(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 355 850**

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

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Número de solicitud europea: **07724348 .3**
(96) Fecha de presentación : **18.04.2007**
(97) Número de publicación de la solicitud: **2123052**
(97) Fecha de publicación de la solicitud: **25.11.2009**

(54) Título: **Flujo de datos de vídeo con calidad ajustable a escala.**

(30) Prioridad: **18.01.2007 US 885534 P**

(45) Fecha de publicación de la mención BOPI:
31.03.2011

(45) Fecha de la publicación del folleto de la patente:
31.03.2011

(73) Titular/es: **Fraunhofer-Gesellschaft zur Förderung der Angewandten Forschung e.V.**
Hansastrasse 27C
80686 München, DE

(72) Inventor/es: **Wiegand, Thomas;**
Kirchhoffer, Heiner y
Schwarz, Heiko

(74) Agente: **Arizti Acha, Mónica**

DESCRIPCIÓN

Flujo de datos de vídeo con calidad ajustable a escala

Antecedentes

La presente invención se refiere a flujos de datos de vídeo con calidad ajustable a escala, a su generación y decodificación tal como la generación y decodificación de flujos de datos de vídeo obtenidos mediante el uso de transformación por bloques.

El actual Equipo Conjunto de Vídeo "JVT" del Grupo de Expertos en Codificación de Vídeo de la ITU-T en el Grupo de Expertos en Imágenes en Movimiento (MPEG) de la ISO/IEC está en la actualidad especificando una extensión ajustable a escala de la norma de codificación de vídeo H.264/MPEG4-AVC. La característica clave de la codificación de vídeo ajustable a escala (SVC) en comparación con la codificación convencional de una única capa es que se proporcionan varias representaciones de una fuente de vídeo con diferentes resoluciones, tasas de transmisión de tramas y/o tasas de transmisión de bits dentro de un único flujo de bits. A partir de un flujo de bits de SVC global puede extraerse mediante sencillas manipulaciones del flujo, tales como la eliminación de paquetes, una representación de vídeo con una resolución espaciotemporal y una tasa de transmisión de bits específicas. Como característica importante del diseño SVC, la mayoría de los componentes de H.264/MPG4-AVC se usan tal como se especifican en la norma. Esto incluye la predicción con compensación de movimiento y la intrapredicción, la codificación de transformada y de entropía, el desbloqueo así como la paquetización de unidades NAL (NAL = capa de abstracción de red). La capa base de un flujo de bits de SVC se codifica generalmente de conformidad con la norma H.264-MPEG4-AVC, y por tanto cualquier decodificador que cumpla con la norma H.264-MPEG4-AVC puede decodificar la representación de la capa base cuando se le proporciona un flujo de bits de SVC. Simplemente se añaden nuevas herramientas para soportar la ajustabilidad a escala espacial y de SNR.

Para la ajustabilidad a escala de SNR, se distingue entre ajustabilidad a escala de grano grueso/grano medio (CGS/MGS) y ajustabilidad a escala de grano fino (FGS) en el actual borrador de trabajo. La codificación ajustable a escala de SNR de grano grueso o grano medio se consigue usando conceptos similares a los usados para la ajustabilidad a escala espacial. Las imágenes de diferentes capas de SNR se codifican de manera independiente con parámetros de movimiento específicos de la capa. Sin embargo, con el fin de mejorar la eficacia de la codificación de las capas mejoradas en comparación con la difusión simultánea, se han introducido mecanismos de predicción entre capas adicionales. Estos mecanismos de predicción se han hecho intercambiables de modo que un codificador pueda elegir libremente qué información de capa base debe aprovechar para una codificación de capa de mejora eficaz. Puesto que los conceptos de predicción entre capas incorporados incluyen técnicas para la predicción residual y de parámetros de movimiento, las estructuras de predicción temporal de las capas de SNR deben alinearse temporalmente para un uso eficaz de la predicción entre capas. Ha de indicarse que todas las unidades NAL para un instante de tiempo forman una unidad de rebosamiento y por tanto tienen que sucederse unas tras otras dentro de un flujo de bits de SVC. En el diseño SVC se incluyen las siguientes tres técnicas de predicción entre capas.

La primera se denomina predicción de movimiento entre capas. Con el fin de emplear datos de movimiento de la capa base para la codificación de capas de mejora, se ha introducido un modo de macrobloque adicional en las capas de mejora de SNR. La división del macrobloque se obtiene copiando la división del macrobloque coubicado en la capa base. Los índices de la imagen de referencia así como los vectores de movimiento asociados se copian de los bloques de capa base coubicados. Adicionalmente, un vector de movimiento de la capa base puede usarse como predictor de vector de movimiento para los modos de macrobloque convencionales.

La segunda técnica de reducción de redundancia entre las diversas capas de calidad se denomina predicción residual entre capas. El uso de la predicción residual entre capas se señaliza mediante un indicador (flag) (residual_prediction_flag) que se transmite para todos los macrobloques intercodificados. Cuando este indicador es verdadero, la señal de capa base del bloque coubicado se usa como predicción para la señal residual del macrobloque actual, de modo que sólo se codifica la correspondiente señal de diferencia.

Finalmente, la intrapredicción entre capas se usa con el fin de aprovechar la redundancia entre las capas. En este modo de intra-macrobloque, la señal de predicción se construye mediante la señal de reconstrucción coubicada de la capa base. Para la intrapredicción entre capas generalmente se requiere que las capas base estén completamente decodificadas incluyendo las operaciones computacionalmente complejas de predicción con compensación de movimiento y desbloqueo. Sin embargo, se ha mostrado que este problema puede evitarse cuando la intrapredicción entre capas se limita a aquellas partes de la imagen de capa inferior que están intracodificadas. Con esta limitación, cada capa objetivo soportada puede decodificarse con un único bucle con compensación de movimiento. Este modo de decodificación

de un solo bucle es obligatorio en la extensión H.264-MPEG4-AVC ajustable a escala.

Puesto que la intrapredicción entre capas sólo puede aplicarse cuando el macrobloque coubicado está intracodificado y la predicción de movimiento entre capas con inferencia del tipo de macrobloque sólo puede aplicarse cuando el macrobloque de capa base está intercodificado, ambos modos se señalizan mediante un único elemento de sintaxis base_mode_flag a nivel de macrobloque. Cuando este indicador es igual a 1, se elige la intrapredicción entre capas cuando el macrobloque de capa base está intracodificado. En caso contrario, el modo de macrobloque así como los índices de referencia y los vectores de movimiento se copian desde el macrobloque de capa base.

Con el fin de soportar una granularidad más fina que la codificación CGS/MGS, se han introducido denominados segmentos de refinamiento progresivo que posibilitan una codificación ajustable a escala de SNR con granularidad más fina (FGS). Cada segmento de refinamiento progresivo representa un refinamiento de la señal residual que corresponde a una bisección del tamaño de paso de cuantificación (incremento de QP de 6). Estas señales están representadas de manera que sólo tienen que realizarse una única transformada inversa para cada bloque de transformada en el lado del decodificador. La ordenación de los niveles de coeficientes de transformada en segmentos de refinamientos progresivos permite truncar las correspondientes unidades NAL en cualquier punto arbitrario en alineación de byte, de modo que la calidad de la capa base de SNR puede refinarse con granularidad fina. Además de un refinamiento de la señal residual, también es posible transmitir un refinamiento de parámetros de movimiento como parte de los segmentos de refinamiento progresivo.

Un inconveniente de la codificación de FGS en el actual borrador de SVC es que aumenta de manera significativa la complejidad del decodificador en comparación con la codificación CGS/MGS. Por un lado, los coeficientes de transformada en un segmento de refinamiento progresivo se codifican usando varios barridos por los bloques de transformada, y en cada barrido sólo se transmiten unos pocos niveles de coeficientes de transformada. Para el decodificador, esto aumento la complejidad ya que se requiere un ancho de banda con más memoria, porque todos los niveles de coeficientes de transformada de diferentes barridos tienen que recopilarse antes de que pueda llevarse a cabo la transformada inversa. Por otro lado, el proceso de análisis sintáctico para segmentos de refinamiento progresivo depende de los elementos de sintaxis de los correspondientes segmentos de capa base. El orden de los elementos de sintaxis así como las tablas de palabras de código para la codificación VLC o la selección del modelo de probabilidad para codificación aritmética dependen de los elementos de sintaxis en la capa base. Esto aumenta adicionalmente el ancho de banda con memoria requerido para la decodificación, ya que hay que acceder a los elementos de sintaxis de la capa base durante el análisis sintáctico de la capa de mejora.

Además, la propiedad especial de los segmentos de refinamiento progresivo de que pueden truncarse es difícil de usar en las redes de conmutación de paquetes actuales. Habitualmente, un dispositivo de red con capacidad para medios o bien entregará o bien eliminará un paquete de un flujo de bits ajustable a escala. El único error que será visible en la capa de aplicación es una pérdida de paquetes.

Por tanto, no sólo con respecto a la anterior norma H.264-MPEG4-AVC sino también a otras técnicas de compresión de vídeo, sería deseable tener un esquema de codificación que esté mejor adaptado a las necesidades actuales que muestran problemas de pérdida de paquetes en lugar de truncamiento por bytes.

La técnica anterior relevante se da a conocer en la solicitud de patente internacional publicada con el número WO 2004/030368 y en el documento “An Optimal Single Pass SNR Scalable Video Coder” de Kondi L.P. *et al*.

Sumario

La presente invención se define en las reivindicaciones 1, 2, 12-15.

Breve descripción de los dibujos

Tomando como base las figuras, a continuación se describen con más detalle algunas realizaciones de la presente invención. En particular,

la figura 1 muestra un diagrama de bloques de un codificador que genera un flujo de datos de vídeo con calidad ajustable a escala según una realización;

la figura 2 muestra un diagrama de bloques de un codificador híbrido de capa superior de la figura 1 según una realización;

la figura 3 muestra un diagrama de bloques de un codificador híbrido de capa base de la figura 1 según una realización;

la figura 4 muestra un diagrama de bloques de una unidad de codificación en capas de la capa de calidad más alta de la figura 1 según una realización;

la figura 5 muestra un diagrama esquemático que ilustra la estructura de una imagen así como su transformación por bloques según una realización;

las figuras 6a-6g muestran diagramas esquemáticos de una parte barrida de un bloque de transformada y su división en subcapas según varias realizaciones;

la figura 7 muestra un diagrama esquemático que ilustra la construcción de subflujos de datos según una realización;

la figura 8 muestra un pseudocódigo que ilustra la codificación de los niveles de coeficientes de transformada pertenecientes a un subflujo de datos específico según una realización;

la figura 9 muestra un pseudocódigo que ilustra otro ejemplo para codificar los niveles de coeficientes de transformada pertenecientes a un subflujo de datos específico;

la figura 10, un diagrama de bloques de un decodificador según otra realización; y

la figura 11, un diagrama de bloques de una realización para la unidad de decodificación de la figura 10.

Descripción detallada de los dibujos

La figura 1 muestra un codificador para generar un flujo de bits con calidad ajustable a escala. A modo de ejemplo, el codificador 10 de la figura 1 está previsto para generar un flujo de bits ajustable a escala que soporta dos capas espaciales diferentes y N+1 capas de SNR. Para ello, el codificador 10 está estructurado en una parte 12 de capa base y una parte 14 de capa de mejora espacial. Una unidad 16 de reducción de calidad del codificador 10 recibe el vídeo 18 original o de calidad más alta representando una secuencia de imágenes 20 y reduce su calidad, en el sentido de resolución espacial en el ejemplo de la figura 1, para obtener una versión 22 de calidad más baja del vídeo 18 original que consiste en una secuencia de imágenes 24, introduciéndose la versión 22 de calidad más baja en la parte 12 de capa base.

La unidad 16 de reducción de calidad realiza, por ejemplo, un submuestreo de las imágenes por un factor de submuestreo de 2, por ejemplo. Sin embargo, ha de entenderse que aunque la figura 1 muestra un ejemplo que soporta dos capas 12, 14 espaciales, la realización de la figura 1 puede aplicarse fácilmente a aplicaciones en las que la reducción de calidad realizada entre el vídeo 18 original y el vídeo 22 de calidad más baja no comprende un submuestreo, sino, por ejemplo, una reducción en la profundidad de bits de la representación de los valores de píxel, o bien la unidad de reducción de calidad simplemente copia la señal de entrada a la señal de salida.

Aunque la parte 12 de capa base recibe el vídeo 22 de calidad más baja, el vídeo 18 original se introduce en la parte 14 de calidad más alta, realizando ambas partes 12, 14 una codificación híbrida sobre el vídeo respectivamente introducido. La parte 12 de capa base recibe el vídeo 22 de calidad más baja y genera un flujo 26 de bits de capa base. Por otro lado, la parte 14 de capa de calidad más alta recibe en su entrada el vídeo 18 original y genera, además de un flujo 28 de bits de capa de mejora espacial, N flujos 30 de bits de capa de refinamiento de SNR. La generación y la interrelación entre flujos 28 y 26 de bits se describirán con más detalle a continuación. Únicamente como medida de precaución, se indica que la parte 12 de capa base también puede acompañar al flujo 26 de bits de capa base mediante varias capas 32 de refinamiento de SNR. Sin embargo, para facilitar la ilustración de los principios de la presente realización, se supone que la ajustabilidad a escala de SNR se limita a la parte 14 de capa de mejora. Sin embargo, la explicación siguiente dejará claro que la funcionalidad descrita a continuación con respecto a la parte 14 de capa de calidad más alta en relación con las capas de refinamiento de SNR puede transferirse fácilmente a la parte 12 de capa base. Esto se indica en la figura 1 mediante una línea 32 de puntos.

Todos los flujos 26 a 32 de bits se introducen en un multiplexor 34 que genera un flujo 36 de bits ajustable a escala a partir de los flujos de datos en su entrada, eventualmente dispuestos en paquetes, tal como se describirá con más detalle a continuación.

