



19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 296 415**

51 Int. Cl.:  
**H04N 7/50** (2006.01)  
**H03M 7/40** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Número de solicitud europea: **99965061 .7**  
86 Fecha de presentación : **30.11.1999**  
87 Número de publicación de la solicitud: **1135934**  
87 Fecha de publicación de la solicitud: **26.09.2001**

54 Título: **Codificación eficaz de cabecero de macrobloque para vídeo compresión.**

30 Prioridad: **30.11.1998 US 201272**

45 Fecha de publicación de la mención BOPI:  
**16.04.2008**

45 Fecha de la publicación del folleto de la patente:  
**16.04.2008**

73 Titular/es: **MICROSOFT CORPORATION**  
**One Microsoft Way, Building 4**  
**Redmond, Washington 98052-6399, US**

72 Inventor/es: **Lin, Chih-Lung (Bruce) y**  
**Lee, Ming-Chieh**

74 Agente: **Carpintero López, Francisco**

ES 2 296 415 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**

Codificación eficaz de cabecero de macrobloque para vídeo compresión.

**5 Campo técnico**

La invención se refiere a codificación de vídeo, y específicamente, a un procedimiento mejorado para codificar parámetros de bloque utilizados en formatos de codificación de vídeo basados en cuadro y basados en objeto.

**10 Antecedentes**

Presentaciones de vídeo de movimiento completo basadas en señales analógicas de vídeo, han estado disponibles mucho tiempo en forma de televisión. Con los avances recientes en las capacidades y susceptibilidad de recursos en el procesamiento por ordenador, las presentaciones de vídeo de movimiento completo basadas en señales de vídeo digital están disponibles cada vez de forma más amplia. Los sistemas de vídeo digital pueden proporcionar mejoras significativas respecto a los sistemas convencionales de vídeo analógico en cuanto a la creación, modificación, transmisión, almacenamiento y reproducción de secuencias de vídeo de movimiento completo.

Las presentaciones de vídeo digital incluyen grandes números de cuadros de imagen que son reproducidos o que son emitidos sucesivamente a frecuencias de entre 30 y 75 Hz. Cada cuadro de imagen es incluso una imagen formada a partir de un conjunto de píxeles en base a la resolución de presentación de un sistema particular. Como ejemplos, los sistemas basados en VHS tienen resoluciones de presentación de 320 x 480 píxeles, los sistemas basados en NTSC tienen resoluciones de presentación de 720 x 846 píxeles, y los sistemas de televisión de alta definición (HDTV) en desarrollo, tienen resoluciones de presentación de 1360 x 1024 píxeles.

Las cantidades de información digital bruta incluida en las secuencias de vídeo, son enormes. El almacenamiento y la transmisión de estas cantidades de información de vídeo son impracticables con el equipamiento de ordenador personal convencional. Considérese, por ejemplo, una forma digitalizada de un formato de imagen de VHS de resolución relativamente baja, que tiene una resolución de 320 x 480 píxeles. Una película cinematográfica de una longitud total de dos horas de duración a esta resolución, corresponde a 100 gigabytes de información de vídeo digital. Por comparación, los discos ópticos compactos convencionales tienen capacidades de alrededor de 0,6 gigabytes, los discos duros magnéticos tienen capacidades de 1-2 gigabytes, y los discos ópticos compactos actualmente en desarrollo tienen capacidades de hasta 8 gigabytes.

Para dirigir las limitaciones en cuanto al almacenamiento o la transmisión de tales cantidades enormes de información de vídeo digital, se han establecido varios procesos o estándares de compresión de vídeo, incluyendo el MPEG-1, el MPEG-2 y el H.26X. Estas técnicas de compresión de vídeo utilizan semejanzas entre cuadros de imagen sucesivos, que se conocen como correlación temporal o inter-cuadro, para proporcionar una compresión inter-cuadro en la que se utilizan datos de movimiento y señales de error para codificar los cambios entre cuadros.

Adicionalmente, las técnicas de compresión de vídeo convencionales utilizan semejanzas dentro de los cuadros de imagen, conocidas como correlación espacial o intra-cuadro, para proporcionar compresión intra-cuadro en la que se comprimen las muestras de imagen del interior de un cuadro de imagen. La compresión intra-cuadro se basa en los procedimientos para comprimir imágenes fijas, tal como la codificación de transformada coseno discreta (DCT). Este tipo de codificación se menciona a veces como codificación de "textura" o de "transformación". Una "textura" se refiere en general a un conjunto de dos dimensiones de valores de muestra de imagen, tal como un conjunto de valores de crominancia y de luminancia, o un conjunto de valores alfa (opacidad). El término "transformación" se refiere en este contexto a cómo son transformadas las muestras de imagen en componentes de frecuencia espaciales durante el procedimiento de codificación. Esta utilización del término "transformación" debe ser diferenciada de una transformación geométrica utilizada para estimar los cambios de escena en algunos métodos de compresión inter-cuadro.

La compresión inter-cuadro utiliza típicamente estimación de movimiento y compensación para codificar cambios de escena entre cuadros. La estimación de movimiento consiste en un proceso para estimar el movimiento de muestras de imagen (por ejemplo, píxeles) entre cuadros. Con la utilización de estimación de movimiento, el codificador trata de emparejar bloques de píxeles de un cuadro con los píxeles correspondientes de otro cuadro. Después de que se ha encontrado el bloque más similar en un área de búsqueda dada, se aproxima y se representa el cambio de posición de las posiciones de píxel de los píxeles correspondientes, como datos de movimiento, tal como un vector de movimiento. La compensación de movimiento es un proceso para la determinación de una imagen pronosticada, y para el cálculo del error entre la imagen pronosticada y la imagen original. Con la utilización de compensación de movimiento, el codificador aplica los datos de movimiento a una imagen, y calcula la imagen pronosticada. La diferencia entre la imagen pronosticada y la imagen de entrada, se denomina señal de error. Puesto que la señal de error consiste justamente en un conjunto de valores que presentan la diferencia entre los valores de muestra de imagen, puede ser comprimida utilizando el mismo procedimiento de codificación de textura que el utilizado para la codificación intra-cuadro de las muestras de imagen.

Aunque difieren en sus implementaciones específicas, los estándares de compresión de vídeo MPEG-1, MPEG-2 y H.26X son similares en un número de aspectos. La descripción que sigue del estándar de compresión de vídeo MPEG-2 es aplicable en general a los otros.

## ES 2 296 415 T3

El MPEG-2 proporciona compresión inter-cuadro y compresión intra-cuadro, basadas en bloques cuadrados de conjuntos de píxeles de las imágenes de vídeo. Una imagen de vídeo se divide en bloques de muestra de imagen denominados macrobloques, que tienen dimensiones de 16 x 16 píxeles. En el MPEG-2, un macrobloque comprende cuatro bloques de luminancia (cada bloque es de 8 x 8 muestras de luminancia (Y)) y dos bloques de crominancia (un bloque de muestra 8 x 8 muestras, cada uno de ellos para Cb y Cr).

En el MPEG-2, la codificación inter-cuadro se realiza sobre macrobloques. Un codificador MPEG-2 realiza estimación y compensación de movimiento para calcular vectores de movimiento y señales de error de bloque. Para cada bloque  $M_N$  de un cuadro N de imagen, se realiza una búsqueda a través de la imagen de un cuadro N+1 de vídeo sucesivo siguiente, o de un cuadro N-1 de imagen inmediatamente precedente (es decir, bidireccionalmente), para identificar los bloques  $M_{N+1}$  o  $M_{N-1}$  respectivos más similares. La posición del bloque más similar con relación al bloque  $M_N$ , se codifica con un vector de movimiento (DX, DY). El vector de movimiento se utiliza a continuación para calcular un bloque de valores de muestra pronosticados. Estos valores de muestra pronosticados se comparan con el bloque  $M_N$  para determinar la señal de error de bloque. La señal de bloque se comprime utilizando un procedimiento de codificación de textura tal como codificación de transformada coseno discreta (DCT).

Se han propuesto técnicas de codificación de vídeo basadas en objeto, como perfeccionamiento respecto a los estándares de codificación convencionales basados en cuadro. En la codificación basada en objeto, características de imagen de forma arbitraria se separan de los cuadros de la secuencia de vídeo con la utilización de un procedimiento denominado "segmentación". Los objetos de vídeo o "segmentos", son codificados de forma independiente. La codificación basada en objeto puede mejorar la velocidad de compresión debido a que incrementa la correlación inter-cuadro entre objetos de vídeo de cuadros sucesivos. También resulta ventajoso para una diversidad de aplicaciones que requieren acceso a, y rastreo de, objetos en una secuencia de vídeo.

