más grandes para áreas con detalles de píxeles más simples (superficies de baja frecuencia) y bloques pequeños para áreas que representan detalles de píxeles muy complejos (bordes o texturas de alta frecuencia).

[0041] Una codificación de macrobloques (MBC) proporciona una representación comprimida de los píxeles de fuente original, con contribuciones de varias particiones posibles en el macrobloque. La calidad de la aproximación de la MBC a los píxeles de fuente original se cuantifica mediante el error de reconstrucción que se basa en la suma de la diferencia entre los píxeles de fuente y la aproximación de píxeles equivalente de la MBC sin comprimir.

[0042] Una métrica de error de reconstrucción común usada para la compresión de vídeo es la suma de diferencias absolutas (SAD), que se calcula como;

5

$$SAD = \left(\sum_{i=1}^N \sum_{j=1}^M |S(i,j) - A(i,j)| \right) \Leftrightarrow \sum_{i=1}^N \sum_{j=1}^M \sqrt{(S(i,j) - A(i,j))^2}$$

10 Donde $S(i,j)$ es el valor del píxel de fuente en el *offset* i,j dentro el macrobloque y $A(i,j)$ es el valor del píxel reconstruido en el *offset* i,j basado en la aplicación de la MBC del tamaño de píxel $N \times M$.

[0043] La compresión de vídeo que emplea la partición de macrobloques tiene el desafío de determinar una partición óptima como una compensación entre la relación de compresión y la calidad de reconstrucción en función del tamaño de subbloque. Este desafío se calcula cuando se dispone de múltiples técnicas para aproximar los píxeles de fuente representados por cada subbloque.

[0044] La escala del desafío se puede apreciar cuando consideramos el número de posibles permutaciones de codificación de macrobloques. Para la partición de macrobloques detallada en la figura 2 existen T técnicas de codificación de bloques diferentes, que se pueden usar para aproximar los píxeles representados por cada uno de los subbloques de macrobloques. Tres ejemplos de técnicas de codificación incluyen compensación de movimiento, codificación de transformación y cuantificación de vector espacial, aunque muchas otras se le ocurrirían naturalmente a un experto en la técnica de la compresión de vídeo.

- 25
- En el nivel 0, cada bloque tiene T codificaciones posibles, donde cada una ofrecerá un error de reconstrucción diferente para un costo de bit dado.
 - En el nivel 1 tenemos la opción de T codificaciones posibles para cada bloque vertical u horizontal más la opción de subdividir cada bloque en dos bloques de nivel 0, donde cada uno tiene T codificaciones posibles. Esto nos da $T + T^2$ permutaciones de codificación para cada bloque de nivel 1.
- 30
- En el nivel 2 tenemos la opción de T codificaciones posibles para cada bloque de nivel 2, o la opción para subdividir cada bloque de nivel 2 en dos bloques horizontales o dos verticales de nivel 1, donde estos bloques de nivel 1 pueden volver a subdividirse en bloques de nivel 0. Esto nos da $T + 2T^2 + 4T^3 + T^4$ permutaciones de codificación para cada bloque de nivel 2.

[0045] Siguiendo esta lógica, el número total de permutaciones de codificación únicas para el macrobloque completo ilustrado en la figura 2 con T técnicas de codificación de bloques diferentes es;

$$T + 2T^2 + 4T(T + 2T^2 + 4T^3 + T^4) + (T + 2T^2 + 4T^3 + T^4)^4$$

40 [0046] Para $T = 4$, hay aproximadamente 9×10^{10} permutaciones de codificación únicas. En el presente método, el objetivo es determinar al menos algunas de estas 9×10^{10} permutaciones de codificación y sumarlas de forma reiterativa hasta lograr un tamaño de datos específico para el fotograma de vídeo, según el factor de compresión ajustable.

45 [0047] La figura 4 es un diagrama de flujo que ilustra la técnica reiterativa que se puede implementar para codificar un fotograma de vídeo. En el paso 40, el compresor 16 recibe un fotograma de vídeo de la cámara 10. En el paso 42, el fotograma de vídeo se particiona en una pluralidad de macrobloques; normalmente se crean alrededor de 1.000 macrobloques en el paso 42 para cada fotograma. En el paso 44, cada macrobloque se particiona, además, en regiones, como se ha descrito anteriormente con referencia a la figura 2. En el ejemplo mostrado en la figura 2, se crean cuarenta y una posibles regiones cuando se particiona un macrobloque; este número se determina sumando todas las subdivisiones del nivel 0 a 3, más el macrobloque completo (nivel 4). En el paso 46 se calculan cuatro codificaciones comprimidas para cada una de estas cuarenta y una regiones usando cuatro técnicas o algoritmos de compresión. La salida del paso 46 es 164 codificaciones comprimidas.