Internamente, la parte 12 de capa base comprende un codificador 38 híbrido y una unidad 40 de codificación en capas conectados en serie, en el orden mencionado, entre la entrada a la que se aplica el vídeo 24 de baja calidad, por un lado, y el multiplexor 34, por otro lado. De manera similar, la parte 14 de

capa de calidad más alta comprende un codificador 42 híbrido y una unidad 44 de codificación en capas conectados entre la entrada a la que se aplica el vídeo 18 original y el multiplexor 44. Cada codificador 42 y 38 híbrido, respectivamente, codifica su señal de entrada de vídeo mediante codificación híbrida, es decir se usa predicción con compensación de movimiento junto con transformación por bloques del residuo de predicción. Por tanto, cada codificador 38 y 42 híbrido, respectivamente, emite datos 46 y 48 de información de movimiento, respectivamente, así como datos 50 y 52 residuales, respectivamente, a la entrada de la posterior unidad 40 y 44 de codificación en capas, respectivamente.

Naturalmente, existe redundancia entre los datos de movimiento, 46 por un lado y 48 por otro lado, así como los datos 50 y 52 residuales. Esta redundancia entre capas la aprovecha el codificador 42 híbrido. En particular, por macrobloques, el codificador 42 híbrido puede elegir entre varias opciones de predicción entre capas. Por ejemplo, el codificador 42 híbrido puede decidir usar o adoptar los datos 46 de movimiento de capa base como los datos 48 de movimiento para la capa de calidad más alta. Alternativamente, el codificador 42 híbrido puede decidir usar los datos 46 de movimiento de capa base como predictor para los datos 48 de movimiento. Como alternativa adicional, el codificador 42 híbrido puede codificar los datos 48 de movimiento completamente desde cero, es decir independientemente de los datos de movimiento de capa base.

De manera similar, el codificador 42 híbrido puede codificar los datos 42 residuales para la capa de calidad más alta de manera predictiva como el residuo de predicción en relación con los datos 50 residuales de capa base como predictor.

Sin embargo, el codificador 42 híbrido también puede usar una reconstrucción del contenido de la imagen de la capa base como predictor para el contenido de la imagen de los datos de vídeo original de modo que, en este caso, los datos de movimiento y/o los datos residuales, 48 y 52 respectivamente, meramente codifican el residuo en relación con los datos de capa base reconstruidos. Tal como se describirá con respecto a la figura 2, la información de imagen de capa base reconstruida puede recibirla el codificador 38 híbrido de capa base o una unidad 54 de reconstrucción dedicada acoplada entre la unidad 40 de codificación de capa base y un codificador 42 híbrido de capa de calidad más alta.

A continuación se describirá con más detalle una estructura interna y la funcionalidad de los codificadores 38 y 42 híbridos así como la unidad 44 de codificación en capas. Con respecto a la unidad 40 de codificación en capas, a continuación se supone que ésta meramente genera el flujo 26 de datos de capa base. Sin embargo, tal como se indicó anteriormente, una alternativa de una realización según la cual la unidad 40 de codificación en capas también genera flujos 32 de datos de capa de refinamiento de SNR puede derivarse fácilmente a partir de la siguiente descripción con respecto a la unidad 44 de codificación en capas.

En primer lugar se describe la estructura interna y la funcionalidad del codificador 38 híbrido de capa base. Tal como se muestra en la figura 3, el codificador 38 híbrido de capa base comprende una entrada 56 para recibir las señales 24 de vídeo de calidad más baja, una salida 58 para los datos 46 de movimiento, una salida 60 para los datos 50 residuales, una salida 62 para acoplar los datos 58 de movimiento al codificador 42 híbrido, una salida 64 para acoplar los datos de imagen de capa base reconstruidos al codificador 42 híbrido, y un salida 66 para acoplar los datos 50 residuales al codificador 42 híbrido.

Internamente, el codificador 38 híbrido comprende una unidad 68 de transformación, una unidad 70 de transformación hacia atrás, un restador 72, un sumador 74, y una unidad 76 de predicción de movimiento. El restador 72 y la unidad 68 de transformación están acoplados, en el orden mencionado, entre la entrada 56 y la salida 60. El restador 72 resta de la señal de vídeo de entrada el contenido de vídeo con predicción de movimiento recibido desde la unidad 76 de predicción de movimiento y reenvía la señal de diferencia a la unidad 68 de transformación. La unidad 68 de transformación realiza una transformación por bloques sobre la señal de diferencia/residual junto con, opcionalmente, una cuantificación de los coeficientes de transformada. El resultado de la transformación se emite por la unidad 68 de transformación a la salida 60 así como a una entrada de la unidad 70 de transformación hacia atrás. La unidad 70 de transformación hacia atrás realiza una transformación inversa sobre los bloques de transformada de coeficientes de transformación con, eventualmente, una descuantificación previa. El resultado es una señal residual reconstruida que se combina mediante suma, por el sumador 74, con el contenido de vídeo con predicción de movimiento emitido por la unidad 76 de predicción de movimiento. El resultado de la suma realizada por el sumador 74 es un vídeo reconstruido en calidad base. La salida del sumador 74 se acopla a una entrada de la unidad 76 de predicción de movimiento así como a la salida 64. La unidad 76 de predicción de movimiento realiza una predicción con compensación de movimiento basándose en las imágenes reconstruidas con el fin de predecir otras imágenes del vídeo introducido en la entrada 56. La unidad 76 de predicción de movimiento produce, mientras realiza predicción de movimiento, datos de movimiento que incluyen, por ejemplo, vectores de movimiento e índices de referencia de imagen de movimiento y emite estos datos de movimiento de modo a la salida 62

así como a la salida 58. La salida de la unidad 68 de transformación también está acoplada a la salida 66 con el fin de reenviar los datos residuales de transformada al codificador 42 híbrido de la capa de calidad más alta. Tal como ya se mencionó anteriormente, la funcionalidad de ambos codificadores 38 y 42 híbridos de la figura 1 es similar una respecto a otra. Sin embargo, el codificador 42 híbrido de la capa de calidad más alta también usa predicción entre capas. Por tanto, la estructura del codificador 42 híbrido mostrada en la figura 2 es similar a la estructura del codificador 38 híbrido mostrada en la figura 3. En particular, el codificador 42 híbrido comprende una entrada 86 para la señal 18 de vídeo original, una salida 88 para los datos 48 de movimiento, una salida 90 para los datos 52 residuales, y tres entradas 92, 94 y 96 para acoplarse a las respectivas salidas 62, 64 y 66 del codificador 38 híbrido de capa base. Internamente, el codificador 42 híbrido comprende dos conmutadores o selectores 98 y 100 para conectar una de dos trayectorias 102 y 104 entre la entrada 86 y la salida 90. En particular, la trayectoria 104 comprende un restador 106, una unidad 108 de transformación y un codificador 110 predictivo residual conectados, en el orden mencionado, entre la entrada 86 y la salida 90 a través de conmutadores 98 y 100. El restador 106 y la unidad 108 de transformación forman, junto con una unidad 112 de transformación hacia atrás, un sumador 114 y una unidad 116 de predicción de movimiento, un bucle de predicción tal como el formado por los elementos 68 a 76 en el codificador 38 híbrido de la figura 3. Por consiguiente, en la salida de la unidad 108 de transformación se obtiene como resultado una versión transformada del residuo con predicción de movimiento que se introduce en el codificador 110 predictivo residual. El codificador 110 predictivo residual también está conectado a la entrada 96 con el fin de recibir los datos residuales de capa base. Mediante el uso de estos datos residuales de capa base como predictor, el codificador 110 predictivo residual codifica una parte de los datos residuales emitidos por la unidad 108 de transformación como residuo de predicción en relación con los datos residuales en la entrada 96. Por ejemplo, el codificador 110 predictivo residual sobremuestrea los datos residuales de capa base y resta los datos residuales sobremuestreados de los datos residuales emitidos por la unidad 108 de transformación. Evidentemente, el codificador 110 predictor residual puede realizar la predicción sólo para una parte de los datos residuales emitidos por la unidad 108 de transformación. Otras trayectorias pasan sin cambios por el codificador 110 predictivo residual. La granularidad de estas partes puede ser de macrobloques. En otras palabras, la decisión sobre si los datos residuales en la entrada 96 pueden usarse como predictor o no puede tomarse por macrobloques y el resultado de la decisión puede indicarse mediante un respectivo elemento de sintaxis residual_prediction_flag.

De manera similar, el codificador 42 híbrido comprende un codificador 118 predictivo de parámetros de movimiento con el fin de recibir los datos de movimiento en la entrada 92 desde la capa base así como la información de movimiento obtenida desde la unidad 116 de predicción de movimiento y conmuta, por macrobloques, entre pasar los datos de movimiento desde la unidad 116 de predicción de movimiento sin cambiar a la salida 88, o codificar de manera predictiva los datos de movimiento usando la información de movimiento de la capa base en la entrada 92 como predictor. Por ejemplo, el codificador 118 predictivo de parámetros de movimiento puede codificar vectores de movimiento de la unidad 116 de predicción de movimiento como vectores de desfase en relación con los vectores de movimiento contenidos en los datos de movimiento de capa base en la entrada 92. Alternativamente, el codificador 118 predictivo de parámetros de movimiento pasa la información de capa base desde la entrada 92 a la unidad 116 de predicción de movimiento para que se usen para la predicción de movimiento en la capa de calidad más alta. En este caso, no tienen que transmitirse datos de movimiento para la respectiva parte de la señal de vídeo de capa de calidad más alta. Como alternativa adicional, el codificador 118 predictivo de parámetros de movimiento ignora la existencia de los datos de movimiento en la entrada 92 y codifica los datos de movimiento desde la unidad 116 de predicción de movimiento directamente a la salida 88. La decisión entre estas posibilidades se codifica dando lugar al flujo de bits con ajustabilidad a escala de la calidad.

Finalmente, el codificador 120 predictivo se proporciona en la trayectoria 102 y se acopla con la entrada 94. El codificador 120 predictivo predice partes de la capa de calidad más alta basándose en respectivas partes de la señal de vídeo de capa base reconstruida de modo que en la salida del codificador 120 predictivo se reenvía meramente el respectivo residuo o diferencia. El codificador 120 predictivo también funciona por macrobloques en cooperación con los conmutadores 98 y 100.

Tal como puede verse a partir de la figura 4, la unidad 44 de codificación en capas de la capa de calidad más alta comprende una entrada 122 para recibir los coeficientes de transformada de datos residuales desde la salida 90 y una entrada 124 para recibir los datos de movimiento desde la salida 88. Una unidad 126 de distribución recibe los coeficientes de transformación y los distribuye a varias capas de mejora. Los coeficientes de transformación así distribuidos se emiten a una unidad 128 de formación. Junto con los coeficientes de transformación distribuidos, la unidad 128 de formación recibe los datos de movimiento desde la entrada 124. La unidad 128 de formación combina ambos datos y forma, basándose en estas entradas de datos, el flujo 28 de datos de capa de mejora de orden cero así como los flujos 30 de datos de capa de refinamiento.

Con el fin de posibilitar una descripción más detallada de la funcionalidad de la unidad 126 de

distribución y la unidad 128 de formación, a continuación se describirá con más detalle con respecto a la figura 5 la operación por bloques subyacente a la transformación realizada por la unidad 108 de transformación y su interrelación con la distribución realizada por la unidad 126 de distribución. La figura 5 representa una imagen 140. La imagen 140 es, por ejemplo, parte de los datos 18 de vídeo de alta calidad (figura 1). En la imagen 140, los píxeles están dispuestos, por ejemplo, en líneas y columnas. La imagen 140 está dividida, por ejemplo, en macrobloques 142, que también pueden disponerse de manera regular en líneas y columnas. Cada macrobloque 142 puede abarcar espacialmente, por ejemplo, un área de imagen rectangular con el fin comprender, por ejemplo, 16x16 muestras de, por ejemplo, la componente luma de la imagen. Para ser aún más precisos, los macrobloques 142 pueden organizarse en pares de macrobloques. En particular, el par de macrobloques 142 adyacentes verticalmente puede formar un par de macrobloques de este tipo y puede asumir, espacialmente, una región 144 de par de macrobloques de la imagen 140. Por pares de macrobloques, el codificador 42 híbrido (figura 1) puede tratar los macrobloques 142 dentro de la respectiva región 144 en modo campo o modo trama. En el caso del modo campo, el vídeo 18 se supone que contiene dos campos entrelazados, un campo superior y un campo inferior, en los que el campo superior contiene las filas de píxeles con numeración par, y el campo inferior contiene las filas con numeración impar empezando por la segunda línea de la imagen 140. En este caso, el macrobloque superior de la región 144, se refiere a los valores de píxel de las líneas de campo superior dentro de la región 144, mientras que el macrobloque inferior de la región 144 se refiere al contenido de las líneas restantes. Por tanto, en este caso, ambos macrobloques asumen espacialmente de manera sustancial la totalidad del área de la región 144 con una resolución verticalmente reducida. En el caso del modo trama, el macrobloque superior se define para abarcar espacialmente la mitad superior de las filas dentro de la región 144, mientras que el macrobloque inferior comprende las muestras de imagen restantes en la región 144.

Tal como ya se indicó anteriormente, la unidad 108 de transformación realiza una transformación por bloques de la señal residual emitida por el restador 106. A este respecto, la operación por bloques para la transformación dentro de la unidad 108 de transformación puede diferir con respecto al tamaño de macrobloque de los macrobloques 142. En particular, cada uno de los macrobloques 142 puede dividirse en cuatro, es decir bloques 146 de transformada de 2x2 o 16, es decir bloques 148 de transformada de 4x4. Usando el ejemplo anteriormente mencionado para el tamaño de macrobloque de 16x16 muestras de imagen, la unidad 108 de transformación transformará los macrobloques 142 de la imagen 140 por bloques en bloques de un tamaño de 4x4 muestras de píxel o 8x8 muestras de píxel. Por tanto, la unidad 108 de transformación emite, para un determinado macrobloque 142, varios bloques 146 y 148 de transformada respectivamente, concretamente 16 bloques de coeficientes de transformada de 4x4 o 4 bloques 146 de coeficientes de transformada de 8x8.

En 150 en la figura 5, se ilustra un caso de un bloque de coeficientes de transformada de 8x8 de un macrobloque codificado en trama. En particular, en 150, cada coeficiente de transformada se asigna a y se representa por un número de posición de barrido, oscilando estos números desde 0 hasta 63. Tal como se ilustra mediante los ejes 152, los respectivos coeficientes de transformación están asociados con una componente de frecuencia espacial diferente. En particular, la frecuencia asociada con un respectivo coeficiente de los coeficientes de transformada aumenta en magnitud desde una esquina superior izquierda hasta la esquina inferior derecha del bloque 150 de transformada. El orden de barrido definido por las posiciones de barrido entre los coeficientes de transformada del bloque de transformada 150, barre los coeficientes de transformada desde la esquina superior izquierda en zigzag hasta la esquina inferior derecha, estando este barrido en zigzag ilustrado mediante las flechas 154.