En los procedimientos de codificación de vídeo basados en objeto propuestos para el estándar MPEG-4, la forma, el movimiento y la textura de los objetos de vídeo son codificados de forma independiente. La forma de un objeto está representada por una máscara binaria o alfanumérica que define el contorno del objeto de forma arbitraria en un cuadro de vídeo. El movimiento de un objeto es similar a los datos de movimiento del MPEG-2, salvo en que se aplica a una imagen de configuración arbitraria del objeto que ha sido segmentado a partir de un cuadro rectangular. La estimación y compensación de movimiento se realiza sobre bloques de un "plano de objeto de vídeo" en vez del cuadro completo. El plano de objeto de vídeo es el nombre que se da a la imagen conformada de un objeto presente en un cuadro simple.

La textura de un objeto de vídeo consiste en la información de muestra de imagen de un plano de objeto de vídeo que cae dentro de la forma del objeto. La codificación de textura de las muestras de imagen del objeto y de las señales de error, se realiza utilizando procedimientos similares de codificación de textura como en la codificación basada en cuadro. Por ejemplo, una imagen segmentada puede ser acoplada en un rectángulo delimitador formado con macrobloques. La imagen rectangular formada por el rectángulo delimitador puede ser comprimida justamente como un cuadro rectangular, salvo en que los macrobloques transparentes no necesitan ser codificados. Los bloques parcialmente transparentes son codificados tras el relleno de las porciones del bloque que caen fuera del contorno de la configuración del objeto con valores de muestra según una técnica conocida como "padding" ("relleno").

Las técnicas de codificación basadas en cuadro, tales como la MPEG-2 y la H26X, y las técnicas de codificación basadas en objeto propuestas para la MPEG-4, son similares debido a que realizan la codificación intra-cuadro e inter-cuadro sobre macrobloques. Cada monobloque incluye una serie de parámetros de cabecera que proporcionan información acerca del macrobloque. Como ejemplo, la Figura 1 muestra parámetros de macrobloque utilizados en el encabezamiento de un macrobloque de inter-cuadro. El parámetro (10) COD es un sólo bit que indica si el macrobloque de inter-cuadro está codificado. En particular, este bit indica si el macrobloque codificado incluye o no datos de movimiento y datos de error codificados de textura. En casos en los que los datos de movimiento y de señal de error son cero, el bit COD reduce la información necesaria para codificar el macrobloque debido a que solamente se envía un único bit en vez de bits adicionales indicadores de que el vector de movimiento y los datos de textura no están codificados.

Adicionalmente al bit COD, la sintaxis de codificación para los macrobloques incluye parámetros de bloque codificados (CBP) que indican si se han transmitido los coeficientes de transformación codificados para crominancia y luminancia para el macrobloque. Si los coeficientes de transformación son todos cero para un bloque, entonces no hay necesidad de enviar datos de textura para el bloque. Los Parámetros de Bloque Codificados para Crominancia (CBPC), son dos bits que indican si los datos de textura codificados para cada uno de los dos bloques de crominancia han sido o no transmitidos.

Los bits CBPC están codificados junto con otra banderola que proporciona información acerca del tipo de cuantificación para el macrobloque. Estas banderolas se combinan para formar un parámetro denominado MCBPC (12), y MCBPC es entropía codificada con la utilización de un método de codificación de entropía tal como codificación de Huffman o aritmética.

El parámetro denominado AC\_Pred\_flag (14) es una banderola que indica si se ha utilizado predicción de AC en el macrobloque.

## ES 2 296 415 T3

El Patrón de Bloque Codificado para luminancia (CBPY) (16) comprende cuatro bits que indican si se han transmitido o no datos de textura codificados para cada uno de los cuatro bloques de crominancia. Al igual que el parámetro MCBPC, las banderolas CBPY son también entropía codificada utilizando cualquiera de entre codificación de Huffman o aritmética.

5

A continuación del parámetro CBPY, el macrobloque incluye datos de vector de movimiento codificado (mostrado como elemento 18 en la Figura 1). A continuación de los datos de vector de movimiento, el "dato de bloque" representa los datos de textura codificados para el macrobloque (mostrado como dato de bloque 20 en la Figura 1).

10

Un inconveniente de la alternativa de codificación ilustrada en la Figura 1 consiste en que codifica banderolas CBPC y CBPY por separado, y por lo tanto, no aprovecha la correlación entre estos parámetros para reducir el encabezamiento de macrobloque. Además, no se aprovecha de la dependencia espacial de los parámetros de bloque codificados.

15

El documento US-A-5 400 075 se refiere a un codificador/descodificador de longitud adaptativa variable, para codificación de longitud variable de la señal de vídeo comprimida.

### Sumario

20

El objeto de la invención consiste en proporcionar un procedimiento y un sistema mejorados de codificación en una aplicación de codificación de vídeo.

Este objeto ha sido resuelto por la invención según se reivindica en las reivindicaciones independientes.

25

Las realizaciones preferidas están definidas por las reivindicaciones dependientes.

30

La invención proporciona un procedimiento mejorado de codificación de parámetros de encabezamiento de macrobloque en aplicaciones de codificación de vídeo. Un aspecto de la invención consiste en un procedimiento de codificación que aprovecha la correlación entre parámetros de bloque codificados mediante la codificación conjunta de todos los parámetros de bloque codificados con un único código de longitud variable. Otro aspecto de la invención consiste en un procedimiento de codificación que se aprovecha de la dependencia espacial entre los patrones de bloque codificados de bloques contiguos.

35

En una implementación de la invención, los parámetros de bloque codificados para luminancia y crominancia de un macrobloque, se forman según un parámetro único, combinado, para el macrobloque. Se asigna un código de longitud variable al parámetro combinado a partir de una tabla de codificación de longitud variable. La tabla de codificación es ejercitada en base a una velocidad de bit objetivo (por ejemplo, aplicaciones de Internet de baja velocidad de bit) y de una clase objetiva de contenido de vídeo (por ejemplo, vídeo de cabezal parlante). Codificando conjuntamente los valores de luminancia y de crominancia, el codificador aprovecha la correlación entre estos parámetros en el macrobloque.

40

45

Para mejorar aún más la eficacia de la codificación, la implementación utiliza predicción para aprovecharse de la dependencia espacial de los parámetros de bloque codificados de bloques contiguos. Con anterioridad a asignar el código de longitud variable al parámetro combinado, se pronostican algunos de los parámetros de bloque codificados a partir de bloques contiguos. Para macrobloques intra-cuadro, por ejemplo, el codificador calcula un valor pronosticado espacialmente para cada parámetro de bloque codificado para luminancia. Este parámetro pronosticado espacialmente forma parte del parámetro combinado para el macrobloque.

50

Las características y ventajas adicionales de la invención resultarán más evidentes a partir de la descripción detallada que sigue y de los dibujos que se acompañan de una implementación de la invención.

### Breve descripción de los dibujos

55

La Figura 1 es un diagrama que ilustra un ejemplo de encabezamiento de macrobloque utilizado en un proceso de codificación de vídeo estándar;

la Figura 2 es un diagrama de bloques de un codificador de vídeo;

la Figura 3 es un diagrama de bloques de un descodificador de vídeo;

60

la Figura 4 es un diagrama que ilustra un ejemplo de encabezamiento de bloque mejorado en el que los parámetros de bloque codificados para crominancia y luminancia son codificados conjuntamente con un único código de longitud variable;

65

la Figura 5 es un diagrama de flujo que ilustra la forma en que una implementación de la invención calcula un código único de longitud variable para los parámetros de bloque codificados de macrobloques de cuadro I y P;

la Figura 6 es un diagrama que ilustra cuatro macrobloques, y sus correspondientes bloques de luminancia (Y);

la Figura 7 es un diagrama que muestra un ejemplo de los gradientes horizontal y vertical de valores de parámetro de bloque codificado para los bloques de luminancia seleccionados en la Figura 6;

la Figura 8 es un diagrama que ilustra un procedimiento para calcular un predictor para los parámetros de bloque codificados, y

la Figura 9 es un diagrama de un sistema de ordenador que sirve como entorno operativo para una implementación de software de la invención.

## 10 Descripción detallada

### *Introducción*

La primera sección que sigue, proporciona una descripción de un codificador y un decodificador de vídeo. Las secciones siguientes describen cómo mejorar la codificación de parámetros de encabezamiento de macrobloque mediante aprovechamiento de la correlación entre parámetros CBPC y CBPY, y aprovecharse de la dependencia espacial de parámetros de bloque codificados de bloques contiguos.