55 [0048] Como se explicó, estas 164 codificaciones comprimidas se pueden combinar en una gran cantidad de formas. En el paso 48, el compresor 16 calcula el tamaño de los datos para varias permutaciones diferentes, junto con el error de reconstrucción. A continuación, estas permutaciones se pueden filtrar y clasificar para retener solo aquellas que se puede clasificar en términos de aumento del tamaño de los datos y la disminución del error de reconstrucción.

60

[0049] En la práctica, puede que no sea factible calcular y ordenar todas las 9×10^{10} permutaciones de codificación y también proporcionar transmisión en tiempo real de los datos comprimidos. Por lo tanto, se pueden implementar varias técnicas para reducir el número de cálculos en el paso 48, como se explicará más adelante.

65 [0050] En el paso 50, el compresor 16 determina una codificación comprimida inicial para el fotograma de vídeo. La codificación comprimida inicial comprende una codificación comprimida para cada macrobloque, de modo que se determina una representación comprimida para todo el fotograma. Para proporcionar una codificación

5 comprimida inicial, la codificación seleccionada para cada macrobloque es la que tiene el tamaño de datos más bajo, independientemente de su error de reconstrucción. Esto proporciona una representación comprimida inicial que tiene el tamaño de datos más pequeño posible, pero puede ofrecer una mala aproximación al fotograma de video original cuando no está comprimido.

10 [0051] En el paso 52, el compresor 16 selecciona una de las codificaciones comprimidas calculadas en el paso 48, en todos los macrobloques en el fotograma, para añadirla a la codificación comprimida inicial. La codificación comprimida seleccionada es la que ofrece la mayor mejora relativa en el error de reconstrucción por aumento de unidad en el tamaño de los datos. Se ha determinado que esto proporciona un mecanismo eficiente para mejorar el error de reconstrucción, mientras que equilibra las demandas para una codificación comprimida con un tamaño de datos bajo.

15 [0052] La mejora relativa por bit (*RIPB*) se calcula en el paso 52, de la siguiente manera:

$$RIPB_n = \frac{r_{n-1} - r_n}{b_n - b_{n-1}}$$

20 [0053] Donde: r_n es el error de reconstrucción para el fotograma de vídeo, actualizado con una nueva codificación comprimida, añadido a la codificación comprimida inicial; r_{n-1} es el error de reconstrucción para la iteración anterior de la codificación comprimida (en la primera iteración corresponderá al error de reconstrucción para la codificación comprimida inicial); b_n es el costo de bit o el tamaño de los datos cuando la nueva codificación comprimida se añade a la codificación comprimida inicial; b_{n-1} es el costo de bit para la iteración anterior de la codificación comprimida.

25 [0054] Por lo tanto, en el paso 52, la *RIPB* se calcula para una pluralidad de posibles codificaciones que podrían usarse potencialmente para actualizar la representación comprimida del fotograma de vídeo. Una codificación comprimida se selecciona en el paso 52, que es la codificación comprimida que ofrece la mayor *RIPB*.

30 [0055] En el paso 54, la codificación comprimida inicial se actualiza añadiendo la codificación comprimida seleccionada en el paso 52. Esta codificación comprimida actualizada proporciona un error de reconstrucción mejorado para un mayor tamaño de datos.

35 [0056] En el paso 56, el compresor 16 analiza si el tamaño de los datos de la codificación comprimida actualizada es menor que el presupuesto de datos asociado al factor de compresión ajustable. Si la respuesta es "sí", entonces aun puede ser posible reducir el error de reconstrucción en el presupuesto de datos. Por lo tanto, los pasos 52 y 54 se repiten iterativamente. De esta manera, el error de reconstrucción se reduce gradualmente y el tamaño de los datos de la codificación comprimida se incrementa gradualmente añadiendo codificaciones comprimidas una por una, que ofrecen la mayor mejora relativa en el error de reconstrucción por aumento de unidad en el tamaño de los datos. Esta iteración continúa hasta que el tamaño de los datos de la codificación comprimida actualizada sea igual al presupuesto de datos. En otra disposición, la codificación comprimida no se actualiza en el paso 54 si esa actualización aumenta el tamaño de la codificación de comprimido más allá del presupuesto. En este punto, el proceso finaliza y la salida es una representación comprimida del fotograma de vídeo que tiene un tamaño de datos menor o igual que el presupuesto de datos, según lo determinado por el factor de compresión ajustable. La representación comprimida del fotograma de vídeo puede luego combinarse con otros datos mediante el multiplexor 18 y transmitirse al servidor 4 a través del canal de ancho de banda variable.