Únicamente para completar, se indica que el barrido entre los coeficientes de transformada puede definirse de manera diferente entre los coeficientes de transformada de un bloque de transformada de un macrobloque codificado en campo. Por ejemplo, tal como se muestra en 156 en la figura 5, en el caso de un macrobloque codificado en campo, el barrido 158 de coeficientes de transformada barre los coeficientes de transformada desde la esquina superior izquierda hasta la esquina inferior derecha en zigzag en una dirección en vaivén o zigzag con una inclinación mayor que la dirección en zigzag de 45° usada en el caso del macrobloque codificado en trama en 150. En particular, un barrido 158 de coeficientes barre los coeficientes de transformada en una dirección en columna dos veces más rápido que en la dirección en línea con el fin de tener en cuenta el hecho de que los macrobloques codificados en campo abarcan muestras de imagen que tienen un paso de columna que es dos veces el paso horizontal o de línea. Por tanto, tal como es el caso con el barrido 154 de coeficientes, el barrido 158 de coeficientes barre los coeficientes de transformada de modo que la frecuencia aumenta a medida que aumenta el número de barrido de posición.

En 150 y 158, se muestran ejemplos de barridos de coeficientes de bloques de coeficientes de transformada de 8x8. Sin embargo, tal como ya se indicó anteriormente, también pueden existir bloques de transformada de menor tamaño, es decir 4x4 coeficientes de transformada. Para estos casos se muestran respectivos barridos de posición en la figura 5 en 160 y 162, respectivamente, estando el barrido 164 en el caso de 160 dedicado a macrobloques codificados en trama, mientras que el barrido 166

ilustrado en 162 está dedicado a macrobloques codificados en campo.

Ha de enfatizarse que los ejemplos específicos mostrados en la figura 5 con respecto a los tamaños y disposiciones de los macrobloques y bloques de transformada son únicamente con carácter ilustrativo, y que variaciones diferentes son aplicables fácilmente. Antes de empezar con la descripción de las figuras siguientes, se indica que la imagen 140 puede subdividirse, por macrobloques, en varios segmentos 168. Un segmento 168 de este tipo se muestra a modo de ejemplo en la figura 5. Un segmento 168 es una secuencia de macrobloques 142. La imagen 140 puede dividirse en uno o varios segmentos 168.

Tras haber descrito la subdivisión de una imagen en regiones de pares de macrobloques, macrobloques y bloques de transformada así como en segmentos, respectivamente, la funcionalidad de la unidad 126 de distribución y la unidad 128 de formación se describe a continuación con más detalle. Tal como puede verse a partir de la figura 5, el orden de barrido definido entre los coeficientes de transformada permite ordenar los coeficientes de transformada ordenados bidimensionalmente en una secuencia lineal de coeficientes de transformada con contenido de frecuencia creciente de manera monótona al que hacen referencia. La unidad 126 de distribución opera para distribuir el coeficiente de transformada de varios macrobloques 142 en capas de diferentes calidad, es decir cualquiera de las capas de orden cero asociadas con un flujo 28 de datos y las capas 30 de refinamiento. En particular, la capa 126 de distribución intenta distribuir los coeficientes de transformada en los flujos 28 y 30 de datos de tal manera que a medida que aumenta el número de capas de contribución desde la capa cero o capa 28 hasta la capa 30 de refinamiento de calidad más alta, aumenta la calidad de SNR del vídeo reconstruible a partir de los respectivos flujos de datos. En general, esto llevará a una distribución en la que los coeficientes de transformada de frecuencia más baja correspondientes a las posiciones de barrido más bajas se distribuyen a las capas de calidad más baja mientras que los coeficientes de transformada de frecuencia más alta se distribuyen a las capas de calidad más alta. Por otro lado, la unidad 126 de distribución tenderá a distribuir coeficientes de transformada con mayores valores de coeficiente de transformada a capas de calidad más baja y coeficientes de transformada con menores valores de coeficiente de transformada o energía a capas de calidad más alta. La distribución formada por la unidad 126 de distribución puede realizarse de tal manera que cada uno de los coeficientes de transformada se distribuye a una única capa. Sin embargo, también es posible que la distribución realizada por la unidad 126 de distribución se realice de tal manera que la cantidad de un coeficiente de transformada también pueda distribuirse a diferentes capas de calidad en partes de modo que las partes distribuidas sumen el valor de coeficiente de transformada. A continuación se describirán detalles de las diferentes posibilidades para la distribución realizada por la unidad 126 de distribución con respecto a la figura 6a-g. La unidad 128 de formación usa la distribución que resulta de la unidad 126 de distribución con el fin de formar respectivos subflujos 28 y 30 de datos. Tal como ya se indicó anteriormente, el subflujo 28 de datos forma el subflujo de datos de refinamiento de capa de calidad más baja y contiene, por ejemplo, los datos de movimiento introducidos en la entrada 124. A este subflujo 128 de datos de orden cero también se le puede proporcionar una primera parte distribuida de los valores de coeficiente de transformada. Por tanto, el subflujo 28 de datos permite un refinamiento del flujo 26 de datos de capa de calidad base a una calidad más alta, en el caso de la figura 1 a una calidad espacial más alta, aunque puede obtenerse una mejora de calidad de SNR adicional acompañando el subflujo 28 de datos con cualquiera de los demás subflujos 30 de datos de refinamiento de calidad más alta. El número de estos subflujos 30 de datos de calidad de refinamiento es N, donde N puede ser uno o más de uno. Los coeficientes de transformada se “distribuyen” por tanto, por ejemplo en orden de importancia creciente respecto a la calidad de SNR, a estos subflujos 28 y 30 de datos.

La figura 6a muestra un ejemplo de una distribución de los primeros 26 valores de coeficiente de transformada de un bloque de transformada de 8x8. En particular, la figura 6a muestra una tabla en la que la primera línea de la tabla enumera las respectivas posiciones de barrido según el orden de barrido 154 y 158, respectivamente (figura 5). Puede observarse que las posiciones de barrido mostradas se extienden, a modo de ejemplo, desde 0 hasta 25. Las siguientes tres líneas muestran los correspondientes valores de contribución incorporados en los respectivos subflujos 28 y 30 de datos, respectivamente, para los valores de coeficiente de transformada individuales. En particular, la segunda línea corresponde, por ejemplo, al subflujo 28 de datos de orden cero mientras que la penúltima línea pertenece a la siguiente capa 30 de refinamiento superior y la última línea se refiere al flujo de datos de capa de calidad de refinamiento par siguiente. Según el ejemplo de la figura 6a, se codifica un “122” en los subflujos 128 de datos para la componente DC, es decir el valor de coeficiente de transformada perteneciente a la posición de barrido 0. Los valores de contribución para este coeficiente de transformada que tiene la posición de barrido 0 dentro de los siguientes dos subflujos 30 de datos, se ajustan a cero tal como se indica mediante el sombreado de las respectivas entradas de tabla. De este modo, según el ejemplo de la figura 6a, el subflujo 28 de datos de capa de mejora de orden cero comprende un valor de distribución para cada uno de los valores de coeficiente de transformada. Sin embargo, en el bloque de transformada de la figura 6a, sólo los valores de coeficiente de transformada de posiciones de barrido 0 a 6, 8 y 9 pertenecen a las capas de calidad de orden cero. Los demás valores de coeficiente de transformada están ajustados

a cero. Ha de enfatizarse que en otros bloques de transformada, los valores de coeficiente de transformada pertenecientes a la capa de calidad de orden cero pueden pertenecer a otras posiciones de barrido. De manera similar, los valores de coeficiente de transformada de posiciones de barrido 7, 10 a 12, 15 a 18 y 21 pertenecen a la siguiente capa de calidad más alta. Los valores de coeficiente de transformada restantes se ajustan a cero. Los valores de coeficientes restantes de las posiciones de barrido restantes se incluyen en el siguiente subflujo de datos de capa de calidad más alta. Como puede observarse, puede ser posible que un determinado valor de coeficiente de transformada sea realmente cero. En el ejemplo de la figura 6a, éste es el caso para la posición de barrido 23. Los correspondientes valores de contribución en las capas de calidad precedentes se ajustan a cero y el valor de coeficiente de transformada para la posición de barrido 23 en la última capa de calidad (última línea) para la posición de barrido 23 es cero a su vez.

Por tanto, para cada una de las posiciones de barrido, los valores de contribución incluidos en los diversos subflujos 28 y 30 de bits de capas de calidad, suman el valor de coeficiente de transformada real de modo que, en el lado del decodificador, el bloque de transformada real puede reconstruirse sumando los valores de contribución para las posiciones de barrido individuales de las diferentes capas de calidad.

Según la realización de la figura 6a, cada uno de los subflujos 28 y 30 de datos comprende un valor de contribución para todos los coeficientes de transformada y para todas las posiciones de barrido, respectivamente. Sin embargo, éste no es necesariamente el caso. En primer lugar, tal como ya se mencionó anteriormente, no es necesario que el subflujo 28 de datos de orden cero contenga algún coeficiente de transformada o valor de contribución. Así, en este último caso, las últimas tres líneas de la tabla de la figura 6a pueden considerarse como pertenecientes a los primeros subflujos 30 de datos de capa de refinamiento, comprendiendo el subflujo 28 de datos de orden cero meramente la información de movimiento de la entrada 124.

Además, se indica que los valores de contribución de la figura 6a ajustados a cero y los valores de coeficiente de transformada reales que son realmente cero se han distinguido usando las entradas de tabla sombreadas únicamente con vistas a una mejor comprensión de la funcionalidad de la unidad 128 de información. Sin embargo, los subflujos 28 y 30 de datos pueden construirse de tal manera que la distinción que acaba de mencionarse entre valores de contribución ajustados a cero y valores de contribución que son cero de manera natural sea transparente para el decodificador. De manera más precisa, algunos de los respectivos valores de contribución para respectivas posiciones de barrido, es decir los números desde la segunda hasta la cuarta línea bajo una respectiva posición de barrido en la primera línea de la figura 6a revelan el valor de coeficiente de transformada independientemente de que los valores de contribución individuales en la suma se ajusten a cero o sean cero de manera natural.

En la realización de la figura 6a, la unidad 128 de formación ha codificado en un subflujo respectivo de los subflujos 28 y 30 de datos, respectivamente, un valor de contribución para cada una de las posiciones de barrido. Esto no es necesario. Según la realización de la figura 6b, por ejemplo, los subflujos de datos con capas de calidad consecutivas comprenden únicamente aquellos valores de coeficiente de transformada pertenecientes a la respectiva capa de calidad.

El orden en el que los valores de contribución y los valores de coeficiente de transformada se codifican en los subflujos 28 y 30 de datos respectivamente, puede variar en las realizaciones de la figura 6a y la figura 6b, respectivamente. Por ejemplo, los subflujos 28 y 30 de datos pueden ser flujos de datos paquetizados en los que cada paquete corresponde a un segmento 168. En un segmento 168, los valores de coeficiente de transformada pueden codificarse en los respectivos paquetes por macrobloques. Es decir, puede definirse un orden de barrido entre los macrobloques 142 dentro de un segmento 168 con los valores de coeficiente de transformada para un macrobloque 142 predeterminado completamente codificados en el respectivo paquete antes del primer valor de coeficiente de transformada de un macrobloque siguiendo un orden de barrido de macrobloque. Dentro de cada macrobloque, puede definirse un orden de barrido entre los respectivos bloques 146 y 148 de transformada, respectivamente, dentro del respectivo macrobloque. De nuevo, los valores de coeficiente de transformada pueden codificarse en un subflujo respectivo de los subflujos 28 y 30 de datos, respectivamente mediante la unidad 128 de formación, de tal manera que los valores de coeficiente de transformada de un bloque respectivo de los bloques de transformada se codifican todos en el respectivo subflujo de datos antes de que el primer valor de coeficiente de transformada de un siguiente bloque de transformada se codifique en el mismo. Dentro de cada bloque de transformada, puede realizarse una codificación de los valores de coeficiente de transformada y los valores de contribución, respectivamente, de la manera explicada a continuación con respecto a la figura 8 ó 9.

Según las realizaciones de las figuras 6a y 6b, los valores de coeficiente de transformada de los diferentes bloques de transformada del segmento 168 pertenecientes a una capa respectiva de las capas de calidad, se han extendido por una parte diferente del orden de barrido. De manera más precisa, aunque en el bloque de transformada específico mostrado a modo de ejemplo en las figuras 6a y 6b, las

posiciones de barrido 0 a 6, 8 y 9 pertenecen a la capa de calidad de orden cero, en otro bloque de transformada, el conjunto de posiciones de barrido perteneciente a esta capa puede ser diferente. Según la realización de la figura 6c, sin embargo, la unidad 126 de distribución distribuye los valores de coeficiente de transformada de los diferentes bloques de transformada dentro de un segmento 168 de tal manera que, para todos los bloques de transformada, los valores de coeficiente de transformada del mismo conjunto de posiciones de barrido pertenecen a la misma capa de calidad. Por ejemplo, en la figura 6c los valores de coeficiente de transformada de las posiciones de barrido de 0 a 11 pertenecen al subflujo 28 de datos de orden cero, siendo esto cierto para todos los bloques de transformada dentro del segmento 168.

Según la realización de la figura 6c, además, los valores de coeficiente de transformada pertenecientes a una capa específica de las capas de calidad se extienden por una secuencia continua de posiciones de barrido consecutivas. Sin embargo, éste no tiene por qué ser el caso. En particular, los valores de coeficiente de transformada pertenecientes a una posición de barrido entre la primera y la última posición de barrido pertenecientes a una capa de calidad específica pueden pertenecer a una de las otras capas de calidad tal como se muestra en la figura 6b. Sin embargo, en el caso de la realización de la figura 6c, es posible indicar las posiciones de barrido incorporadas en cualquiera de los subflujos 28 y 30 de datos de capas de calidad, respectivamente, usando meramente dos elementos de sintaxis, uno indicando la primera posición de barrido de la respectiva capa de calidad, es decir scan_idx_start y el otro indicando la última posición de barrido para la respectiva capa de calidad, es decir scan_idx_end.