Útil para la codificación de vídeo tanto en base a cuadro como en base a objeto, la invención mejora la codificación de parámetros de macrobloque, en caso de que los macrobloques sean componentes de objetos de vídeo arbitrarios segmentados a partir de una secuencia de cuadros, como en el caso de que sean cuadros de imagen de forma rectangular. La codificación basada en objeto utiliza módulos de codificación de movimiento y textura similares a los utilizados en la codificación basada en cuadro. Adicionalmente, los codificadores basados en objeto incluyen también módulos de codificación de forma. La sintaxis de bloque relevante para la invención es similar tanto en codificación basada en cuadro como en codificación basada en objeto. Mientras que el codificador y el decodificador descritos en la sección que sigue están basados en objeto, los mismos proporcionan una base suficiente para explicar cómo implementar la invención en ambos esquemas de codificación en base a cuadro y en base a objeto.

### *Descripción de un Ejemplo de Codificador y de Decodificador*

La Figura 2 es un diagrama de bloques que ilustra una implementación de un codificador de vídeo basado en objeto. La entrada 30 al codificador incluye una serie de objetos, información de su forma, y rectángulos delimitadores. La información de forma está disponible, por lo tanto, con anterioridad a que el codificador realice la codificación de los datos de textura o de movimiento. La codificación basada en cuadro difiere en que se codifica el cuadro completo sin información de forma.

El módulo 32 de codificación de forma recibe la definición de un objeto incluyendo su rectángulo delimitador, y extiende su rectángulo delimitador a múltiplos enteros de macrobloques. La información de forma para un objeto comprende una máscara o "plano alfa". El módulo 32 de codificación de forma lee esta máscara y la comprime, utilizando por ejemplo un procedimiento convencional de codificación en cadena para codificar el contorno del objeto.

El módulo 34 de estimación de movimiento lee un objeto incluyendo su rectángulo delimitador y una imagen 36 previamente reconstruida, y calcula datos de estimación de movimiento utilizados para pronosticar el movimiento del objeto de un cuadro a otro. Tras la identificación de los macrobloques en la imagen de objeto actual, el módulo 34 de estimación de movimiento busca el macrobloque más similar en la imagen reconstruida para cada macrobloque de la imagen objeto actual, para calcular los datos de movimiento para cada macrobloque. El formato específico de los datos de movimiento a partir del módulo 34 de estimación de movimiento, puede variar dependiendo del método de estimación de movimiento utilizado. La implementación descrita en lo que sigue calcula un vector de movimiento para cada macrobloque, que es conforme con los formatos MPEG y H26X actuales.

El módulo 38 de compensación de movimiento lee los vectores de movimiento calculados por el módulo de estimación de movimiento y la imagen 36 previamente reconstruida, y calcula una imagen pronosticada para el cuadro actual. El codificador encuentra la diferencia entre los valores de muestra de imagen en el bloque de imagen de entrada según se especifica en la entrada 30, y los valores de muestra correspondientes en el bloque de imagen pronosticada según se calcula en el módulo 38 de compensación de movimiento, para determinar la señal de error para el macrobloque.

El módulo 40 de codificación de textura comprime esta señal de error para objetos codificados de inter-cuadro, y comprime valores de muestra de imagen para el objeto a partir de la corriente 30 de datos de entrada para los objetos codificados de intra-cuadro. La trayectoria 42 de realimentación procedente del módulo 40 de codificación de textura, representa la señal de error descodificada. El codificador utiliza los macrobloques de señal de error junto con los macrobloques de imagen pronosticada procedentes del módulo de compensación de movimiento, para calcular la imagen 36 previamente reconstruida.

El módulo 40 de codificación de textura codifica bloques de datos de intra-cuadro y de señal de error para un objeto, utilizando cualquier diversidad de técnicas de compresión de imagen fija. Ejemplos de técnicas de compresión incluyen técnicas basadas en transformación tales como codificación DCT y de tren de ondas, así como otros métodos convencionales de compresión de imagen tales como codificación de Pirámide Laplaciana.

## ES 2 296 415 T3

La corriente de bits de la secuencia de vídeo comprimida incluye información codificada de forma, de movimiento y de textura, procedente de los módulos de codificación de forma, de estimación de movimiento y de codificación de textura. El multiplexor 44 combina y formatea estos datos con la sintaxis apropiada, y los presenta a la salida para la memoria intermedia 46.

Mientras que el codificador puede ser implementado en software o en hardware, es más probable que sea implementado en software. En una implementación de software, los módulos del codificador representan instrucciones de software almacenadas en la memoria de un ordenador y ejecutadas en el procesador, y datos de vídeo almacenados en memoria. Un codificador de software puede ser almacenado y distribuido en una diversidad de medios convencionales legibles con ordenador. En implementaciones de hardware, los módulos de codificador se implementan en lógica digital, preferentemente en un circuito integrado. Algunas de las funciones del codificador pueden ser optimizadas en dispositivos lógicos digitales de propósito específico, en un periférico de ordenador para descargar la carga de procesamiento desde un ordenador anfitrión.

La Figura 3 es un diagrama de bloques que ilustra un descodificador para un procedimiento de codificación de vídeo basado en objeto. Un demultiplexor 60 recibe una corriente de datos 62 que representa una secuencia de vídeo comprimida, y separa los datos codificados de formas, de movimiento y de textura sobre un objeto, en base al objeto. El módulo 64 de descodificación de forma, descodifica la forma o el contorno en relación con el objeto actual que está siendo procesado. Para realizar esto, emplea un descodificador de forma que implementa la inversa del procedimiento de codificación de forma que se ha utilizado en el codificador de la Figura 2. Los datos de forma resultantes consisten en una máscara, tal como un plano alfa binario o un plano alfa de escala de grises que representa la forma del objeto.

El módulo 66 de descodificación de movimiento descodifica la información de movimiento de la corriente de datos. La información de movimiento descodificada incluye datos de movimiento tal como vectores de movimiento para bloques de macrobloques o coeficientes de transformación geométrica, dependiendo del tipo de método de estimación utilizado en el codificador. El módulo 66 de descodificación de movimiento suministra esta información de movimiento al módulo 68 de compensación de movimiento, y el módulo 68 de compensación de movimiento aplica los datos de movimiento a los datos 70 de objeto previamente reconstruido.

El módulo 74 de descodificación de textura descodifica señales de error para datos de textura codificados inter-cuadro, y un conjunto de valores de color para datos de textura intra-cuadro, y pasa esta información a un módulo 72 para el cálculo y acumulación de la imagen reconstruida. Para objetos codificados inter-cuadro, este módulo 72 aplica los datos de señal de error a la salida de imagen pronosticada procedente del módulo de compensación de movimiento, para calcular el objeto reconstruido para el cuadro actual. Para objetos codificados intra-cuadro, el módulo de descodificación 74 descodifica los valores de muestra de imagen para el objeto, y sitúa el objeto reconstruido en el módulo 72 de objeto reconstruido. Los objetos reconstruidos previamente son almacenados en la memoria 70 de objeto, y se utilizan para construir el objeto para otros cuadros.

Al igual que el codificador, el descodificador puede ser implementado en hardware, software, o mediante una combinación de ambos. En implementaciones de software, los módulos del descodificador son instrucciones de software almacenadas en la memoria de un ordenador y ejecutadas por el procesador, y datos de vídeo almacenados en memoria. Un descodificador de software puede estar almacenado y distribuido en una diversidad de medios convencionales legibles con ordenador. En implementaciones de hardware, los módulos de descodificador se implementan en lógica digital, preferentemente en un circuito integrado. Algunas de las funciones del descodificador pueden ser optimizadas en dispositivos lógicos digitales de propósito específico en un periférico de ordenador, para descargar la carga de procesamiento desde un ordenador anfitrión.

### *Codificación Mejorada de Encabezamiento de Macrobloque*

La invención incluye innovaciones que mejoran la codificación de parámetros de encabezamiento de macrobloque. Una innovación consiste en un método para la codificación de los parámetros de bloque codificados para aprovechar la correlación entre CBPC y CBPY. La innovación se implementa codificando conjuntamente un parámetro combinado de CBPC y CBPY con un código simple de longitud variable. Otra innovación mejora aún más la eficiencia de codificación de los parámetros de encabezamiento mediante el aprovechamiento de la dependencia espacial de los parámetros de bloque codificados. En particular, los parámetros de bloque codificados son comprimidos más eficazmente pronosticándolos a partir del parámetro de los bloques contiguos.