50 [0057] Cabe señalar que las codificaciones sucesivas seleccionadas durante la iteración en el paso 52 pueden diferir entre sí solo en la técnica de compresión usada para representar un subbloque. Por lo tanto, cada técnica de aproximación de bloques puede ofrecer una forma de reducir aun más el error de reconstrucción a expensas de un mayor tamaño de datos. Esto permite que el proceso comience con la técnica que da un mayor error de reconstrucción, pero con un costo de bit muy bajo, y la reemplace con una codificación de bloques alternativa más adelante en el proceso de compresión.

55 [0058] Como se explicó, puede que no sea factible en el paso 48 calcular y clasificar todas las permutaciones de codificación posibles para un fotograma de vídeo y también proporcionar transmisión en tiempo real de los datos comprimidos. Esto se puede lograr en parte reconociendo que la lista de posibles permutaciones de codificación se puede filtrar para que se pueda ordenar en términos de aumentar el tamaño de los datos y disminuir el error de reconstrucción. No hay ninguna ventaja en el cálculo de las opciones de codificación comprimida si no pueden ofrecer un error de reconstrucción mejorado en comparación con otra opción de codificación que tiene un tamaño de datos más pequeño.

65 [0059] En un ejemplo, la subdivisión del nivel 2 en la figura 2 comprende cuatro regiones dentro de un macrobloque. Cada una de estas cuatro regiones se puede codificar usando cuatro algoritmos de compresión. Por lo tanto, hay 16 codificaciones posibles para los datos subyacentes y 4 para cada región. Cada codificación tiene un tamaño de datos y un error reconstrucción asociados.

5 [0060] Por lo tanto, las cuatro codificaciones para cada región se pueden clasificar en orden de tamaño de datos
creciente. Entonces las codificaciones se pueden omitir si alguna no produce una disminución en el error de
reconstrucción para un aumento en el tamaño de datos. Estas codificaciones se pueden omitir porque serían
selecciones redundantes en el paso 52. Estas codificaciones nunca serían selecciones viables: no podrían
ofrecer la RIPB más grande porque existe una codificación alternativa (producida con un algoritmo de
compresión diferente) que produciría una mayor RIPB. Al identificar y omitir dichas codificaciones, es posible
10 realizar una reducción drástica en el número total de permutaciones, ya que cualquier permutación que incluya
una codificación omitida se excluye automáticamente.

[0061] Además, para lograr la compresión, el tamaño de los datos de la representación comprimida debe ser
menor que la matriz original. Por lo tanto, se puede establecer un límite superior para la cardinalidad del conjunto
de permutaciones de codificación, lo que significa que se pueden omitir las permutaciones de codificación de una
15 consideración adicional.

[0062] Usando estas técnicas, es posible reducir el número de cálculos en el paso 48 de manera muy
significativa. Como se explicó, si pudieran usarse cuatro posibles algoritmos de compresión junto con 1.000
macrobloques subdivididos de la manera de la figura 2, entonces habría alrededor de 9×10^{10} permutaciones de
20 codificación para calcular en el paso 48. Este número se puede reducir a alrededor de 1.000 usando las técnicas
anteriormente descritas. Por lo tanto, es posible derivar las mejores codificaciones comprimidas posibles sin
tener que derivar todas las codificaciones comprimidas posibles.

[0063] Estas técnicas también admiten la compresión de un fotograma de vídeo a un número predeterminado de
25 bits. Esto permite la compresión de una secuencia de fotogramas de vídeo a una tasa de bits constante que es
independiente del tamaño del fotograma y del contenido del fotograma de fuente. Esto puede permitir la
transmisión de vídeo de baja latencia a través de canales de comunicación de ancho de banda bajo o ancho de
banda variable. Por lo tanto, esta técnica es para transmitir fotogramas de vídeo con el error de reconstrucción
30 óptimo posible dentro de límites de ancho de banda específicos.