La reserva de un conjunto específico de posiciones de barrido para una respectiva capa de las capas de calidad por un lado y la distribución de los coeficientes de transformada en función de la importancia de la calidad a las capas de calidad individuales por otro lado, pueden combinarse tal como se muestra en la siguiente realización. Por ejemplo, la figura 6d muestra una realización en la que la unidad 126 de distribución ha distribuido los coeficientes de transformada por las capas de calidad tal como se mostró con respecto a la figura 6a. Esta distribución difiere de un bloque de transformada a otro. Sin embargo, por otro lado, a cada una de las capas de calidad se le asigna una parte específica de las posiciones de barrido en común para todos los bloques de transformada. Por ejemplo, a la capa de calidad más baja se le asigna el conjunto completo de posiciones de barrido desde la posición de barrido 0 a la posición de barrido 63. Por tanto, para cada bloque de transformada, la capa de calidad más baja comprende 64 valores de contribución. El siguiente subflujo de datos de capa de calidad más alta comprende valores contribución o de coeficientes de transformada para todos los bloques de transformada en un intervalo específico de posiciones de barrido que se extiende desde la posición de barrido 6 a la 63. El intervalo de posiciones de barrido de la siguiente capa de calidad se extiende desde la posición de barrido 13 a la 63. De nuevo, el decodificador no necesita conocer si un valor específico de los valores de contribución es un valor de contribución que se ha ajustado a 0 (entrada sombreada) o si realmente indica un valor de coeficiente de transformada 0 o un valor de coeficiente de transformada despreciable. Sin embargo, tiene que conocer el elemento de sintaxis scan_idx_start que indica, para el respectivo segmento 168, a partir de qué posición de barrido van a usarse en los valores de coeficiente de transformada o de contribución contenidos en el respectivo subflujo de datos. De manera más precisa, en la realización de la figura 6d, por ejemplo, el subflujo de datos correspondiente a la penúltima línea comprende, para un bloque 58 de transformada individual, valores de coeficiente de transformada o de contribución. El primero, en el caso del bloque de transformada de la figura 6d, es 0, mientras que el segundo es 22. Mediante el uso del elemento de sintaxis scan_idx_start en el lado del decodificador, se sabe que el primer valor de coeficiente de transformada de la respectiva capa de calidad corresponde a la posición de barrido 6, mientras que los valores de coeficiente de transformada restantes de esta capa de calidad se refieren a las siguientes posiciones de barrido. De manera similar a las realizaciones de la figura 6d, la figura 6e muestra una realización en la que un elemento de sintaxis scan_idx_end indica para los subflujos de datos individuales, la última posición de barrido hasta la que el respectivo subflujo de datos de capa de calidad comprende subcoeficientes o valores de contribución.

Una combinación de las realizaciones de las figuras 6d y 6e se muestra en la figura 6f. Según esta realización, el respectivo conjunto de posiciones de barrido perteneciente a una capa específica de las capas de calidad se extiende desde una primera posición de barrido indicada mediante el elemento de sintaxis scan_idx_start hasta una última posición de barrido indicada mediante el elemento de sintaxis last_idx_end. Por ejemplo, en la capa de calidad correspondiente a la penúltima línea, el respectivo conjunto de posiciones de barrido se extiende desde la posición de barrido 6 hasta la posición de barrido 21. Finalmente, la realización de la figura 6g muestra que el uso del elemento de sintaxis scan_idx_start y/o scan_idx_end pueden combinarse con el objeto de la realización de la figura 6c según la cual la distribución de los valores de coeficiente de transformación individuales de los diferentes bloques de transformada dentro de un segmento 168 es común para los bloques de transformada. Por consiguiente, según la realización de la figura 6g, dentro de una capa específica de las capas de calidad, todos los valores de coeficiente de transformada dentro de scan_idx_start hasta scan_idx_end se distribuyen a la respectiva capa de calidad. Por tanto, a diferencia de la realización de la figura 6f, en la realización de la figura 6g, todos los valores de coeficiente de transferencia dentro de la posición de barrido 6 hasta la

posición de barrido 21 se asignan a la capa de calidad correspondiente la penúltima línea en la figura 6g. A diferencia de ello, en la realización de la figura 6f, varios de los valores de contribución dentro de este intervalo de posiciones de barrido desde la 6 hasta la 21 pueden ajustarse a 0, pudiendo ser la distribución de valores de coeficiente de transformada que se han ajustado a 0 y los valores de coeficiente de transformada que no se han ajustado a 0 dentro de este intervalo de posiciones de barrido desde la 6 hasta la 21, diferente de cualquier otro bloque de transformada dentro del actual segmento.

A continuación se describe de manera ilustrativa la cooperación entre el codificador 42 híbrido, la unidad 44 de codificación en capas, la unidad 126 de distribución y la unidad 128 de formación, con respecto a la figura 7 que muestra un ejemplo de la estructura de los subflujos 28 y 30 de datos, respectivamente. Según la realización de la figura 7, la unidad 28 de formación está diseñada de tal manera que los subflujos 28 y 30 de datos individuales, respectivamente, se paquetizan, es decir comprenden uno o más paquetes. En particular, la unidad 128 de formación puede diseñarse para generar un paquete para cada segmento 168 dentro de una imagen 140 dentro de cada subflujo 28 y 30 de bits, respectivamente. Tal como se muestra en la figura 7, un paquete puede comprender una cabecera 170 de segmento por un lado y datos 172 residuales por otro lado, a excepción del subflujo 28 de bits que opcionalmente comprende únicamente la cabecera de segmento dentro de cada uno de los paquetes.

Con respecto a la descripción de los datos 172 residuales, es decir los datos residuales #1, los datos residuales #2,..., los datos residuales #N, se hace referencia a la descripción anterior con respecto a las figuras 6a a 6g, donde por ejemplo, las líneas segunda a cuarta en estas tablas corresponden a los datos residuales #1, los datos residuales #2 y los datos residuales #3, por ejemplo. En otras palabras, los datos 172 residuales indicados en la figura 7 incluyen los valores de coeficiente de transformada comentados en las figuras 6a a 6g, cuya distribución entre los respectivos subflujos 28 y 30 de datos no vuelve a describirse aquí. Sin embargo, la figura 7 muestra elementos de sintaxis adicionales contenidos en la cabecera 170 de segmento y los datos 172 residuales procedentes del codificador 42 híbrido. Tal como se describió anteriormente, el codificador 42 híbrido conmuta, por macrobloques, entre varios modos de predicción entre capas de modo que se basa en la información de movimiento de la capa base, o genera nueva información de movimiento para un respectivo bloque de movimiento de la capa de refinamiento superior con codificación predictiva de la información de movimiento como un residuo respecto de la información de movimiento de la capa base, o con codificación de esta información de movimiento de nuevo. Por tanto, tal como se indica en la figura 7, los datos 172 residuales pueden comprender, para cada macrobloque, elementos de sintaxis que indican parámetros de movimiento, modos de macrobloque tales como codificación en campo o en trama, o un modo de inferencia que indica la reutilización de los parámetros de movimiento de la capa base con el respectivo macrobloque. Esto es especialmente cierto para el subflujo 28 de datos o cero. Sin embargo, esta información de movimiento no vuelve a refinarse en las siguientes capas de refinamiento y los siguientes subflujos 30_1 a 30_N de datos de calidades más altas, y por tanto, la unidad 128 de formación está diseñada para omitir estos elementos de sintaxis por macrobloques relativos a modos de macrobloque, parámetros de movimiento e indicación de modo de inferencia en los datos residuales de estos subflujos 30_1 a 30_N de datos o para ajustar los elementos de sintaxis en estos subflujos 30_1 a 30_N de datos para que sean o bien iguales a los modos de macrobloque y los parámetros de movimiento para el respectivo macrobloque contenido en el subflujo 28 de datos o bien para indicar el modo de inferencia para el respectivo macrobloque con el fin de indicar que deben usarse los mismos ajustes en la respectiva capa de refinamiento. Según la realización de la presente invención, todos los datos 172 residuales dentro de los diversos subflujos 28 y 30_1 a 30_N de datos se pasan usando la misma estructura de sintaxis de modo que también los datos residuales dentro de los subflujos 30_1 a 30_N de datos de refinamiento comprenden información definida por macrobloques sobre la activación/desactivación de modo de macrobloque, parámetro de movimiento y/o modo de inferencia.

Tal como puede deducirse también de la figura 7, la unidad 128 de formación puede estar diseñada para proporcionar a la cabecera 170 de segmento el elemento de sintaxis scan_idx_start y/o scan_idx_end. Alternativamente, los datos 170 de la cabecera de segmento pueden comprender otros elementos de sintaxis que definen para cada segmento o paquete individual, un conjunto de posiciones de barrido a las que se refieren los datos residuales correspondientes a los respectivos datos de cabecera de segmento. Tal como ya se indicó anteriormente, los datos de cabecera de segmento de paquetes del subflujo 28 de datos pueden no comprender elementos de sintaxis relativos a la definición de posiciones de barrido específicas de capa en caso de que el subflujo 28 de datos no comprenda datos residuales, sino meramente modos de macrobloque y/o parámetros de movimiento e indicaciones de modo de inferencia, respectivamente. Además, tal como ya se indicó anteriormente, los datos 170 de cabecera de segmento pueden comprender sólo uno de scan_idx_start y scan_idx_end. Finalmente, scan_idx_start y/o scan_idx_end pueden proporcionarse una vez por categoría de tamaño de bloque de transformada, es decir 4x4 y 8x8, o sólo una vez para cada segmento/imagen/subflujo de datos de manera común para todas las categorías de tamaño de bloque de transformada, tomándose medidas respectivas para transferir scan_idx_start y scan_idx_end a otros tamaños de bloque tal como se describirá a continuación.

Además, los datos de cabecera de segmento pueden comprender un elemento de sintaxis que indica el nivel de calidad. Para ello, la unidad 128 de formación puede diseñarse de tal manera que el elemento de sintaxis o indicador de calidad distingue simplemente entre el nivel 28 de calidad de orden cero por un lado y las capas 30_1 a 30_N de refinamiento por otro lado. Alternativamente, el indicador de calidad puede distinguir todas las capas de calidad entre las capas 28 y 30_1 a 30_N de refinamiento. En los últimos dos casos, el indicador de calidad permitiría la omisión de cualquier modo de macrobloque, parámetro de movimiento y/o modo de inferencia, definidos por macrobloques, dentro de los paquetes de los subflujos 30_1 a 30_N de datos ya que, en este caso, en el lado del decodificador, se sabe que estos subflujos 30_1 a 30_N de datos de capas de refinamiento refinan solamente los coeficientes de transformada usando los modos de macrobloque, parámetros de movimiento y modos de inferencia desde el subflujo 28 de datos de modo cero.

Aunque no se ha descrito con más detalle anteriormente, la unidad 28 de formación puede diseñarse para realizar codificación de entropía sobre los paquetes dentro de los subflujos 28 y 30_1 a 30_N de datos. En esta realización, las figuras 8 y 9 muestran posibles ejemplos para codificar los coeficientes de transformada en los datos residuales pertenecientes a un bloque de transformada según dos realizaciones. La figura 8 muestra un pseudocódigo de un primer ejemplo para una posible codificación de los coeficientes de transformada en un bloque de transformada en cualquiera de los datos 172 residuales. Suponiendo que se aplica el siguiente ejemplo:

Posición de barrido	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Número de coeficientes	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Nivel de coeficientes de transformada	7	6	-2	0	-1	0	0	1	0	0	0	0	0	0	0	0

Basándose en este ejemplo, a continuación se explica el pseudocódigo de la figura 8 mostrando el modo en que la unidad 128 de formación puede codificar los datos residuales en uno de los bloques de transformada en cualquiera de los subflujos de datos.

Con el fin de transportar la información de los coeficientes de transformada, según la figura 8, en primer lugar se proporciona un parámetro coeff_token 240. El parámetro coeff_token es una palabra de código que indica el número de coeficientes distintos de cero, es decir total_coeff(coeff_token), y el número de coeficientes de transformada en la serie de coeficientes de transformada que tienen un valor absoluto igual a uno al final de la secuencia de coeficientes de transformada distintos de cero, es decir trailing_ones(coeff_token). En este ejemplo, total_coeff(coeff_token) es 5 (números de coeficientes de transformada 0,1,2,4 y 7) y trailing_ones(coeff_token) es 2 (número de coeficientes de transformada 4 y 7). Por tanto, al proporcionar el parámetro coeff_token 240, las posiciones de los coeficientes de transformada significativos se han determinado en la medida en que no existen más de total_coeff(coeff_token) coeficientes de transformada distintos de cero.

Entonces se proporcionan los valores de los niveles de estos coeficientes de transformada distintos de cero. Esto se realiza en orden de barrido inverso. De manera más específica, en primer lugar se comprueba si el número total de coeficientes de transformada distintos de cero es mayor que cero 242. Éste es el caso en el ejemplo anterior, puesto que total_coeff(coeff_token) es 5.

Entonces, se avanza gradualmente por los coeficientes no de transformada a través de un orden de barrido inverso 244. El orden de barrido inverso todavía no es obvio simplemente a partir la observación del incremento del parámetro de recuento i++ en el bucle 244 *for* (para), pero resultará evidente a partir de la siguiente evaluación. Mientras se avanza gradualmente a través de estos coeficientes no de transformada en orden de barrido inverso, para el primero de estos coeficientes de transformada distintos de cero, sólo se proporciona su signo de coeficientes de transformada 248. Esto se realiza para el primer número de trailing_ones(coeff_token) de los coeficientes de transformada distintos de cero cuando se avanza gradualmente a través de los mismos en un orden de barrido inverso, ya que para estos coeficientes de transformada ya se sabe que el valor absoluto de estos coeficientes de transformada es uno (compárese con la definición anterior de trailing_ones(coeff_token)). Los signos de coeficientes así proporcionados se usan para almacenar temporalmente en coeficientes de vector auxiliares level[i] para el nivel de coeficientes de transformada de los niveles de coeficientes de transformada distintos de cero que tienen el valor absoluto de 1, donde i es una numeración de los coeficientes de transformada distintos de cero cuando se escanean en orden de barrido inverso (250). En

este ejemplo, después de las dos primeras rondas del bucle 244 *for*, se obtiene level[0] = 1 y level[1] = -1.

A continuación, los niveles de coeficientes coeff_level para los coeficientes de transformada distintos de cero restantes se proporcionan (252) en orden de barrido inverso y se almacenan temporalmente en los coeficientes de vector auxiliares level[i] (254). El resto de ciclos del bucle *for* dan como resultado level[2] = -2, level[3] = 6 y level[4] = 7.