La Figura 4 es un diagrama que ilustra parámetros de bloque de encabezamiento calculados por una implementación de la invención. Al igual que la información de encabezamiento mostrada en la Figura 1, este bloque de encabezamiento incluye un parámetro COD 80, uno AC\_Pred\_flag 82, datos de vector de movimiento (MV 84), y datos 86 de bloque. A diferencia con el encabezamiento de la Figura 1, los parámetros MCBPC y CBPY son codificados conjuntamente con un código simple de longitud variable, denominado MBCBCPY 88. Este código combina los parámetros de bloque codificados para crominancia y luminancia, así como la banderola para el tipo de macrobloque.

La Figura 5 es un diagrama de flujo que ilustra cómo la implementación genera un código de longitud variable para cuadros Intra (I) y cuadros pronosticados (P). En esta implementación particular, los bloques de encabezamiento para cuadros I y P son codificados de manera diferente. Para los cuadros I, el codificador realiza la etapa adicional de predecir los parámetros de bloque codificados para luminancia con anterioridad a seleccionar el código de longitud

## ES 2 296 415 T3

variable. También es posible usar predicción para los cuadros P. Sin embargo, la predicción no mejora la eficacia de codificación significativamente en los cuadros P, y en algunos casos, puede incluso reducir la eficacia de codificación.

El objetivo de usar predicción para parámetros de bloque codificados consiste en producir tantos valores cero para estos parámetros como sea posible. Haciendo cero la mayor parte de los valores, el codificador reduce la varianza de los parámetros de bloque codificados. El proceso de ejercitación de la tabla de codificación de longitud variable puede favorecer entonces el valor cero, lo que mejora la eficacia de codificación. En cuadros P, especialmente en aplicaciones de baja velocidad de bit, los parámetros de bloque codificados son cero en su mayor parte con anterioridad a la predicción. Como tal, la predicción no tiende a incrementar el número de valores cero, y a veces, incluso reduce el número de valores cero. Por lo tanto, la implementación mostrada en la Figura 5 no utiliza predicción para los cuadros P.

Para cuadros P, el codificador empieza encontrando los parámetros de bloque codificados para luminancia y crominancia como se muestra en la etapa 100. Los parámetros de bloque son, cada uno de ellos, un sólo bit que indica si un bloque correspondiente está codificado en textura. Los parámetros de bloque codificados son calculados en el módulo de codificación de textura (40 en la Figura 2), lo que establece una banderola de bloque codificado para cada bloque que tenga valores de textura codificados que no sean cero. A la inversa, el valor del parámetro de bloque codificado para un bloque en el que los valores de textura son todos cero (o tan próximos a cero como para ser despreciables), es cero.

Puesto que existen dos bloques para crominancia (cada uno para los bloques U y V de 8 por 8 píxeles) y cuatro bloques para luminancia (cada uno para los cuatro bloques de 8 por 8) en el macrobloque, el parámetro combinado para el patrón de bloque codificado consiste en un total de seis bits. Combinando este número de 6 bits con el bit simple para el tipo de macrobloque, el codificador forma un número de 7 bits como se muestra en la etapa 102. El tipo de macrobloque indica si el macrobloque es para un cuadro I o para uno P.

Una vez que se ha formado el MBCBPCY combinado, el parámetro combinado se busca en una tabla de codificación de longitud variable, para encontrar un código de longitud variable correspondiente asociado al parámetro, como se muestra en la etapa 104. El codificador asigna un código simple de longitud variable al parámetro MBCPCY combinado.

La tabla de codificación en la tabla de implementación, es una tabla de codificación de Huffman. La tabla se ejercita preferentemente en base a la velocidad de objetivo y al escenario de objetivo. La tabla 1 que sigue es una tabla de Codificación de Longitud Variable (VLC) obtenida para un escenario de "cabezal parlante" de velocidad de bit. Para cada macrobloque de un cuadro P, la información MBCBPCY combinada es codificada con la utilización de la palabra de código para la entrada correspondiente en esta tabla.

TABLA 1

*Tabla VLC para patrón de bloque Codificado de luminancia y crominancia para imagen P*

Índice	Tipo MB	CBPCY Y(1234)UV	Número de bits	Código
0	I	000000	7	1000000
1	I	000001	13	1001111001001
2	I	000010	12	100111111101
3	I	000011	15	000000111111100
4	I	000100	12	100111111100
5	I	000101	18	000000101010000011
6	I	000110	17	10010110100110100
7	I	000111	16	1000001110111100
8	I	001000	12	100000111010
9	I	001001	17	00000011111111000
10	I	001010	16	0000001111111101
11	I	001011	16	0000001111111111
12	I	001100	13	0000001111001
13	I	001101	18	000000101010000010
14	I	001110	16	1001011010011101
15	I	001111	16	0000001010100100

ES 2 296 415 T3

16		010000	12	100101111000
17		010001	17	00000010101000011
18		010010	15	1000001110111111
19		010011	17	00000011111111001
20		010100	13	1001011110011
21		010101	18	100101101001101011
22		010110	18	100101111011111001
23		010111	16	0000001111111010
24		011000	14	10000011101110
25		011001	20	10010110100110101011
26		011010	16	1001011010011100
27		011011	18	100101111011111000
28		011100	13	1001011010010
29		011101	18	000000101010000101
30		011110	16	1001011010011110
31		011111	15	100101111001000
32		100000	12	000000111101
33		100001	17	100101111011111111
34		100010	16	0000001010100010
35		100011	16	1001011010011111
36		100100	14	10010111101110
37		100101	21	100101101001101010101
38		100110	17	10010111101111101
39		100111	17	10010111101111110
40		101000	12	100111100101
41		101001	18	000000101010000001
42		101010	19	1001011010011010100
43		101011	16	1000001110111101
44		101100	13	0000001111000
45		101101	16	1001011010011011
46		101110	16	0000001111111110
47		101111	16	0000001010100101
48		110000	13	0000001111110
49		110001	18	000000101010000000
50		110010	16	0000001010100011
51		110011	16	0000001111111011
52		110100	13	1000001110110
53		110101	18	000000101010000100
54		110110	15	000000101010011
55		110111	15	100101111001001
56		111000	13	0000001010101
57		111001	21	100101101001101010100
58		111010	15	100101111011110
59		111011	14	10010111100101
60		111100	10	1001011011
61		111101	15	100101101001100



ES 2 296 415 T3

5

10

15

20

25

30

35

40

45

50

55

60

65

62	I	111110	12	100101101011
63	I	111111	12	100101101010
64	P	000000	2	01
65	P	000001	7	000000
66	P	000010	6	100110
67	P	000011	9	100101011
68	P	000100	3	111
69	P	000101	10	1000001111
70	P	000110	9	000000100
71	P	000111	12	000000101000
72	P	001000	3	110
73	P	001001	10	1000001010
74	P	001010	9	100101000
75	P	001011	12	000000101011
76	P	001100	5	10001
77	P	001101	11	00000011011
78	P	001110	9	100111010
79	P	001111	11	10011111111
80	P	010000	4	0011
81	P	010001	10	1001110111
82	P	010010	9	100000110
83	P	010011	12	100000111001
84	P	010100	4	1011
85	P	010101	10	1001111011
86	P	010110	9	100101100
87	P	010111	11	10010111111
88	P	011000	6	001001
89	P	011001	12	000000110101
90	P	011010	10	1001111110
91	P	011011	13	1001111001000
92	P	011100	6	000001
93	P	011101	11	10010101010
94	P	011110	10	1000001000
95	P	011111	12	000000101001
96	P	100000	4	0001
97	P	100001	10	1001010100
98	P	100010	9	100101110
99	P	100011	12	100000111000
100	P	100100	6	100100
101	P	100101	11	10011110011
102	P	100110	10	1001110110
103	P	100111	13	1001011110110
104	P	101000	5	00001
105	P	101001	10	1001111010
106	P	101010	9	100111110
107	P	101011	12	000000111110
108	P	101100	6	001000
109	P	101101	11	10000010011
110	P	101110	10	0000001100
111	P	101111	11	10010111110
112	P	110000	5	10100
113	P	110001	11	10000010010
114	P	110010	10	1001010011
115	P	110011	12	100101111010
116	P	110100	6	100001
117	P	110101	11	10010101011

118	P	110110	10	1000001011
119	P	110111	12	000000110100
120	P	111000	5	10101
121	P	111001	10	1001111000
122	P	111010	10	1001010010
123	P	111011	12	100101101000
124	P	111100	5	00101
125	P	111101	10	0000001011
126	P	111110	8	10011100
127	P	111111	10	0000001110

En la implementación mostrada en la Figura 5, los cuadros I son codificados de una manera diferente a los cuadros P, debido a que el codificador utiliza predicción para aprovechar la dependencia espacial de los parámetros de bloque codificados. Para cada macrobloque, el codificador empieza por obtener los parámetros de bloque codificados para crominancia y luminancia, como se muestra en la etapa 106.