[0064] Para un fotograma representado por un único macrobloque, la compresión a un número objetivo de bits se
logra primero determinando este conjunto óptimo de MBCs ordenadas y luego extrayendo cada MBC a su vez
hasta que se extrae una entrada que requiere más bits para transmitir que el objetivo para la compresión del
35 fotograma. La MBC usada para aproximar los píxeles del macrobloque en este caso sería la última MBC extraída
del conjunto ordenado que tenía un costo de bit menor o igual que el objetivo de costo de bit del fotograma. Para
un fotograma representado por múltiples macrobloques, se construye un conjunto ordenado de conjuntos de
manera que las MBCs para todos los macrobloques se ordenan colectivamente por mejora relativa por bit y costo
de bit creciente. La compresión del fotograma de vídeo a un número objetivo de bits se logra extrayendo MBCs
40 del conjunto ordenado de conjuntos donde registramos el costo de bit de la última entrada extraída de cada
conjunto de MBC individual. La suma del costo de bit corresponde al número de bits necesarios para transmitir la
última MBC extraída de cada uno de los conjuntos de MBC y, por lo tanto, el costo de codificar el fotograma de
vídeo. Por lo tanto, al igual que con el ejemplo de macrobloque único, la compresión se completa cuando la
siguiente mejor MBC extraída del conjunto ordenado de conjuntos tiene un costo de bit que daría como resultado
45 un costo de bit de fotograma que excede el costo de bit de fotograma objetivo.

[0065] Este proceso proporciona una codificación óptima para un número fijo de bits, pero solo es viable si el
proceso de ordenar el conjunto de conjuntos de MBC es computable en tiempo real en un *hardware* de
procesamiento práctico. Para este proceso es crítico el uso de un conjunto diferente que proporcione el orden de
los conjuntos individuales de MBCs de macrobloque, donde este conjunto de conjuntos se reordena después de
50 la extracción de cada MBC en función de la mejor RIPB. Esto proporciona una reducción significativa en la
complejidad, ya que las MBCs para cada macrobloque se pueden ordenar en paralelo.

REIVINDICACIONES

- 5 1. Método para comprimir una matriz de entradas de datos en forma de señales electrónicas digitales, donde el método comprende los pasos de:
- (a) proporcionar una matriz de datos original que se va a comprimir;
- (b) dividir (42) la matriz de datos original en macrobloques y dividir (44) los macrobloques en bloques de uno o más tamaños, donde cada bloque puede ser el macrobloque completo o una división del macrobloque;
- 10 (c) calcular (46), para cada bloque de la matriz de datos original, una pluralidad respectiva de codificaciones comprimidas posibles usando una técnica de compresión diferente respectiva para cada codificación del respectivo bloque, donde cada codificación comprimida posible en las pluralidades respectivas de codificaciones comprimidas posibles tiene un respectivo tamaño de datos asociado y un respectivo error de reconstrucción;
- 15 (d) inicializar (50) una codificación comprimida de la matriz de datos original, donde la codificación comprimida de la matriz de datos original es una combinación de las posibles codificaciones comprimidas calculadas en el paso (c) y que, cuando se descomprime, corresponde a una matriz de datos de referencia, que es una aproximación inicial a la matriz de datos original, donde la codificación comprimida de la matriz de datos original tiene un primer tamaño de datos asociado y un primer error de reconstrucción, que representa una diferencia cuantitativa entre la matriz de datos original y la matriz de datos de referencia;
- 20 (e) generar una lista de las posibles codificaciones comprimidas calculadas en el paso (c) y seleccionar una de las posibles codificaciones comprimidas de la lista (52) que, cuando se añade a la codificación comprimida de la matriz de datos original, proporciona la mayor mejora relativa en el error de reconstrucción por aumento de unidad en el tamaño de los datos;
- 25 (f) en una primera iteración:
- (A) actualizar (54) la codificación comprimida de la matriz de datos original generada en el paso (d) añadiendo la codificación comprimida seleccionada en el paso (e), y
- 30 (B) actualizar la matriz de datos de referencia generada en el paso (d); y en iteraciones posteriores:
- (A) actualizar (54) la codificación comprimida de la matriz de datos original generada en la iteración anterior del paso (f) añadiendo la codificación comprimida en el paso (e), y
- 35 (B) actualizar la matriz de datos de referencia generada en la iteración anterior del paso (f);
- Y
- (g) repetir recursivamente los pasos (e) y (f) hasta alcanzar un tamaño de datos máximo dentro de un presupuesto de tamaño de datos específico para la codificación comprimida de la matriz de datos original (56).
- 40 2. Método según la reivindicación 1, donde el paso de seleccionar una de las posibles codificaciones comprimidas comprende seleccionar una segunda codificación con respecto a un bloque para el cual se ha seleccionado previamente una primera codificación en una iteración anterior.
- 45 3. Método según la reivindicación 2, donde el paso (f) de actualizar, en una primera iteración, la codificación comprimida de la matriz de datos original generada en el paso (d) y actualizar, en iteraciones posteriores, la codificación comprimida de la matriz de datos original generada en la iteración anterior del paso (f) comprende añadir la segunda codificación además de la primera codificación.
- 50 4. Método según la reivindicación 2, donde el paso (f) de actualizar, en una primera iteración, la codificación comprimida de la matriz de datos original generada en el paso (d) y actualizar, en iteraciones posteriores, la codificación comprimida de la matriz de datos original generada en la iteración anterior del paso (f) comprende añadir la segunda codificación en lugar de la primera codificación.
- 55 5. Método según cualquiera de las reivindicaciones anteriores, donde la lista de codificaciones comprimidas posibles comprende solo aquellas codificaciones comprimidas que se pueden proporcionar en una secuencia que tiene tanto un aumento en el tamaño de los datos como una disminución en el error de reconstrucción.
- 60 6. Método según cualquiera de las reivindicaciones anteriores, donde el paso de generar la lista de posibles codificaciones comprimidas comprende el paso de clasificar las posibles codificaciones para cada bloque en términos de aumento del tamaño de los datos y excluir de la lista cualquiera de las posibles codificaciones comprimidas para cualquier bloque dado que no produce una disminución en el error de reconstrucción para ese bloque para un aumento en el tamaño de los datos.
- 65 7. Método según cualquiera de las reivindicaciones anteriores, que comprende un paso de deduplicar la lista de las posibles codificaciones comprimidas para identificar codificaciones comprimidas que tienen el mismo tamaño