Ahora, con el fin de hacer que la determinación de las posiciones de los coeficientes de transformada significativos sea única, se proporcionan dos parámetros adicionales denominados total_zeros y run_before a menos que total_coeff(coeff_token) ya sea igual al número máximo de coeficientes de transformada en un bloque de transformada, es decir, sea igual a maxNumCoeff. De manera más específica, se comprueba si total_coeff(coeff_token) es igual a maxNumCoeff (256). Si éste no es el caso, se proporciona (258) el parámetro total_zeros y un parámetro auxiliar zerosLeft se inicializa al valor de total_zero (260). El parámetro total_zeros especifica el número de ceros entre el último coeficiente distinto de cero en el orden de barrido y el inicio del barrido. En el ejemplo anterior, total_zeros es 3 (números de coeficientes 3, 5 y 6). Por tanto, zerosLeft se inicializa a 3.

Para cada uno de los coeficientes de transformada distintos de cero excepto el último con respecto al orden de barrido inverso (número de coeficientes 0), empezando con el último coeficiente de transformada distinto de cero (número de coeficientes 7) con respecto al orden de barrido (62), se proporciona un parámetro run_before (64) que indica la longitud de la serie de coeficientes de transformada de nivel cero dispuesta directamente en frente del respectivo coeficiente de transformada distinto de cero visto en orden de barrido. Por ejemplo, para i igual a cero, el último coeficiente de transformada distinto de cero con respecto al orden de barrido es el coeficiente de transformada distinto de cero en cuestión. En este ejemplo, es el coeficiente de transformada que tiene el número 7 y el nivel 1. La serie de ceros en frente de este coeficiente de transformada tiene una longitud de 2, es decir, los coeficientes de transformada 5 y 6. Por tanto, en este ejemplo, el primer parámetro run_before es 2. Este parámetro se almacena temporalmente en el coeficiente de vector auxiliar run[0] (266). Esto se repite en orden de barrido inverso para run[i], siendo i el recuento de los coeficientes de transformada distintos de cero cuando se realiza el barrido en orden de barrido inverso. Disminuyendo el parámetro auxiliar zerosLeft por el parámetro run_before en cada ciclo del bucle *for* (261) se determina para cada ciclo cuántos coeficientes de transformada de nivel cero quedan. Si zerosLeft es cero, ya no se proporciona ningún parámetro run_before (270) y los coeficientes restantes de la serie de vectores se ajustan a cero (272). En cualquier caso, no se proporciona ningún parámetro run_before para el último coeficiente de transformada distinto de cero cuando se avanza progresivamente en orden de barrido inverso, es decir, ningún parámetro run_before para el primer coeficiente de transformada distinto de cero con respecto al orden de barrido. Este parámetro se deduce a partir del número de coeficientes de transformada de nivel cero que quedan, indicado por el parámetro auxiliar zerosLeft (274). En este ejemplo, los coeficientes de vector auxiliares de la serie de vectores son run[0] = 2, run[1] = 1, run[2]=0, run[3]=0, y run[4]=0.

Al final, en el bucle *for* indicado en 276, los valores de los niveles de coeficientes de transformada almacenados en nivel de vector auxiliar se asignan a sus posiciones copiando los valores de los coeficientes de nivel de vector para la respectiva posición en la matriz unidimensional coeff_Level. De manera más específica, en el primer ciclo del bucle *for* 276, i = 4 y coeffNum que se inicializó a 0 (278) se incrementa por run[4] + 1 = 0 + 1 = 1 dando como resultado coeffNum = 0 y asignándose a coeffLevel[0] el valor de level [4] = 7. Esto se repite para los siguientes coeficientes de vector auxiliares level[3] a level[0]. Puesto que las posiciones restantes de la matriz coeffLevel se han inicializado al valor de cero (280), todos los coeficientes de transformada se han codificado.

Los elementos de sintaxis en negrita en la figura 8 pueden codificarse en el respectivo subflujo de datos mediante codificación de longitud variable, por ejemplo.

La figura 9 muestra otro ejemplo para codificar un bloque de transformada. En este ejemplo, el orden de barrido se pone de manifiesto en “i++” en el bucle 310 *while* (mientras) indicando que el parámetro de recuento i se incrementa por cada iteración del bucle *while*.

Para cada coeficiente en el orden de barrido, se proporciona un símbolo de un bit significant_coeff_flag (312). Si el símbolo significant_coeff_flag es 1 (314), es decir, si existe un coeficiente distinto de cero en esta posición de barrido i, se proporciona un símbolo de un bit adicional last_significant_coeff_flag (316). Este símbolo indica si un coeficiente significativo actual es el último dentro del bloque o si coeficientes significativos adicionales siguen en el orden de barrido. Por tanto, si el símbolo last_significant_coeff_flag es uno (318), esto indica que el número de coeficientes, es decir numCoeff, es i+1 (320) y puede deducirse que los niveles de los posteriores coeficientes de transformada son cero (322). A este respecto, los elementos de sintaxis last_significant_coeff_flag y significant_coeff_flag pueden considerarse como un mapa de significancia. Entonces, para el último coeficiente de transformada en orden de barrido, se proporciona el valor absoluto del nivel menos 1, es

decir, coeff_abs_level_minusl, y su signo, es decir coeff_sign_flag, (324), indicando de este modo el nivel de coeficientes de transformada de este último coeficiente de transformada significativo (326). Estas etapas 324, 326 se repiten (328) para los restantes coeficientes de transformada significativos (330) en orden de barrido inverso (332), poniéndose de manifiesto el orden de barrido inverso en i--, es decir i = i - 1, que indica que el parámetro de recuento i disminuye cada ciclo en el bucle *for*. El análisis sintáctico de los elementos de sintaxis coeff_abs_level_minusl empieza derivando una binarización para los posibles valores del elemento de sintaxis. El esquema de binarización puede ser un UEG0, es decir, un proceso de binarización Exp-Golomb unario/de orden cero concatenado. Dependiendo de las posibles binarizaciones, el respectivo elemento de sintaxis puede codificarse aritméticamente de manera binaria de intervalo a intervalo. A este respecto, puede usarse un esquema de codificación aritmética binaria con adaptación al contexto para una parte de prefijo de la binarización de coeff_abs_level_minusl mientras se usa un proceso de derivación de decodificación sin adaptación para una parte de sufijo.

Para completar, se indica que quedó claro a partir de la figura 5 que el número de posiciones de barrido distinguibles dentro de los bloques de transformada de 8x8 es de 64 mientras que el número posiciones de barrido distinguibles dentro de los bloques de transformada de 4x4 es de sólo 16. Por consiguiente, el elemento de sintaxis anteriormente mencionado scan_idx_start y scan_idx_end puede definirse o bien con una precisión que permita una distinción entre las 64 posiciones de barrido, o meramente una distinción entre 16 posiciones de barrido. En este último caso, por ejemplo, los elementos de sintaxis pueden aplicarse a cada cuádruplo de coeficientes de transformada consecutivos en los bloques de transformada de 8x8. De manera más precisa, los bloques de transformada de 8x8 pueden codificarse usando

residual_block(LumaLevel8x8,

4*scan_idx_start, 4*scan_ids_end +3, 64)

y en el caso de bloques de transformada de 4x4 usando

residual_block(LumaLevel4x4,

scan_idx_start, scan_idx_end, 16).

siendo residual_block o bien residual_block_cavlc o residual_block_cabac, e indicando LumaLevel4x4 y LumaLevel8x8 una matriz de muestras de luma del respectivo bloque de transformada de 4x4 y 8x8, respectivamente. Como puede verse, scan_idx_start y scan_idx_end se definen para discriminar entre 16 posiciones de barrido de modo que indiquen el intervalo de posiciones en bloques de 4x4 de manera exacta. Sin embargo, en bloques de 8x8, la precisión de estos elementos de sintaxis no es suficiente de modo que en estos bloques el intervalo se ajusta de manera cuádruple.

Además, también pueden codificarse bloques de coeficientes de transformada de 8x8 dividiendo los 64 coeficientes de un bloque de 8x8 en 4 conjuntos de 16 coeficientes, por ejemplo colocando cada cuarto coeficiente en el conjunto enésimo empezando con el coeficiente n, con n en el intervalo de 0 a 3, inclusive, y codificando cada conjunto de 16 coeficientes usando la sintaxis de bloque residual para bloques de 4x4. En el lado del decodificador, estos 4 conjuntos de 16 coeficientes se recombinan para formar un conjunto de 64 coeficientes que representa un bloque de 8x8.

Tras haber descrito realizaciones para un codificador, se explica un decodificador para decodificar el respectivo flujo de datos con calidad ajustable a escala con respecto a las figuras 10 y 11. La figura 10 muestra la construcción general de un decodificador 400. El decodificador 400 comprende un demultiplexor 402 que tiene una entrada 404 para recibir el flujo 36 de bits ajustable a escala. El demultiplexor 402 demultiplexa la señal 36 de entrada en los flujos 26 a 32 de datos. Para ello, el demultiplexor puede realizar una función de decodificación y/o análisis sintáctico. Por ejemplo, el demultiplexor 402 puede decodificar las codificaciones de bloque de transformada de la figura 8 y 9. Además, se remite a la figura 6a-6g. Por consiguiente, el demultiplexor 402, puede usar información de subflujos de datos precedentes con el fin de conocer, al analizar sintácticamente un subflujo de datos actual, cuántos valores de coeficiente de transformada o valores de contribución se esperan para un bloque de transformada específico. Los flujos de datos así recuperados se reciben en una unidad 406 de decodificación que, basándose en estos flujos de datos, reconstruye el vídeo 18 y emite el respectivo vídeo 408 reconstruido en una salida 410 respectiva.

La estructura interna de la unidad 406 de decodificación se muestra más detalladamente en la figura 11. Tal como se muestra en la misma, la unidad 406 de decodificación comprende una entrada 412 de datos de movimiento de capa base, una entrada 414 de datos residuales de capa base, una entrada 416 de datos de movimiento de capa de refinamiento de orden cero, una entrada 418 de datos de coeficientes de transformada de refinamiento de orden cero de coeficientes de transformada opcional y una entrada 420 para los subflujos 30 de datos. Como se muestra, las entradas 412 y 414 son para recibir

el flujo 26 de datos, mientras que las entradas 416 y 418 actúan conjuntamente para recibir el flujo 28 de datos. Además de esto, la unidad 406 de decodificación comprende una salida 422 de señal de vídeo de reconstrucción de calidad más baja, una salida 424 de señal de vídeo de reconstrucción con codificación entre capas de calidad más alta, y una salida 426 de señal de vídeo de reconstrucción codificada internamente, proporcionando estas últimas la información para una señal de vídeo de calidad más alta.

Un combinador 428 tiene entradas conectadas a las entradas 418 y 420 y una salida para emitir niveles de coeficientes de transformada para los bloques de transformada individuales obtenidos recopilando los correspondientes valores de contribución a partir de las diversas capas de calidad. La recopilación puede implicar una suma de los valores de contribución para un coeficiente de transformada específico dentro de varios de los flujos 30 y 28 de datos. Sin embargo, también es posible que el combinador 428 presente todos los valores de coeficiente de transformada en cero y sustituya cualquiera de estos ceros únicamente en caso de que un valor de contribución sea distinto de cero para la respectiva posición de barrido. Mediante esta medida, el combinador recopila información sobre los coeficientes de transformada de los diversos bloques de transformada. La asociación de los valores de contribución o de coeficientes de transformada en las capas individuales puede implicar que el combinador use la información de posición de barrido de la capa actual tal como scan_idx_start y/o scan_idx_end. Alternativamente, el combinador puede usar el conocimiento de los valores de coeficiente de transformada en los bloques de transformada individuales recibidos hasta ahora desde capas de SNR de calidad más baja

Los bloques de transformada emitidos por el combinador 428 los recibe un decodificador 430 predictivo residual y un sumador 432.

Entre el decodificador 430 predictivo residual y la entrada 414, se conecta una unidad 432 de transformación hacia atrás o inversa con el fin de reenviar datos residuales transformados inversamente al decodificador 430 predictivo residual. Este último usa los datos residuales transformados inversamente con el fin de obtener un predictor que va a añadirse a los coeficientes de transformada de los bloques de transformada emitidos por el combinador 428, eventualmente tras realizar un sobremuestreo u otra adaptación de calidad. Por otro lado, una unidad 434 de predicción de movimiento está conectada entre la entrada 412 y una entrada de un sumador 436. Otra entrada del sumador 436 está conectada a la salida de una unidad 432 de transformación hacia atrás. Mediante esta medida, la unidad 434 de predicción de movimiento usa los datos de movimiento en la entrada 412 para generar una señal de predicción para la señal residual transformada inversamente emitida por la unidad 432 de transformación hacia atrás. Un resultado del sumador 436 en la salida del sumador 436 es una señal de vídeo de capa base reconstruida. La salida del sumador 436 se conecta a la salida 432 así como a una entrada del decodificador 432 predictivo. El decodificador 432 predictivo usa la señal de capa base reconstruida como predicción para las partes de capas intracodificadas del contenido de vídeo emitido por el combinador 428, eventualmente usando un sobremuestreo. Por otro lado, la salida del sumador 436 también está conectada a una entrada de la unidad 434 de predicción de movimiento con el fin de permitir que la unidad 434 de predicción de movimiento utilice los datos de movimiento en la entrada 412 para generar una señal de predicción para la segunda entrada del sumador 436 basándose en las señales reconstruidas a partir del flujo de datos de capa base. Los valores de coeficiente de transformada decodificados de manera predictiva emitidos por el decodificador 430 predictivo residual se transforman hacia atrás mediante la unidad 438 de transformación hacia atrás. En la salida de la unidad 438 de transformación hacia atrás, se obtienen como resultado datos de señal de vídeo residuales de calidad más alta. Esta señal de vídeo de datos residuales de calidad más alta se suma por un sumador 440 con una señal de vídeo con predicción de movimiento emitida por una unidad 442 de predicción de movimiento. En la salida del sumador 440, se obtiene como resultado la señal de vídeo de calidad más alta reconstruida que llega a la salida 424 así como a una entrada adicional de la unidad 442 de predicción de movimiento. La unidad 442 de predicción de movimiento realiza la predicción de movimiento basándose en la señal de vídeo reconstruida emitida por el sumador 440 así como en la información de movimiento emitida por un decodificador 444 de predicción de parámetros de movimiento que está conectado entre la entrada 416 y una respectiva entrada de la unidad 442 de predicción de movimiento. El decodificador 444 predictivo de parámetros de movimiento usa, de manera selectiva por macrobloques, datos de movimiento de la entrada 412 de datos de movimiento de capa base como predictor, y en función de estos datos, emite los datos de movimiento a la unidad 442 de predicción de movimiento usando, por ejemplo, los vectores de movimiento en la entrada 416 como vectores de desfase para vectores de movimiento en la entrada 412.