A continuación, el codificador calcula el predictor de los parámetros de bloque codificados para luminancia. En esta implementación particular, el codificador utiliza solamente predicción para los parámetros CBPY. Sin embargo, podría utilizarse también el mismo método de predicción para pronosticar los parámetros de bloque codificados para crominancia. En el caso de la crominancia, la predicción se calcula en base a bloques de crominancia de 8 por 8 píxeles en macrobloques contiguos en vez de en los bloques de luminancia contiguos de 8 por 8 píxeles, lo que puede ocurrir en el mismo macrobloque o en un macrobloque contiguo. Puesto que cada macrobloque tiene cuatro bloques de luminancia, los bloques contiguos para un bloque de luminancia dado pueden proceder del mismo macrobloque o de uno contiguo. Para una predicción que incluya bloques de crominancia, los bloques contiguos proceden de macrobloques contiguos.

El codificador realiza predicción espacial sobre parámetros de bloque codificados. En primer lugar busca en los parámetros de bloque codificados para bloques contiguos, con el fin de determinar si el valor del parámetro de bloque es posible que cambie de un bloque contiguo al bloque actual de interés. Si se puede identificar la posición de un bloque que representa el cambio más pequeño en el parámetro de bloque codificado (es decir, el gradiente espacial más bajo de los parámetros de bloque codificados), entonces el parámetro de bloque codificado para el bloque en esa posición se utiliza como predictor. En otro caso, no importa qué contiguo se elija como predictor, y se puede seleccionar simplemente uno. Un ejemplo específico de selección del predictor va a ser descrito e ilustrado con mayor detalle con referencia a las Figuras 6-8 que siguen.

En la siguiente etapa 110, el codificador calcula un valor pronosticado para los parámetros de bloque codificados. El valor pronosticado representa el cambio en el parámetro de bloque codificado respecto al bloque predictor y al bloque actual. Para calcular el valor pronosticado, el codificador realiza una OR exclusiva (XOR) en forma de bit, sobre el valor pronosticado y el valor de bloque actual. Al vector resultante, denominado CBPCY\_XOR, se le asigna a continuación un código de longitud variable desde la tabla de Huffman. El codificador busca la entrada para CPCY\_XOR en la tabla y halla el correspondiente código de longitud variable. La tabla 2 que sigue muestra la tabla VLC utilizada para codificar valores CBPCY pronosticados para cuadros I en la implementación.

TABLA 2

Tabla VLC para patrón de bloque Codificado de crominancia y luminancia para imagen I

Índice	CBPCY XOR Y(1234)UV	Número de bits	Código
0	000000	1	1
1	000001	6	010111
2	000010	5	01001
3	000011	5	00101
4	000100	5	00110
5	000101	9	001000111
6	000110	7	0100000
7	000111	7	0010000
8	001000	5	00010
9	001001	9	001111100
10	001010	7	0111010
11	001011	7	0011101
12	001100	6	00010
13	001101	9	011101100
14	001110	8	01110111

ES 2 296 415 T3

	15	001111	8	0000000
	16	010000	5	00011
5	17	010001	9	010110111
	18	010010	7	0101100
	19	010011	7	0010011
	20	010100	6	000001
10	21	010101	10	0101101000
	22	010110	8	01000110
	23	010111	8	00111111
	24	011000	6	011110
15	25	011001	13	0011100010010
	26	011010	9	010110101
	27	011011	8	01000010
20	28	011100	7	0100010
	29	011101	11	00111000101
	30	011110	10	0100011110
	31	011111	9	010000111
25	32	100000	4	0110
	33	100001	9	000000011
	34	100010	7	0011110
	35	100011	6	011100
30	36	100100	7	0010010
	37	100101	12	001110001000
	38	100110	9	001000100
35	39	100111	9	001110000
	40	101000	6	011111
	41	101001	11	01000111110
	42	101010	8	00111001
40	43	101011	9	010001110
	44	101100	7	0000001
	45	101101	11	00111000110
	46	101110	9	010110110
45	47	101111	9	001000101
	48	110000	6	010100
	49	110001	11	01000111111
	50	110010	9	001111101
50	51	110011	9	000011000
	52	110100	7	0000111
	53	110101	11	00111000111
55	54	110110	9	010000110
	55	110111	9	000011001
	56	111000	6	010101
	57	111001	10	0111011011
60	58	111010	9	000000010
	59	111011	9	001000110
	60	111100	8	00001101

65

61	111101	13	0011100010011
62	111110	10	0111011010
63	111111	10	0101101001

5

Las Figuras 6-8 ilustran la predicción espacial realizada en el codificador, con mayor detalle. La Figura 6 es un diagrama que muestra cuatro macrobloques contiguos (parte superior izquierda - 120, parte superior derecha - 122, parte inferior izquierda - 124, y parte inferior derecha - 126). El ejemplo que sigue está enfocado al bloque inferior derecho, el cual está enmarcado. Cada uno de los macrobloques incluye cuatro bloques de 8 por 8 píxeles para luminancia, etiquetados como Y1, Y2, Y3 e Y4.

10

Como ejemplo, considérese el bloque Y1 de luminancia de la parte superior izquierda para el macrobloque 126. Los bloques utilizados para calcular el predictor están enmarcados por medio de una línea 128 de puntos. El bloque de interés es Y1 (etiquetado como bloque 130a), y los bloques utilizados para calcular el predictor son los bloques contiguos etiquetados como 132a, 134a, y 136a.

15

Para dar un ejemplo específico, la Figura 7 muestra valores de los parámetros patrón de bloque codificados para cada uno de los bloques del interior de la línea punteada de la Figura 6. Los números de referencia 130b, 132b, 134b y 136b corresponden a los bloques 130a, 132a, 134a y 136a de la Figura 6, respectivamente. Los gradientes espaciales de los parámetros codificados de bloques contiguos se utilizan para seleccionar el predictor. En particular, el gradiente vertical se calcula a partir de los parámetros de bloque codificados de los bloques (136a, 132a, contiguos de la izquierda y de la parte superior izquierda, que se muestran enmarcados con 140 en la Figura 7). El gradiente horizontal se calcula a partir de los parámetros de bloque codificados de los bloques (136a, 130a, que se muestran enmarcados con 142 en la Figura 7).

20

La Figura 8 es un diagrama de flujo que ilustra las etapas para hallar el predictor. En primer lugar, el codificador encuentra los gradientes vertical y horizontal. Cada uno de ellos se calcula como la OR exclusiva de los parámetros de bloque codificados que se muestran enmarcados en la Figura 7 (140 es el gradiente vertical y 142 es el gradiente horizontal). A continuación, el codificador compara los valores de gradiente. Si los gradientes no son iguales, el codificador elige el predictor como el valor asignado al bloque en la dirección del gradiente inferior. En el ejemplo mostrado en la Figura 7, el gradiente vertical es cero, mientras que el gradiente horizontal es uno. De ese modo, la dirección del gradiente inferior es ascendente. Como tal, el valor del parámetro de bloque codificado para el bloque 134a se utiliza como predictor debido a que está situado en dirección "ascendente" con relación al bloque de interés.

25

Si se usa o no predicción para modificar los parámetros de bloque codificados, el resultado final es un código simple de longitud variable que representa todos los parámetros de bloque codificados para el macrobloque. Puesto que los cuadros I y P están codificados de manera diferente en la implementación, el descodificador trata los macrobloques respecto a estos cuadros de una manera diferente. Para los cuadros P, el descodificador usa la tabla VLC 1 para buscar el código simple de longitud variable y encontrar la entrada correspondiente que almacena el parámetro combinado que representa los parámetros de bloque codificados para luminancia y crominancia. Para los cuadros I, el descodificador usa la tabla VLC 2 para buscar el código simple de longitud variable y encontrar la entrada correspondiente que almacena el parámetro combinado que representa los parámetros de bloque codificados para luminancia y crominancia. En ambos cuadros I y P, el módulo de descodificación de textura (bloque 74 en la Figura 3) utiliza los parámetros de bloque codificados para determinar si los datos de textura para el bloque correspondiente necesitan ser descodificados. El descodificador omite descodificar la textura para bloques que tengan un parámetro de bloque codificado que sea cero.