de datos y omitir cualquiera de las codificaciones comprimidas identificadas que tenga un error de reconstrucción mayor.

5 8. Método según cualquiera de las reivindicaciones anteriores, donde la lista de las posibles codificaciones comprimidas comprende solo aquellas codificaciones comprimidas que se pueden proporcionar en una lista que tiene un tamaño de datos que es menor que el tamaño de datos de la matriz original.

10 9. Método según cualquier reivindicación precedente, en la que el paso de generar una lista de las posibles codificaciones comprimidas comprende determinar una pluralidad de codificaciones comprimidas de macrobloques para cada macrobloque, donde cada codificación comprimida de macrobloques comprende un conjunto de codificaciones comprimidas para bloques en el macrobloque respectivo y el paso de seleccionar una de las posibles codificaciones comprimidas comprende seleccionar una de las codificaciones comprimidas de macrobloques.

15 10. Método según cualquiera de las reivindicaciones anteriores, donde la matriz de entradas de datos forma parte de un flujo de vídeo.

20 11. Método según cualquiera de las reivindicaciones anteriores, que comprende, además, el paso de transmitir la codificación comprimida a través de un canal de ancho de banda variable una vez que se ha alcanzado el tamaño de datos máximo.

25 12. Codificador de vídeo (2) configurado para enviar datos de vídeo a través de un canal, donde el codificador comprende: un compresor (16) configurado para comprimir datos de vídeo usando un factor de compresión ajustable y un transmisor (6) configurado para transmitir paquetes de datos construidos a través del canal, donde el compresor está configurado para:

- (a) recibir una matriz de datos original que se va a comprimir;
- 30 (b) dividir (42) la matriz de datos original en macrobloques y dividir (44) los macrobloques en bloques de uno o más tamaños, donde cada bloque puede ser el macrobloque completo o una división del macrobloque;
- (c) calcular (46), para cada bloque de la matriz de datos original, una pluralidad respectiva de codificaciones comprimidas posibles usando una técnica de compresión diferente respectiva para cada codificación del respectivo bloque, donde cada codificación comprimida posible en las respectivas pluralidades de codificaciones comprimidas posibles tiene un respectivo tamaño de datos asociado y un respectivo error de reconstrucción;
- 35 (d) inicializar (50) una codificación comprimida de la matriz de datos original, donde la codificación comprimida de la matriz de datos original es una combinación de las posibles codificaciones comprimidas calculadas en el paso (c) y que, cuando está descomprimida, corresponde a una matriz de datos de referencia, que es una aproximación inicial a la matriz de datos original, donde la codificación comprimida de la matriz de datos original tiene un primer tamaño de datos asociado y un primer error de reconstrucción, que representa una diferencia cuantitativa entre la matriz de datos original y la matriz de datos de referencia;
- 40 (e) generar una lista de las posibles codificaciones comprimidas calculadas en el paso (c) y seleccionar una de las posibles codificaciones comprimidas de la lista (52) que, cuando se añade a la codificación comprimida de la matriz de datos original, proporciona la mayor mejora relativa en el error de reconstrucción por aumento de unidad en el tamaño de los datos;
- 45 (f) en una primera iteración:

- (A) actualizar (54) la codificación comprimida de la matriz de datos original generada en el paso (d) añadiendo la codificación comprimida seleccionada en el paso (e), y
- 50 (B) actualizar la matriz de datos de referencia; y
- en iteraciones posteriores:

- (A) actualizar (54) la codificación comprimida de la matriz de datos original generada en la iteración anterior del paso (f) añadiendo la codificación comprimida seleccionada en el paso (e), y
- 55 (B) actualizar la matriz de datos de referencia generada en la iteración anterior del paso (f); y

(g) repetir recursivamente los pasos (e) y (f) hasta alcanzar un tamaño de datos máximo dentro de un presupuesto de tamaño de datos específico para la codificación comprimida de la matriz de datos original (56).