Las realizaciones anteriormente descritas permiten un aumento de la granularidad de la codificación de SNR ajustable a escala en cuanto a imagen/segmento en comparación con la codificación de CGS/MGS descrita en la parte introductoria, pero sin el significativo aumento de complejidad presente en la codificación de FGS. Además, puesto que se cree que la característica de FGS de que los paquetes pueden truncarse no se usará de manera generalizada, la adaptación del flujo de bits es posible simplemente mediante eliminación de paquetes.

Las realizaciones anteriormente descritas tienen en común la idea básica de dividir los niveles de coeficientes de transformada de un paquete de CGS/MGS tradicional, tal como se especifica actualmente en el borrador de SVC, en subconjuntos, que se transmiten en diferentes paquetes y diferentes capas de refinamiento de SNR. Como ejemplo, las realizaciones anteriormente descritas se refieren a la codificación de CGS/MGS con una capa base y una de mejora. En lugar de que la capa de mejora incluya, para cada imagen, modos de macrobloque, modos de intrapredicción, vectores de movimiento, índices de imagen de referencia, otros parámetros de control así como niveles de coeficientes de transformada para todos los macrobloques, con el fin de aumentar la granularidad de la codificación de SNR ajustable a escala, estos datos se han distribuido por diferentes segmentos, diferentes paquetes, y diferentes capas de mejora. En la primera capa de mejora se transmiten los modos de macrobloque, los parámetros de movimiento, otros parámetros de control así como, opcionalmente, un primer subconjunto de niveles de coeficientes de transformada. En la siguiente capa de mejora, se usan los mismos modos de macrobloque y vectores de movimiento, pero se codifica un segundo subconjunto de niveles de coeficientes de transformada. Todos los coeficientes de transformada que ya se han transmitido en la primera capa de mejora pueden ajustarse a cero en la segunda y en todas las capas de mejora siguientes. En todas las capas de mejora siguientes (tercera, etc.) vuelven a usarse los modos de macrobloque y los parámetros de movimiento de la primera capa de mejora, pero se codifican subconjuntos de niveles de coeficientes de transformada adicionales.

Ha de indicarse que esta división no aumenta, o sólo lo hace muy ligeramente, la complejidad en comparación con la codificación de CGS/MGS tradicional especificada en el borrador de SVC actual. Todas las mejoras de SNR pueden analizarse sintácticamente en paralelo, y los coeficientes de transformada no tienen que recopilarse a partir de diferentes barridos por la imagen/segmento. Esto significa, por ejemplo, que un decodificador puede analizar sintácticamente todos los coeficientes de transformada para un bloque a partir de todas las mejoras de SNR, y entonces puede aplicar la transformada inversa para este bloque sin almacenar los niveles de coeficientes de transformada en una memoria intermedia temporal. Cuando todos los bloques de un macrobloque se han analizado sintéticamente por completo, puede aplicarse la predicción con compensación de movimiento y puede obtenerse la señal de reconstrucción final para este macrobloque. Ha de indicarse que todos los elementos de sintaxis en un segmento se transmiten macrobloque por macrobloque, y dentro de un macrobloque, los valores de coeficiente de transformada se transmiten bloque de transformada por bloque de transformada.

Es posible que esté codificado al nivel del segmento un indicador que señala si todos los modos de macrobloque y parámetros de movimiento se infieren a partir de la capa base. Dada la sintaxis actual de los paquetes de CGS/MGS, esto significa especialmente que no se transmiten todos los elementos de sintaxis mb_skip_run y mb_skip_flag sino que se infiere que son igual a 0, que no se transmiten todos los elementos de sintaxis mb_field_decoding_flag sino que se infiere que son iguales a sus valores en los macrobloques de capa base coubicados, y que no se transmiten todos los elementos de sintaxis base_mode_flag y residual_prediction_flag sino que se infiere que son iguales a 1. En la primera capa de mejora de SNR, este indicador debe ajustarse habitualmente a 0, ya que para esta mejora debe ser posible transmitir vectores de movimiento que son diferentes de la capa base con el fin de mejorar la eficacia de la codificación. Sin embargo, en todas las demás capas de mejora, este indicador se ajusta igual a 1, ya que estas capas de mejora sólo representan un refinamiento de niveles de coeficientes de transformada de posiciones de barrido que no se han codificado en las capas de mejora de SNR previas. Asimismo, ajustando este indicador igual a 1, la eficacia de la codificación puede mejorarse para este caso, ya que no es necesaria una transmisión de elementos de sintaxis no requeridos y por tanto se ahorra tasa de transmisión de bits asociada.

Tal como se describió anteriormente, la primera posición de barrido x para los niveles de coeficientes de transformada en los diversos bloques de transformada pueden transmitirse a un nivel de segmento, sin que se transmita ningún elemento de sintaxis a nivel de macrobloque para coeficientes de transformada con una posición de barrido inferior a x. Además de la descripción anterior, en la que la primera posición de barrido se transmite sólo para un tamaño de transformada específico y la primera posición de barrido para otro tamaño de transformada se infiere basándose en el valor transmitido, será posible transmitir una primera posición de barrido para todos los tamaños de transformada soportados.

De manera similar, la última posición de barrido y para los niveles de coeficientes de transformada en los diversos bloques de transformada puede transmitirse a nivel de segmento, sin que se transmita ningún elemento de sintaxis a nivel de macrobloque para coeficientes de transformada con una posición de barrido superior a y. De nuevo, es posible o bien transmitir una última posición de barrido para todos los tamaños de transformada soportados, o bien transmitir la última posición de barrido sólo para un tamaño de transformada específico e inferir la última posición de barrido para otros tamaños de transformada basándose en el valor transmitido.

La primera posición de barrido para cada bloque de transformada en una capa de mejora de

SNR puede inferirse, alternativamente, basándose en los coeficientes de transformada que se han transmitido en una capa de mejora previa. Esta regla de inferencia puede aplicarse de manera independiente a todos los bloques de transformada, y en cada bloque puede derivarse un primer coeficiente de transformada diferente mediante, por ejemplo, el combinador 428.

Además, puede realizarse una combinación de señalización e inferencia de la primera posición de barrido. Esto significa que la primera posición de barrido puede inferirse básicamente basándose en niveles de coeficientes de transformada ya transmitidos en capas de mejora de SNR previas, pero para esto se usa el conocimiento adicional de que la primera posición de barrido no puede ser inferior a un valor x, que se transmite en la cabecera de segmento. Con este concepto es posible, de nuevo, tener un primer índice de barrido diferente en cada bloque de transformada, que puede elegirse con el fin de maximizar la eficacia de la codificación.

Como otra alternativa adicional, la señalización de la primera posición de barrido, la inferencia de la primera posición de barrido, o la combinación de éstas, pueden combinarse con la señalización de la última posición de barrido.

A este respecto, la descripción anterior posibilita un posible esquema que permite la ajustabilidad a escala de SNR, en el que sólo se transmiten subconjuntos de niveles de coeficientes de transformada en diferentes capas de mejora de SNR, y este modo se señala mediante uno o más elementos de sintaxis en la cabecera de segmento, que especifican que los modos de macrobloque y los parámetros de movimiento se infieren para todos los tipos de macrobloque y/o que los coeficientes de transformada para varias posiciones de barrido no están presentes a nivel de bloque de transformada. Puede usarse un elemento de sintaxis a nivel de segmento que señala que los modos de macrobloque y los parámetros de movimiento para todos los macrobloques se infieren a partir de los macrobloques de capa base coubicados. Específicamente, pueden usarse los mismos modos de macrobloque y parámetros de movimiento, y los correspondientes elementos de sintaxis pueden no transmitirse a nivel de segmento. La primera posición de barrido x para todos los bloques de transformada puede señalizarse mediante elementos de sintaxis de cabecera de segmento. A nivel de macrobloque, no se transmiten elementos de sintaxis para valores de coeficiente de transformada de posiciones de barrido inferiores a x. Alternativamente, la primera posición de barrido para un bloque de transformada puede inferirse basándose en los niveles de coeficientes de transformada transmitidos de la capa base. También es posible una combinación de estas últimas alternativas. De manera similar, la última posición de barrido y para todos los bloques de transformada puede señalizarse mediante elementos de sintaxis de cabecera de segmento, donde, a nivel de macrobloque, no se transmiten elementos de sintaxis para valores de coeficiente de transformada de posiciones de barrido superiores a y.

Tal como se indicó anteriormente, las realizaciones descritas de manera detallada de las figuras 1-11 pueden variarse de diferentes maneras. Por ejemplo, aunque las realizaciones anteriores se han ejemplificado con respecto a dos entornos de capa espacial, las realizaciones anteriores pueden transponerse fácilmente a una realización con sólo una capa de calidad o con más de una capa de calidad pero con las N+1 capas de refinamiento ajustables a escala de SNR. Supóngase, por ejemplo, que falta la parte 12 en la figura 1. En este caso, el codificador 42 híbrido actúa como medio de codificación para codificar la señal 18 de vídeo usando transformación por bloques para obtener bloques 146, 148 de transformada de valores de coeficiente de transformación para una imagen 140 de la señal de vídeo, mientras que la unidad 44 actúa como medio de formación, para cada una de una pluralidad de capas de calidad, conteniendo un subflujo 30 o 28 más 30 de datos de vídeo información de alcance de barrido que indica un subconjunto de las posibles posiciones de barrido, e información de coeficiente de transformada sobre valores de coeficiente de transformación pertenecientes al subconjunto de posibles posiciones de barrido. No estará implicada una predicción entre capas. Además, el codificador 42 puede simplificarse para no realizar predicción de movimiento sino meramente transformación por bloques. De manera similar, en el caso de una capa de calidad, el demultiplexor 402 actuará como medio de análisis sintáctico para analizar sintácticamente los subflujos de datos de vídeo de la pluralidad de capas de calidad, para obtener, para cada capa de calidad, la información de alcance de barrido y la información de coeficiente de transformada, y el combinador 428 actuará como medio para construir, usando la información de alcance de barrido, para cada capa de calidad, los bloques de transformada asociando los valores de coeficiente de transformación de los respectivos bloques de transformada a partir de la información de coeficiente de transformada con el subconjunto de posibles posiciones de barrido, reconstruyendo la unidad 438 de transformación hacia atrás la imagen de la señal de vídeo mediante una transformación hacia atrás de los bloques de transformada.

Además, la realización en la figura 1 puede variarse de manera que el codificador 12 de capa base opere con la misma resolución espacial y la misma profundidad de bit que el codificador 14 de capa de mejora. En ese caso, la realización representa codificación ajustable a escala de SNR con una capa 26 base estándar y varias capas 28, 30 de mejora que contienen divisiones de los coeficientes de transformada.

En otras palabras, según una realización de la presente invención, un aparato para reconstruir una señal de vídeo a partir de un flujo de datos de vídeo con calidad ajustable a escala que comprende, para cada una de una pluralidad de capas de calidad, un subflujo de datos de vídeo, comprende medios para analizar sintácticamente los subflujos de datos de vídeo de la pluralidad de capas de calidad, para obtener, para cada capa de calidad, una información de alcance de barrido e información de coeficiente de transformada sobre valores de coeficiente de transformación dispuestos bidimensionalmente de diferentes bloques de transformada, en el que un orden de barrido predeterminado con posibles posiciones de barrido ordena los valores de coeficiente de transformación en una secuencia lineal de valores de coeficiente de transformación, y la información de alcance de barrido indica un subconjunto de las posibles posiciones de barrido; medios para construir, usando la información de alcance de barrido, para cada capa de calidad, los bloques de transformada asociando los valores de coeficiente de transformación de los respectivos bloques de transformada a partir de la información de coeficiente de transformada con el subconjunto de posibles posiciones de barrido; y medios para reconstruir una imagen de la señal de vídeo mediante transformación hacia atrás de los bloques de transformada, en el que los medios para analizar sintácticamente están configurados de tal manera que la información de coeficiente de transformada de más de una de las capas de calidad puede contener un valor de contribución relativo a un valor de coeficiente de transformación, y en el que los medios de construcción están configurados para derivar el valor para el valor de coeficiente de transformada basándose en una suma de los valores de contribución relativos al valor de coeficiente de transformación. Entre los bloques de transformada, al menos un primer bloque de transformada puede estar compuesto por 4 x 4 valores de coeficiente de transformación y al menos un segundo bloque de transformada está compuesto por 8 x 8 valores de coeficiente de transformación, y en el que los medios para analizar sintácticamente están configurados para esperar que la información de alcance de barrido para cada una de la pluralidad de capas de calidad indica que el subconjunto de posibles posiciones de barrido para el al menos un primer bloque de transformada con una precisión que permita una distinción entre todas las 16 posibles posiciones de barrido del al menos un primer bloque de transformada, y para el al menos un segundo bloque de transformada con una precisión que permita una distinción entre los 16 cuádruplos de coeficientes de transformada consecutivos del al menos un segundo bloque de transformada. Los medios para analizar sintácticamente pueden estar configurados para esperar que los valores de coeficiente de transformación pertenecientes al subconjunto de posibles posiciones de barrido estén codificados por bloques en la información de coeficiente de transformada de modo que los valores de coeficiente de transformación de un bloque de transformada predeterminado formen una parte continua de la información de coeficiente de transformada. Los medios para analizar sintácticamente pueden estar configurados para decodificar la parte consecutiva decodificando un mapa de significancia (significant_coeff_flag, last_significant_coeff_flag) que especifica las posiciones de los valores de coeficiente de transformación distintos de cero y pertenecientes al subconjunto de posibles posiciones de barrido en el bloque de transformada predeterminado en el subflujo de datos de vídeo, y posteriormente, en un orden de barrido inverso invertido con respecto al orden de barrido predeterminado empezando con el último valor de coeficiente de transformación distinto de cero y perteneciente al subconjunto de posibles posiciones de barrido dentro del bloque de transformada predeterminado, decodificar los valores de coeficiente de transformada distintos de cero y pertenecientes al subconjunto de posibles posiciones de barrido dentro del bloque de transformada predeterminado. Los medios para analizar sintácticamente pueden estar configurados para decodificar el mapa de significancia decodificando, en el orden de barrido predeterminado, un indicador de significancia (significant_coeff_flag) por valor de coeficiente de transformación perteneciente al subconjunto de posibles posiciones de barrido a partir del primer valor de coeficiente de transformación perteneciente al subconjunto de posibles posiciones de barrido hasta el último valor de coeficiente de transformada perteneciente al subconjunto de posibles posiciones de barrido y distinto de cero, dependiendo los indicadores de significancia de que el respectivo valor de coeficiente de transformación sea cero o distinto de cero, y, siguiendo a cada indicador de significancia de un respectivo valor de coeficiente de transformación distinto de cero, decodificar un último indicador (last_significant_coeff_flag) dependiendo del respectivo valor de coeficiente de transformación que es el último valor de coeficiente de transformación perteneciente a los subconjuntos de posibles posiciones de barrido dentro del bloque de transformada predeterminado distinto de cero o no. Alternativamente, los medios para analizar sintácticamente pueden estar configurados para decodificar la parte consecutiva decodificando una información de significancia que especifica el número de valores de coeficiente de transformación distintos de cero y pertenecientes al subconjunto de posibles posiciones de barrido dentro del bloque de transformada predeterminado así como el número de valores de cola consecutivos de coeficiente de transformación que tienen un valor absoluto de uno dentro del número de valores de coeficiente de transformación distintos de cero y pertenecientes al subconjunto de posibles posiciones de barrido dentro del bloque de transformada predeterminado; decodificar los signos de los valores de cola consecutivos de coeficiente de transformación y los valores de coeficiente de transformación restantes distintos de cero y pertenecientes al subconjunto de posibles posiciones de barrido dentro del bloque de transformada predeterminado; decodificar el número total de valores de coeficiente de transformación iguales a cero y pertenecientes al subconjunto de posibles posiciones de barrido hasta el último valor de coeficiente de transformación distinto de cero y perteneciente al subconjunto de posibles posiciones de