30

En casos en los que los parámetros de bloque codificados son también pronosticados, el descodificador utiliza los parámetros de bloques descodificados previamente a partir de los bloques contiguos, para calcular el parámetro de bloque codificado para el bloque actual de interés. En primer lugar, el descodificador calcula la posición del bloque predictor en base a los gradientes espaciales de la misma manera que en el codificador. A continuación calcula el valor del parámetro de bloque codificado para el bloque actual, calculando la OR exclusiva del valor descodificado y el parámetro de bloque codificado del bloque predictor (el operador OR exclusiva tiene la siguiente propiedad:  $X \text{ XOR } Y = Z$ ;  $Z \text{ XOR } X = Y$ ). Después de esta etapa de predicción inversa, el descodificador de textura utiliza entonces el parámetro de bloque codificado para determinar si omite la descodificación de textura para el bloque.

35

### *Breve Compendio de un Sistema de Ordenador*

La Figura 9 y la discusión que sigue, han sido previstas a efectos de proporcionar una descripción breve, general, de un entorno de computación adecuado en el que puede ser implementada la invención. Aunque la invención o los aspectos de la misma pueden ser implementados en un dispositivo de hardware, el codificador y el descodificador descritos en lo que antecede se implementan en instrucciones ejecutables con ordenador, organizadas en módulos de programa. Los módulos de programa incluyen las rutinas, los programas, los objetos, los componentes y las estructuras de datos que realizan las tareas, e implementan los tipos de datos descritos en lo que antecede.

40

Mientras que la Figura 9 muestra una configuración típica de un ordenador de sobremesa, la invención pueden ser implementada en otras configuraciones de sistema de ordenador, incluyendo los dispositivos portátiles, los sistemas de microprocesador, los dispositivos electrónicos basados en microprocesador o programables por el usuario, los

microordenadores, los ordenadores principales, y similares. La invención puede ser utilizada también en entornos de cálculo distribuido en los que las tareas se realizan mediante dispositivos de procesamiento remoto que están enlazados a través de una red de comunicaciones. En un entorno de cálculo distribuido, los módulos de programa pueden estar situados en dispositivos de almacenamiento en memoria tanto locales como remotos.

5 La Figura 9 ilustra un sistema de ordenador que sirve como entorno operativo para la invención. El sistema de ordenador incluye un ordenador 920 personal, que incluye una unidad 921 de procesamiento, una memoria 922 de sistema, y un bus 923 de sistema que interconecta diversos componentes de sistema, incluyendo la memoria de sistema, con la unidad 921 de procesamiento. El bus de sistema, puede comprender cualquiera de los diversos tipos de estructuras de bus incluyendo un bus de memoria o controlador de memoria, un bus periférico, y un bus local utilizando una arquitectura de bus tal como PCI, VESA, Microchannel (MCA), ISA y EISA, por nombrar unos pocos. La memoria de sistema incluye la memoria 924 de sólo lectura (ROM) y la memoria 925 de acceso aleatorio (RAM). Un sistema 926 básico de entrada/salida (BIOS), que contiene las rutinas básicas que ayudan a transferir información entre elementos del interior del ordenador 920 personal, tal como durante el arranque, se encuentra almacenado en la ROM 924. El ordenador 920 personal incluye además una unidad 927 de disco duro, una unidad 928 de disco magnético, por ejemplo, para leer desde o escribir en un disco 929 extraíble, y una unidad 930 de disco óptico, por ejemplo, para la lectura de un disco 931 CD-ROM o para leer desde o escribir en otro medio óptico. La unidad 927 de disco duro, la unidad 928 de disco magnético, y la unidad 930 de disco óptico, están conectadas al bus 923 de sistema por medio de una interfaz 932 de unidad de disco duro, una interfaz 933 de unidad de disco magnético, y una interfaz 934 de unidad de disco óptico, respectivamente. Las unidades y sus medios asociados susceptibles de lectura con ordenador, proporcionan un almacenamiento no volátil de datos, de estructuras de datos, de instrucciones ejecutables por ordenador (un código de programa tal como librerías de enlace dinámico, y archivos ejecutables), etc., para el ordenador 920 personal. Aunque la descripción de medios susceptibles de ser leídos con ordenador que antecede se refiere a un disco duro, un disco magnético y un CD extraíbles, también pueden incluir otros tipos de medios que sean susceptibles de lectura mediante un ordenador, tal como casetes magnéticos, tarjetas de memoria flash, discos de vídeo digital, cartuchos de Bernouilli, y similares.

Un número de módulos de programa pueden encontrarse almacenados en las unidades y en la RAM 925, incluyendo un sistema operativo 935, uno o más programas 936 de aplicación, otros módulos 937 de programa, y datos 938 de programa. Un usuario puede introducir comandos e información en el ordenador 920 personal a través de un teclado 940 y de un dispositivo de apuntamiento tal como un ratón 942. Otros dispositivos (no representados) pueden incluir un micrófono, un joystick, un control de mandos, una antena parabólica, o similar. Estos y otros dispositivos de entrada se encuentran con frecuencia conectados a la unidad 921 de procesamiento a través de una interfaz 946 de puerto serie que está acoplada al bus de sistema, pero pueden estar conectados por medio de otras interfaces, tal como un puerto paralelo, un puerto para juego o un bus serie universal (USB). Un monitor 947 u otro tipo de dispositivo de visualización, se encuentra también conectado al bus 923 de sistema por medio de una interfaz, tal como un controlador de visualización o un adaptador de vídeo 948. Adicionalmente al monitor, los ordenadores personales incluyen típicamente otros dispositivos de salida periféricos (no representados), tales como altavoces e impresoras.

El ordenador 920 personal puede operar en un entorno con conexiones en red que utiliza conexiones lógicas con uno o más ordenadores remotos, tal como un ordenador 949 remoto. El ordenador 949 remoto puede ser un servidor, un enrutador, un dispositivo de búsqueda digital, u otro nodo de red común, y típicamente incluye muchos de, o todos, los elementos descritos en relación con el ordenador 920 personal, aunque en la Figura 9 solamente se ha ilustrado un dispositivo 950 de almacenamiento en memoria. Las conexiones lógicas representadas en la Figura 9 incluyen una red 951 de área local (LAN) y una red 952 de área amplia (WAN). Tales entornos de conexionado en red son lugares comunes en oficinas, redes amplias de ordenadores de empresa, intranets e Internet.

Cuando se usa en un entorno de conexionado en red LAN, el ordenador 920 personal se conecta a la red 951 local a través de una interfaz o adaptador 953 de red. Cuando se utiliza en un entorno de conexionado en red WAN, el ordenador 920 personal incluye típicamente un módem 954 u otro medio para establecer comunicaciones por la red 952 de área amplia, tal como Internet. El módem 954, que puede ser interno o externo, se conecta al bus 923 de sistema a través de la interfaz 946 de puerto serie. En un entorno de conexionado en red, los módulos de programa representados en relación con el ordenador 920 personal, o porciones de los mismos, pueden estar almacenados en el dispositivo remoto de almacenamiento en memoria. Las conexiones de red mostradas son simplemente ejemplos, y se pueden utilizar otros medios de establecimiento de un enlace de comunicaciones entre los ordenadores.

## 55 *Conclusión*

Mientras que la invención ha sido ilustrada utilizando una implementación específica como ejemplo, el ámbito de la invención no se limita a la implementación específica. Por ejemplo, es posible usar predicción espacial para ambos bloques de crominancia y luminancia utilizando técnicas similares. Además, se puede usar predicción espacial para codificar los parámetros de bloque codificados para ambos cuadros intra y pronosticados. La implementación utiliza tablas de Huffman para generar códigos de longitud variable. De hecho, se puede utilizar una diversidad de procedimientos de codificación de entropía para generar un código de longitud variable para cada parámetro de bloque codificado combinado. Por ejemplo, se pueden usar diversas formas de codificación aritmética y/o de longitud de secuencia. Cada uno de estos procedimientos de codificación asigna códigos más largos a las señales de entrada que se generan menos frecuentemente, mientras que asigna códigos más cortos a las señales de entrada más frecuentes. Según se ha indicado anteriormente, se pueden aplicar procedimientos de codificación para mejorar la eficacia de los encabezamientos de macrobloque en los procedimientos de codificación basados en cuadro y basados en objeto.