60 13. Medio de almacenamiento legible por ordenador no transitorio, que tiene un programa informático almacenado en el mismo, donde el programa informático comprende instrucciones legibles por procesador que, cuando se ejecutan, dirigen un dispositivo para realizar acciones que comprenden:

- 65 (a) proporcionar una matriz de datos original que se va a comprimir;
- (b) dividir (42) la matriz de datos original en macrobloques y dividir (44) los macrobloques en bloques de uno o más tamaños, donde cada bloque puede ser el macrobloque completo o una división del macrobloque;

- 5 (c) calcular (46), para cada bloque de la matriz de datos original, una pluralidad respectiva de codificaciones comprimidas posibles usando una respectiva técnica de compresión diferente para cada codificación del respectivo bloque, donde cada codificación comprimida posible en las pluralidades respectivas de codificaciones comprimidas posibles tiene un respectivo tamaño de datos asociado y un respectivo error de reconstrucción;
- 10 (d) inicializar (50) una codificación comprimida de la matriz de datos original, donde la codificación comprimida de la matriz de datos original es una combinación de las posibles codificaciones comprimidas calculadas en el paso (c) y que, cuando está descomprimida, corresponde a una matriz de datos de referencia, que es una aproximación inicial a la matriz de datos original, donde la codificación comprimida de la matriz de datos original tiene un primer tamaño de datos asociado y un primer error de reconstrucción, que representa una diferencia cuantitativa entre la matriz de datos original y la matriz de datos de referencia;
- 15 (e) generar una lista de las posibles codificaciones comprimidas calculadas en el paso (c) y seleccionar una de las posibles codificaciones comprimidas de la lista que, cuando se añade a la codificación comprimida de la matriz de datos original, proporciona la mayor mejora relativa en el error de reconstrucción por aumento de unidad en el tamaño de los datos;
- (f) en una primera iteración:
- 20 (A) actualizar (54) la codificación comprimida de la matriz de datos original generada en el paso (d) añadiendo la codificación comprimida seleccionada en el paso (e), y
(B) actualizar la matriz de datos de referencia generada en el paso (d); y
en iteraciones posteriores:
- 25 (A) actualizar (54) la codificación comprimida de la matriz de datos original generada en la iteración anterior del paso (f) añadiendo la codificación comprimida seleccionada en el paso (e), y
(B) actualizar la matriz de datos de referencia generada en la iteración anterior del paso (f); y
- 30 (g) repetir recursivamente los pasos (e) y (f) hasta alcanzar un tamaño de datos máximo dentro de un presupuesto de tamaño de datos específico para la codificación comprimida de la matriz de datos original (56).