barrido dentro del bloque de transformada predeterminado; decodificar el número de valores de coeficiente de transformación consecutivos iguales a cero y pertenecientes al subconjunto de posibles posiciones de barrido inmediatamente precedente a cualquiera del número de valores de coeficiente de transformación distintos de cero y pertenecientes al subconjunto de posibles posiciones de barrido dentro del bloque de transformada predeterminado en un orden de barrido invertido. Además, el orden de barrido predeterminado puede barrer los valores de coeficiente de transformación de los bloques de transformada de tal manera que los valores de coeficiente de transformación pertenecientes a una posición de barrido superior en el orden de barrido predeterminado se refieran a frecuencias espaciales más altas.

Dependiendo de una implementación real, el esquema de la invención puede implementarse en hardware o en software. Por tanto, la presente invención también se refiere a un programa informático, que puede almacenarse en un medio legible por ordenador tal como un CD, un disquete o cualquier otro soporte de datos. La presente invención también es, por tanto, un programa informático que tiene un código de programa que, cuando se ejecuta en un ordenador, realiza el método de la invención en conexión con las figuras anteriores.

Además, ha de indicarse que todas las etapas o funciones indicadas en los diagramas de flujo pueden implementarse mediante respectivos medios en el codificador y que las implementaciones pueden comprender subrutinas ejecutadas en una CPU, partes de circuito de un ASIC o similares.

REIVINDICACIONES

1. Aparato para generar un flujo (36) de datos de vídeo con calidad ajustable a escala, en el que una secuencia unidimensional de valores de coeficiente de transformada se dispone en una pluralidad de capas de calidad, que comprende:

medios (42) para codificar una señal (18) de vídeo usando transformación por bloques para obtener bloques (146, 148) de transformada de valores de coeficiente de transformada dispuestos bidimensionalmente para una imagen (140) de la señal de vídeo, en el que un orden (154, 156, 164, 166) de barrido predeterminado con posibles posiciones de barrido ordena los valores de coeficiente de transformada en dicha secuencia unidimensional de valores de coeficiente de transformada; y

medios (44) para formar, para cada una de la pluralidad de capas de calidad, un subflujo (30; 28, 30) de datos de vídeo que contiene información de alcance de barrido que indica un subconjunto de las posibles posiciones de barrido, de tal manera que el subconjunto de cada una de la pluralidad de capas de calidad comprende al menos una posible posición de barrido no incluida en el subconjunto de cualquier otra de la pluralidad de capas de calidad y una de las posibles posiciones de barrido está incluida en más de uno de los subconjuntos de capas de calidad, e información de coeficiente de transformada sobre valores de coeficiente de transformada pertenecientes al subconjunto de posibles posiciones de barrido de la respectiva capa de calidad, que contiene un valor de contribución por cada posible posición de barrido del subconjunto de posibles posiciones de barrido de la respectiva capa de calidad, de tal manera que el valor de coeficiente de transformada de la posible posición de barrido puede derivarse basándose en una suma de los valores de contribución para la posible posición de barrido de los más de un subconjuntos de las capas de calidad.

2. Aparato para reconstruir una señal de vídeo a partir de un flujo (36) de datos de vídeo con calidad ajustable a escala, en el que una secuencia unidimensional de valores de coeficiente de transformada se dispone en una pluralidad de capas de calidad, y que comprende, para cada una de la pluralidad de capas de calidad, un subflujo (30; 28, 30) de datos de vídeo, que comprende:

medios (402) para analizar sintácticamente los subflujos de datos de vídeo de la pluralidad de capas de calidad, para obtener, para cada capa de calidad, una información de alcance de barrido e información de coeficiente de transformada sobre valores de coeficiente de transformada dispuestos bidimensionalmente de diferentes bloques de transformada, en el que un orden de barrido predeterminado con posibles posiciones de barrido ordena los valores de coeficiente de transformada en dicha secuencia unidimensional de valores de coeficiente de transformada, y la información de alcance de barrido indica un subconjunto de las posibles posiciones de barrido;

medios (428) para, usando la información de alcance de barrido, para cada capa de calidad, construir los bloques de transformada asociando los valores de coeficiente de transformada de los respectivos bloques de transformada a partir de la información de coeficiente de transformada al subconjunto de posibles posiciones de barrido; y

medios (438) para reconstruir una imagen de la señal de vídeo mediante una transformación hacia atrás de los bloques de transformada, en el que los medios para analizar sintácticamente están configurados de tal manera que la información de coeficiente de transformada de más de una de las capas de calidad contiene un valor de contribución relativo a un valor de coeficiente de transformada, y en el que los medios de construcción están configurados para derivar el valor para el valor de coeficiente de transformada basándose en una suma de los valores de contribución relativos al valor de coeficiente de transformada.

3. Aparato según la reivindicación 2, en el que los medios (428) de construcción están configurados para usar la información de alcance de barrido como si indicara una primera posición de barrido de entre las posibles posiciones de barrido dentro del subconjunto de posibles posiciones de barrido en el orden de barrido predeterminado.

4. Aparato según las reivindicaciones 2 ó 3, en el que los medios (428) de construcción están configurados para usar la información de alcance de barrido como si indicara una última posición de barrido de entre las posibles posiciones de barrido dentro del subconjunto de posibles posiciones de barrido en el orden de barrido predeterminado.

5. Aparato según cualquiera de las reivindicaciones 2 a 4, en el que los medios de reconstrucción están configurados para reconstruir la señal de vídeo usando predicción con compensación de movimiento basándose en información de movimiento y combinando un resultado de predicción con compensación de movimiento con un residuo de predicción con compensación de movimiento, obteniéndose el residuo de predicción con compensación de movimiento mediante una transformación inversa por bloques de los bloques de transformada de valores de coeficiente de transformada.

6. Aparato según la reivindicación 5, en el que los medios (402) para analizar sintácticamente están configurados para esperar que cada subflujo de datos contenga una indicación que indica la existencia de información de movimiento o la no existencia de información de movimiento para la respectiva capa de calidad y que el subflujo de datos de una primera capa de las capas de calidad contenga la información de movimiento y tenga la indicación que indica la existencia de información de movimiento, o la indicación dentro del subflujo de datos de la primera capa de calidad indica la no existencia de información de movimiento, comprendiendo una parte del flujo de datos de vídeo con calidad ajustable a escala diferente de los subflujos de datos la información de movimiento, y para esperar que el (los) subflujo(s) de datos de la(s) otra(s) capa(s) de calidad tenga(n) la indicación que indica la no existencia de información de movimiento.

7. Aparato según la reivindicación 6, en el que los medios para analizar sintácticamente están configurados para esperar que el subflujo de datos de la primera capa de calidad tenga la indicación que indica la existencia de información de movimiento, siendo la información de movimiento igual a la información de movimiento de calidad más alta o igual a una información de refinamiento que permite una reconstrucción de la información de movimiento de calidad más alta basándose en la información de movimiento de calidad más baja, y que la parte del flujo de datos de vídeo con calidad ajustable a escala también contenga la información de movimiento de calidad más baja.

8. Aparato según las reivindicaciones 6 ó 7, en el que los medios para analizar sintácticamente están configurados de tal manera que la información de movimiento y la indicación se refieren a un macrobloque de la imagen.

9. Aparato según cualquiera de las reivindicaciones 2 a 8, en el que los medios para analizar sintácticamente están configurados para analizar sintácticamente cada subflujo de datos individualmente de manera independiente, con respeto a un resultado del análisis sintáctico, del (de los) otro(s) subflujo(s) de datos.

10. Aparato según la reivindicación 9, en el que los medios de construcción están configurados para asociar la respectiva información de coeficiente de transformada con los valores de coeficiente de transformada, siendo el resultado de la asociación independiente del (de los) otro(s) subflujo(s) de datos.

11. Aparato según la reivindicación 10, en el que se define un orden de capas entre las capas de calidad, y el subflujo de datos de una primera capa de calidad en el orden de capas permite una asociación de la respectiva información de coeficiente de transformada con los valores de coeficiente de transformada independiente del (de los) subflujo(s) de datos de la(s) siguiente(s) capa(s) de calidad, mientras que el (los) subflujo(s) de datos de las siguientes capas de calidad en el orden de capas permite(n) una asociación de la respectiva información de coeficiente de transformada con los valores de coeficiente de transformada meramente en combinación con el (los) subflujo(s) de datos de (una) capa(s) de calidad que precede(n) a la respectiva capa de calidad, en el que los medios de construcción están configurados para asociar la información de coeficiente de transformada de una respectiva capa de calidad con los valores de coeficiente de transformada usando los subflujos de datos de la respectiva capa de calidad y la(s) capa(s) de calidad que precede(n) a la respectiva capa de calidad.

12. Método para generar un flujo (36) de datos de vídeo con calidad ajustable a escala, en el que una secuencia unidimensional de valores de coeficiente de transformada se dispone en una pluralidad de capas de calidad, que comprende:

codificar una señal (18) de vídeo usando transformación por bloques para obtener bloques (146, 148) de transformada de valores de coeficiente de transformada dispuestos bidimensionalmente para una imagen (140) de la señal de vídeo, en el que un orden (154, 156, 164, 166) de barrido predeterminado con posibles posiciones de barrido ordena los valores de coeficiente de transformada en dicha secuencia unidimensional de valores de coeficiente de transformada; y

formar, para cada una de la pluralidad de capas de calidad, un subflujo (30; 28, 30) de datos de vídeo que contiene información de alcance de barrido que indica un subconjunto de las posibles posiciones de barrido, de tal manera que el subconjunto de cada una de la pluralidad de capas de calidad comprende al menos una posible posición de barrido no incluida en el subconjunto de cualquier otra de la pluralidad de capas de calidad y una de las posibles posiciones de barrido está incluida en más de uno de los subconjuntos de las capas de calidad, e información de coeficiente de transformada sobre valores de coeficiente de transformada pertenecientes al subconjunto de posibles posiciones de barrido de la respectiva capa de calidad, que contiene un valor de contribución por cada posible posición de barrido del subconjunto de posibles posiciones de barrido de la respectiva capa de calidad de tal manera que el valor de coeficiente de transformada de la posible posición de barrido puede derivarse basándose en una suma de los valores de contribución para la posible posición de barrido de los más de un subconjuntos de las capas de calidad.

13. Método para reconstruir una señal de vídeo a partir de un flujo (36) de datos de vídeo con calidad ajustable a escala, en el que una secuencia unidimensional de valores de coeficiente de transformada se dispone en una pluralidad de capas de calidad, y que comprende, para cada una de la pluralidad de capas de calidad, un subflujo (30; 28, 30) de datos de vídeo, que comprende:

analizar sintácticamente los subflujos de datos de vídeo de la pluralidad de capas de calidad, para obtener, para cada capa de calidad, una información de alcance de barrido e información de coeficiente de transformada sobre valores de coeficiente de transformada dispuestos bidimensionalmente de diferentes bloques de transformada, en el que un orden de barrido predeterminado con posibles posiciones de barrido ordena los valores de coeficiente de transformada en dicha secuencia unidimensional de valores de coeficiente de transformada, y la información de alcance de barrido indica un subconjunto de las posibles posiciones de barrido;

usando la información de alcance de barrido, para cada capa de calidad, construir los bloques de transformada asociando los valores de coeficiente de transformada de los respectivos bloques de transformada a partir de la información de coeficiente de transformada al subconjunto de posibles posiciones de barrido; y

reconstruir una imagen de la señal de vídeo mediante una transformación hacia atrás de los bloques de transformada, en el que el análisis sintáctico de los subflujos de datos de vídeo se realiza de tal manera que la información de coeficiente de transformada de más de una de las capas de calidad contiene un valor de contribución relativo a un valor de coeficiente de transformada, y la construcción de los bloques de transformada comprende derivar el valor para el valor de coeficiente de transformada basándose en una suma de los valores de contribución relativos al valor de coeficiente de transformada.