## REIVINDICACIONES

5 1. Un procedimiento de reconstrucción de una o más imágenes de vídeo en una secuencia de vídeo durante la descodificación de vídeo en un sistema de ordenador, estando el procedimiento **caracterizado** por:

para cada macrobloque de una pluralidad de macrobloques,

10 recibir un código de longitud variable en una corriente bits, en el que un patrón de bloque codificado incluye una primera información patrón de bloque codificada para una pluralidad de bloques de luminancia del macrobloque, en el que el patrón de bloque codificado incluye una segunda información patrón de bloque codificada para una pluralidad de bloques de crominancia del macrobloque, en el que el código de longitud variable recibido indica una entrada de tabla de código en una tabla de código, especificando la entrada de tabla de código un parámetro combinado que representa la primera información patrón de bloque codificada y la segunda información patrón de bloque codificada del patrón de bloque codificado para el macrobloque, en el que el patrón de bloque codificado indica cuál de la pluralidad de bloques de luminancia y de la pluralidad de bloques de crominancia tiene datos (86) correspondientes de coeficiente de transformación en la corriente de bits, y en el que el macrobloque tiene un tipo de macrobloque;

20 recuperar el patrón de bloque codificado para el macrobloque en base, al menos en parte, al código de longitud variable recibido, en el que la recuperación incluye buscar la entrada de tabla de código en la tabla de código, y en el que el tipo de macrobloque es intra para el menos uno de la pluralidad de macrobloques para el que se han recuperado patrones de bloque codificados, y

25 utilizar el patrón de bloque codificado para el macrobloque durante la reconstrucción de la una o más imágenes de vídeo

2. El procedimiento de la reivindicación 1, en el que cada macrobloque de la pluralidad de macrobloques consiste en cuatro bloques de luminancia y dos bloques de crominancia.

30 3. El procedimiento de la reivindicación 1, en el que cada macrobloque de la pluralidad de macrobloques incluye un conjunto de píxeles de 16x16, y en el que cada bloque de la pluralidad de bloques de luminancia y de la pluralidad de bloques de crominancia incluye un conjunto de píxeles de 8x8.

35 4. El procedimiento de la reivindicación 1, en el que la recuperación del patrón de bloque codificado está basado además, al menos en parte, en los datos descodificados a partir de dicho uno o más de los otros macrobloques.

5. El procedimiento de la reivindicación 4, en el que los datos descodificados a partir de dicho uno o más de los otros macrobloques comprende información patrón descodificada de bloque codificado.

40 6. El procedimiento de la reivindicación 1, en el que el código de longitud variable recibido es diferente dependiendo de si el tipo de macrobloque del macrobloque es intra.

7. El procedimiento de la reivindicación 1, en el que el patrón de bloque codificado para cada macrobloque de la pluralidad de macrobloques consiste en seis bits.

45 8. El procedimiento de la reivindicación 7, en el que cuatro de los seis bits indican qué bloques de la pluralidad de bloques de luminancia tienen, respectivamente, datos de coeficiente de transformación en la corriente de bits.

50 9. Un procedimiento de codificación de una o más imágenes de vídeo de una secuencia de vídeo en un sistema de ordenador, estando el procedimiento **caracterizado** por:

recibir una o más imágenes de vídeo de una secuencia de vídeo, y

para cada macrobloque de la pluralidad de macrobloques de la una o más imágenes de vídeo,

55 determinar un código de longitud variable para un patrón de bloque codificado del macrobloque, en el que el patrón de bloque codificado incluye una primera información (100, 106 "CBPY") patrón de bloque codificada para la pluralidad de bloques de luminancia del macrobloque, en el que el patrón de bloque codificado incluye una segunda información (100, 106, "CBPC") patrón de bloque codificada para la pluralidad de bloques de crominancia del macrobloque, en el que el código de longitud variable indica una entrada de tabla de código en una tabla de código, especificando la entrada de tabla de código un parámetro combinado que representa la primera información patrón de bloque codificada y la segunda información patrón de bloque codificada del patrón de bloque codificado para el macrobloque, en el que el patrón de bloque codificado indica cuál de la pluralidad de bloques de luminancia y de la pluralidad de bloques de crominancia tienen datos (86) correspondientes de coeficiente de transformación en una corriente de bits, en el que el macrobloque tiene un tipo de macrobloque, y en el que el tipo de macrobloque es intra para al menos uno de la pluralidad de macrobloques para el que se determinan códigos de longitud variable para patrones de bloque codificados, y

## ES 2 296 415 T3

presentar a la salida el código de longitud variable en la corriente bits, con lo que un decodificador de vídeo que recibe el código de longitud variable, determina el patrón de bloque codificado para el macrobloque en base a, al menos en parte, la búsqueda de la entrada de tabla de código en la tabla de código.

5        10. El procedimiento de la reivindicación 9, en el que cada macrobloque de la pluralidad de macrobloques consiste en cuatro bloques de luminancia y dos bloques de crominancia.

11. El procedimiento de la reivindicación 9, en el que cada macrobloque de la pluralidad de macrobloques incluye un conjunto de píxeles de 16x16, y en el que cada bloque de la pluralidad de bloques de luminancia y de la pluralidad de bloques de crominancia incluye un conjunto de píxeles de 8x8.

12. El procedimiento de la reivindicación 9, en el que la determinación del código de longitud variable se basa además, al menos en parte, en los datos procedentes de uno o más de los otros macrobloques.

15       13. El procedimiento de la reivindicación 12, en el que los datos de dicho uno o más de los otros macrobloques comprende información patrón de bloque codificada.

14. El procedimiento de la reivindicación 9, en el que el código de longitud variable es diferente dependiendo de si el tipo del macrobloque es intra.

20       15. El procedimiento de la reivindicación 9, en el que el patrón de bloque codificado para cada macrobloque de la pluralidad de macrobloques consiste en seis bits.

16. El procedimiento de la reivindicación 15, en el que cuatro de los seis bits indican qué bloques de la pluralidad de bloques de luminancia tienen respectivamente datos de coeficiente de transformación en la corriente de bits.

25       17. Un medio susceptible de lectura con ordenador que almacena instrucciones ejecutables con ordenador para provocar que el sistema de ordenador lleve a cabo el procedimiento de cualquiera de las reivindicaciones 1 a 16.

30

35

40

45

50

55

60

65

FIG. 1 (Técnica Anterior)