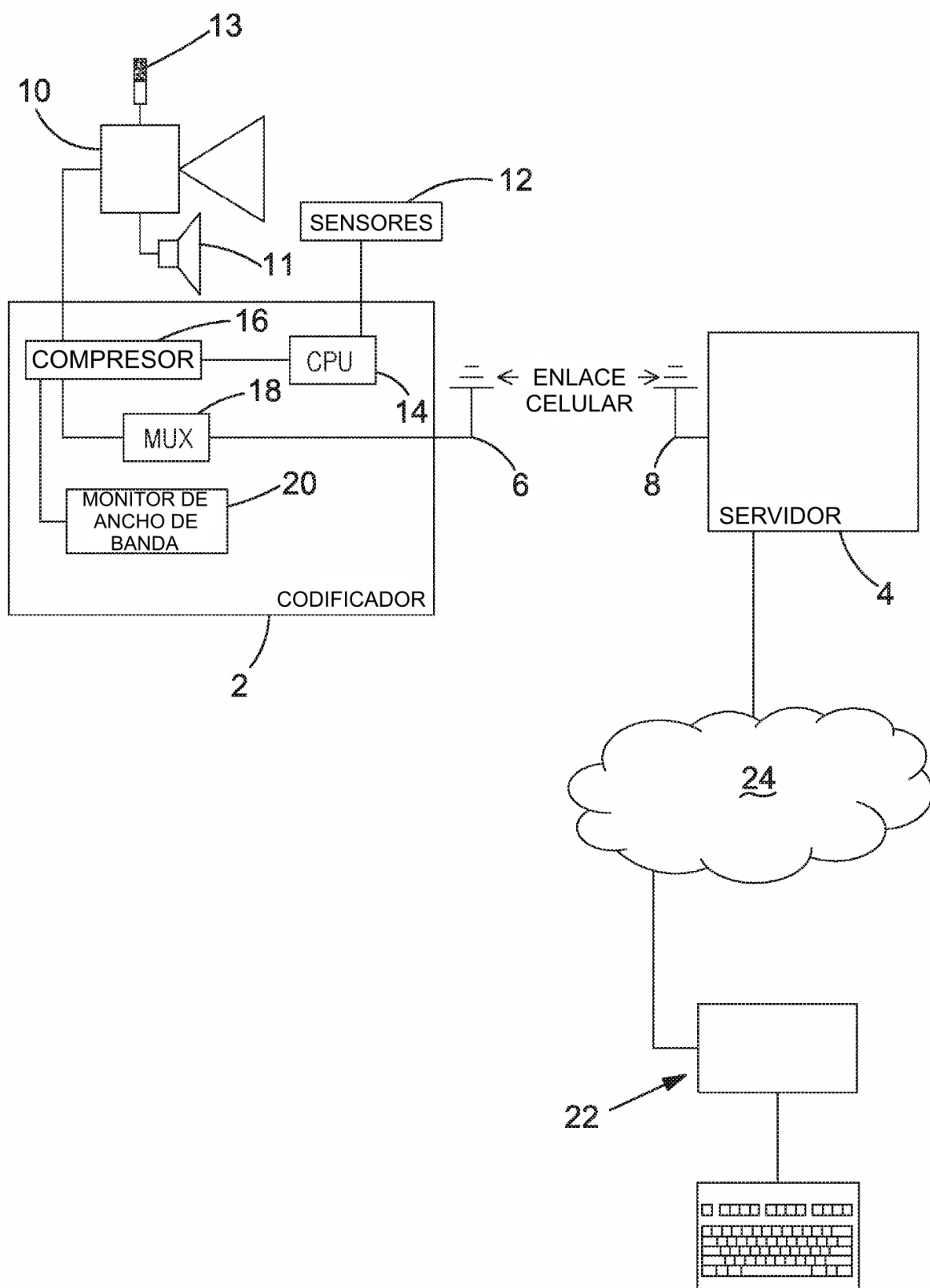


FIG.1

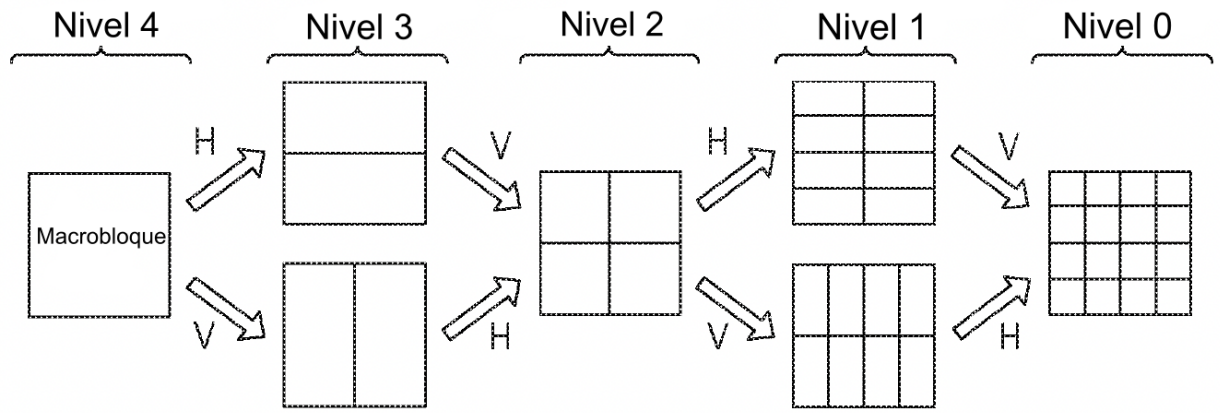


FIG.2

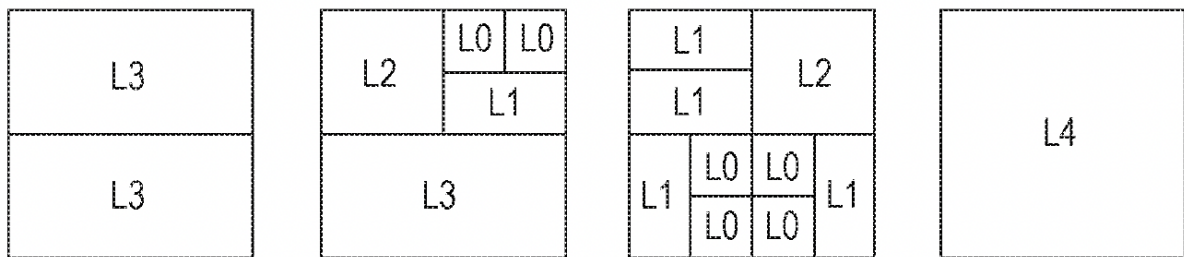


FIG.3

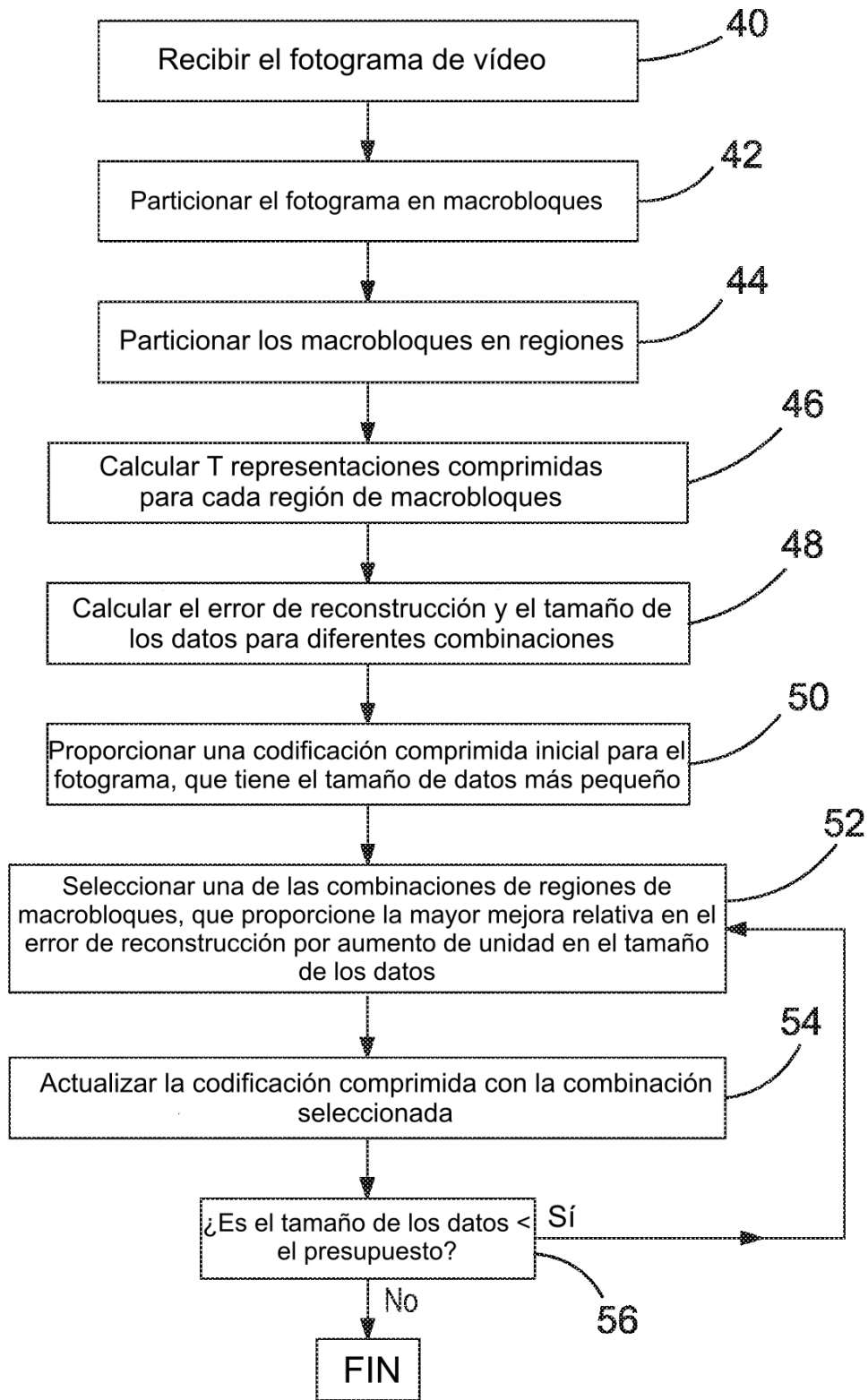


FIG.4