FIG. 4

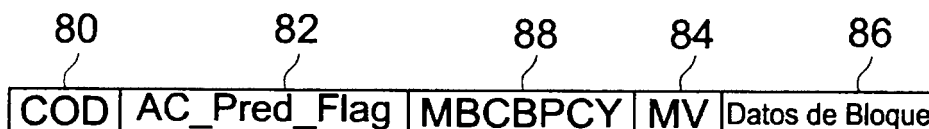


FIG. 5

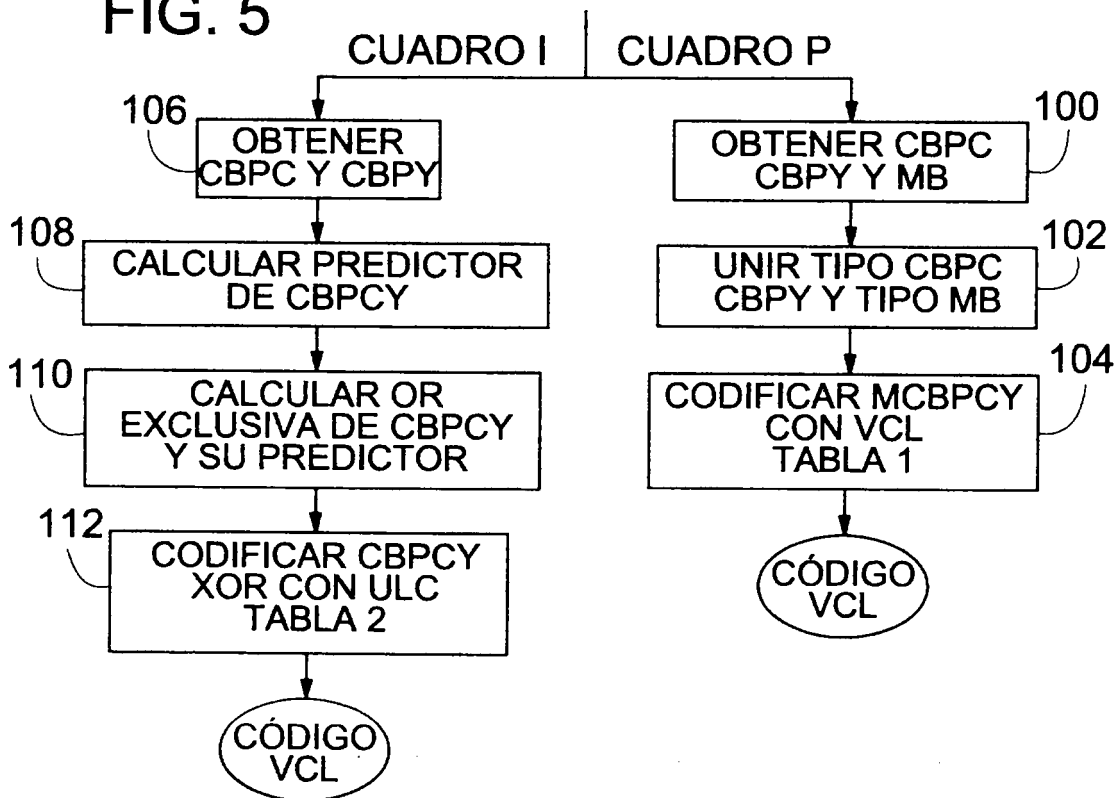




FIG. 2

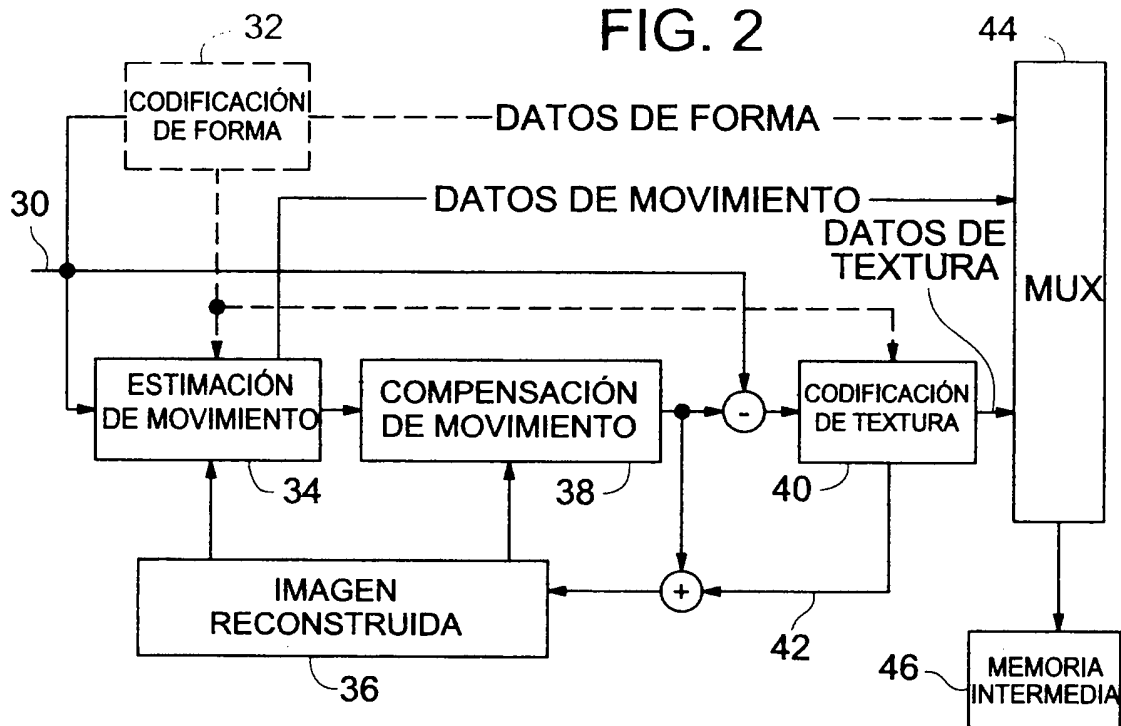


FIG. 3

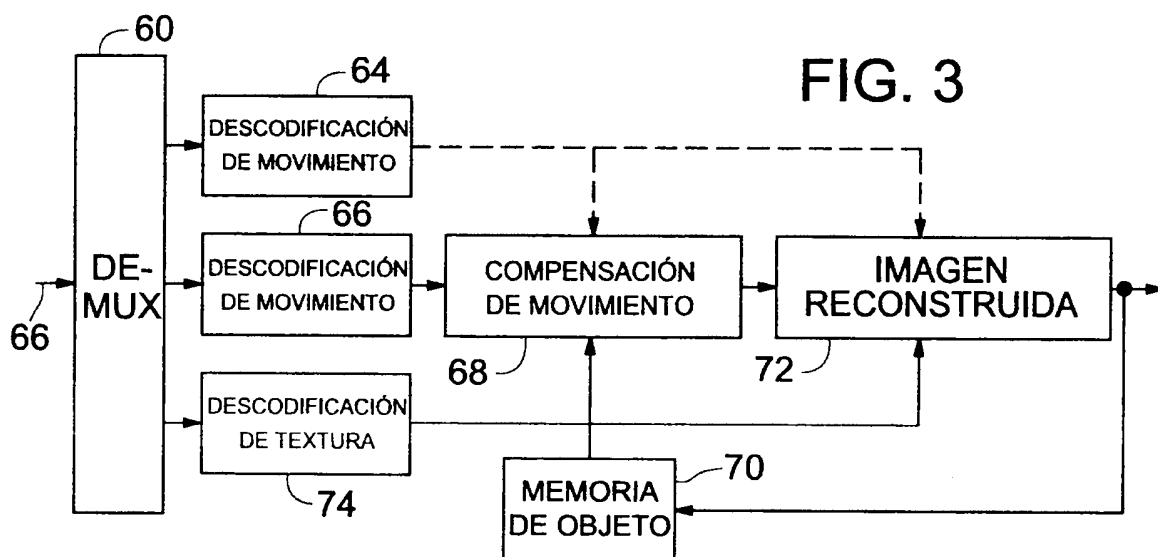


FIG. 6

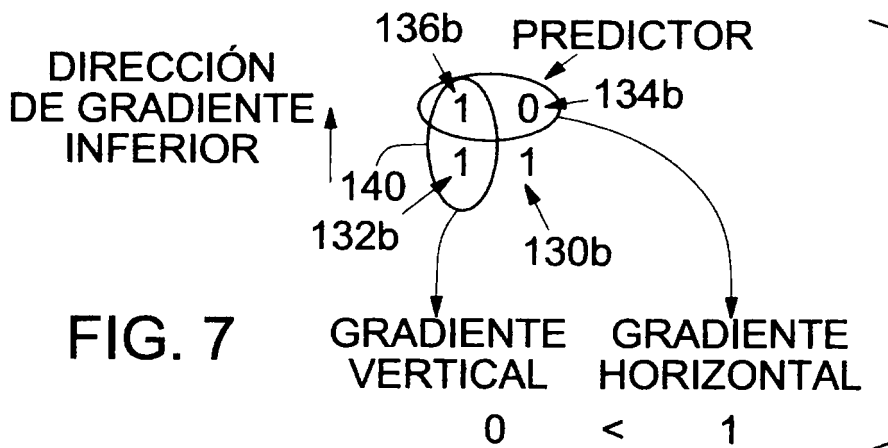
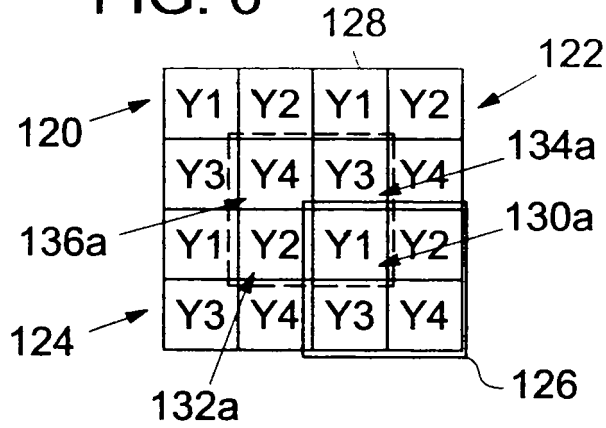


FIG. 7

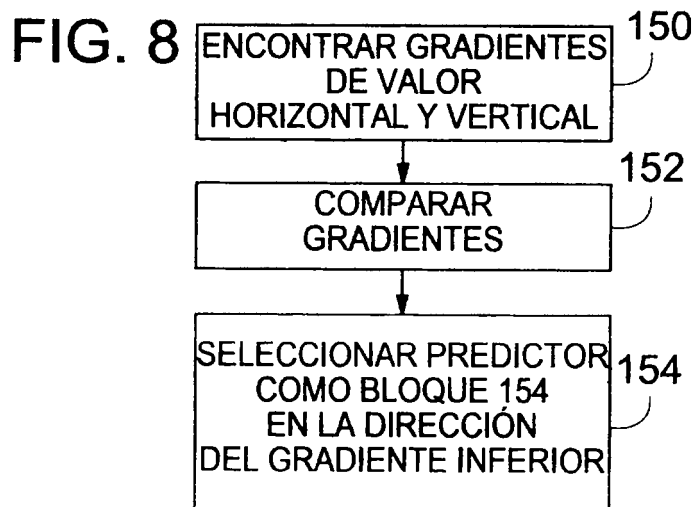


FIG. 9