14. Flujo de datos de vídeo con calidad ajustable a escala en el que una secuencia unidimensional de valores de coeficiente de transformada se dispone en una pluralidad de capas de calidad, y que permite una reconstrucción de una señal de vídeo que comprende, para cada una de la pluralidad de capas de calidad, una información de alcance de barrido e información de coeficiente de transformada sobre valores de coeficiente de transformada dispuestos bidimensionalmente de diferentes bloques de transformada, en el que un orden de barrido predeterminado con posibles posiciones de barrido ordena los valores de coeficiente de transformada en dicha secuencia unidimensional de valores de coeficiente de transformada, y la información de alcance de barrido indica un subconjunto de las posibles posiciones de barrido, de tal manera que el subconjunto de cada una de la pluralidad de capas de calidad comprende al menos una posible posición de barrido no incluida en el subconjunto de cualquier otra de la pluralidad de capas de calidad, en el que la información de coeficiente de transformada se refiere a valores de coeficiente de transformada pertenecientes al subconjunto de posibles posiciones de barrido, en el que la información de coeficiente de transformada de más de una de las capas de calidad contiene un valor de contribución relativo a un valor de coeficiente de transformada, y el valor de coeficiente de transformada de la posible posición de barrido puede derivarse basándose en una suma de los valores de contribución para la posible posición de barrido de los más de un subconjuntos de las capas de calidad.

15. Programa informático que tiene un código de programa para realizar, cuando se ejecuta en un ordenador, un método según la reivindicación 12 ó 13.

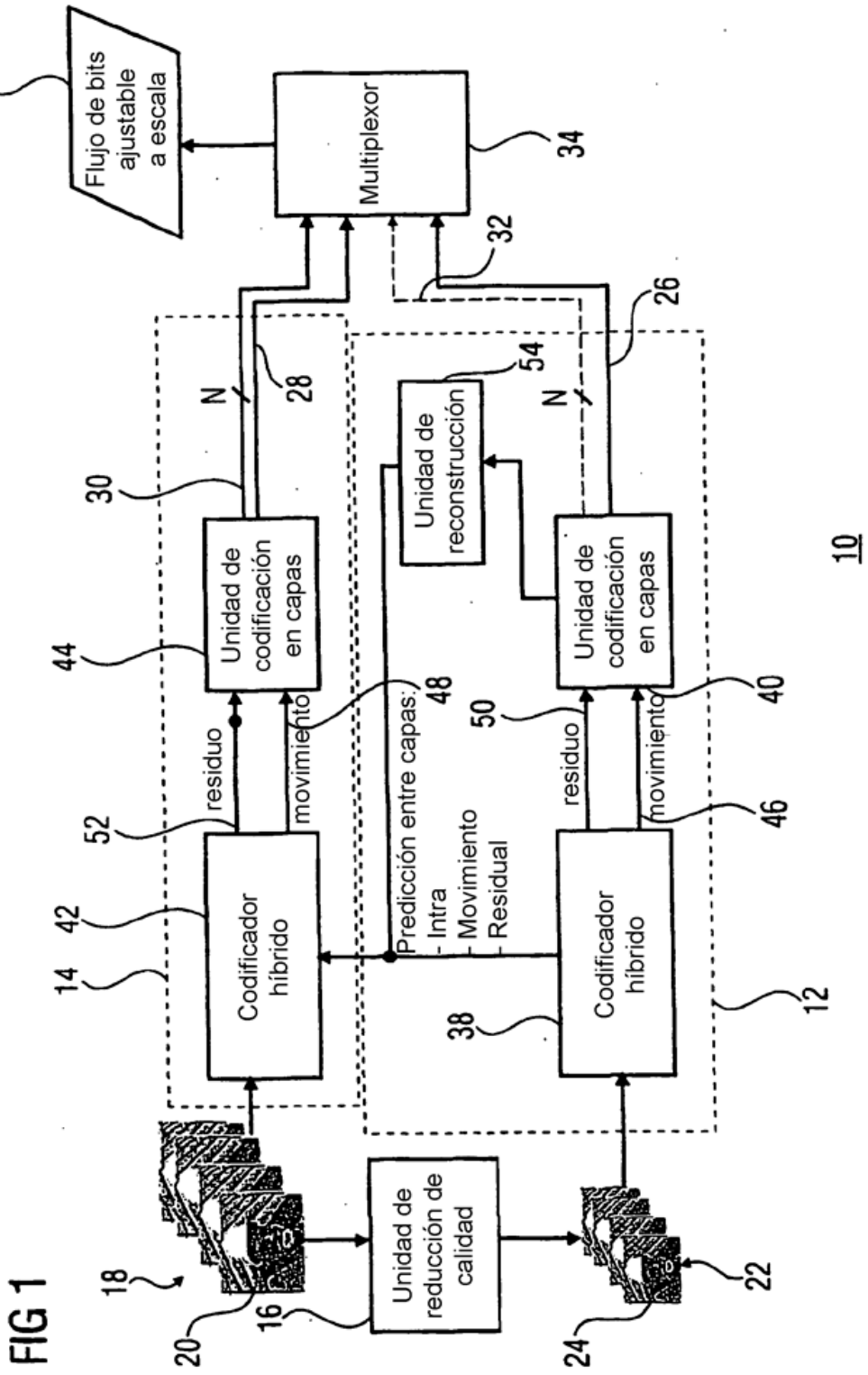

Flujo de bits ajustable a escala
Multiplexor
Unidad de codificación en capas
Unidad de reconstrucción
Unidad de codificación en capas
Codificador híbrido
Codificador híbrido
Unidad de reducción de calidad
Predicción entre capas:
- Intra
- Movimiento
- Residual
residuo
movimiento
residuo
movimiento
N
N
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54

FIG 1

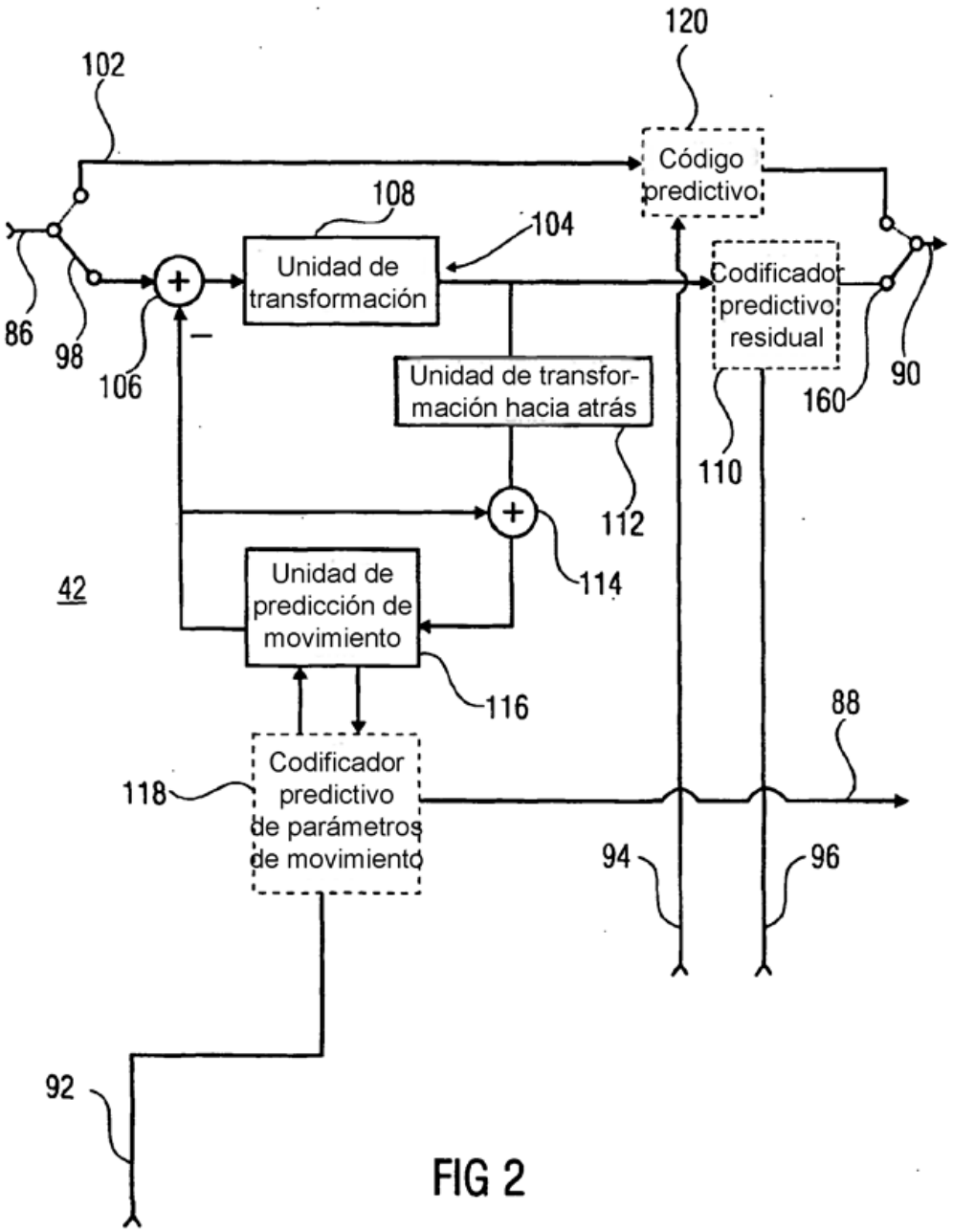

120
102
Código predictivo
108
104
Unidad de transformación
Codificador predictivo residual
86
98
106
90
160
Unidad de transformación hacia atrás
110
112
114
42
Unidad de predicción de movimiento
116
88
118
Codificador predictivo de parámetros de movimiento
94
96
92

FIG 2

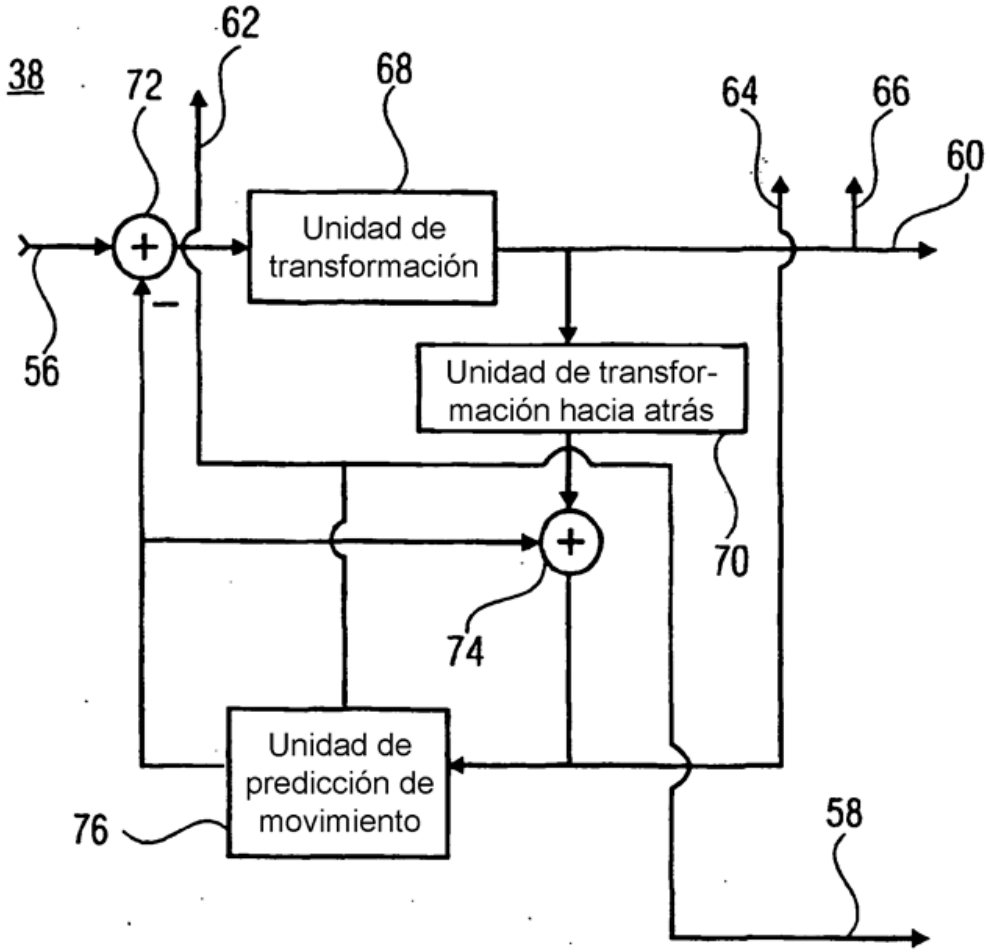

38
72
62
68
64
66
60
56
Unidad de transformación
Unidad de transformación hacia atrás
70
74
Unidad de predicción de movimiento
76
58

FIG 3

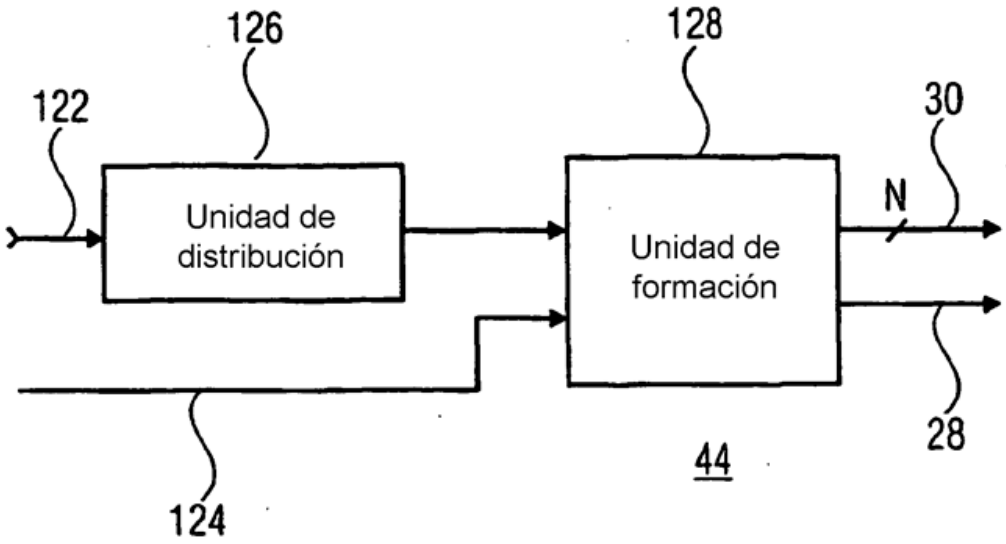

126
128
122
30
Unidad de distribución
Unidad de formación
N
28
44
124

FIG 4

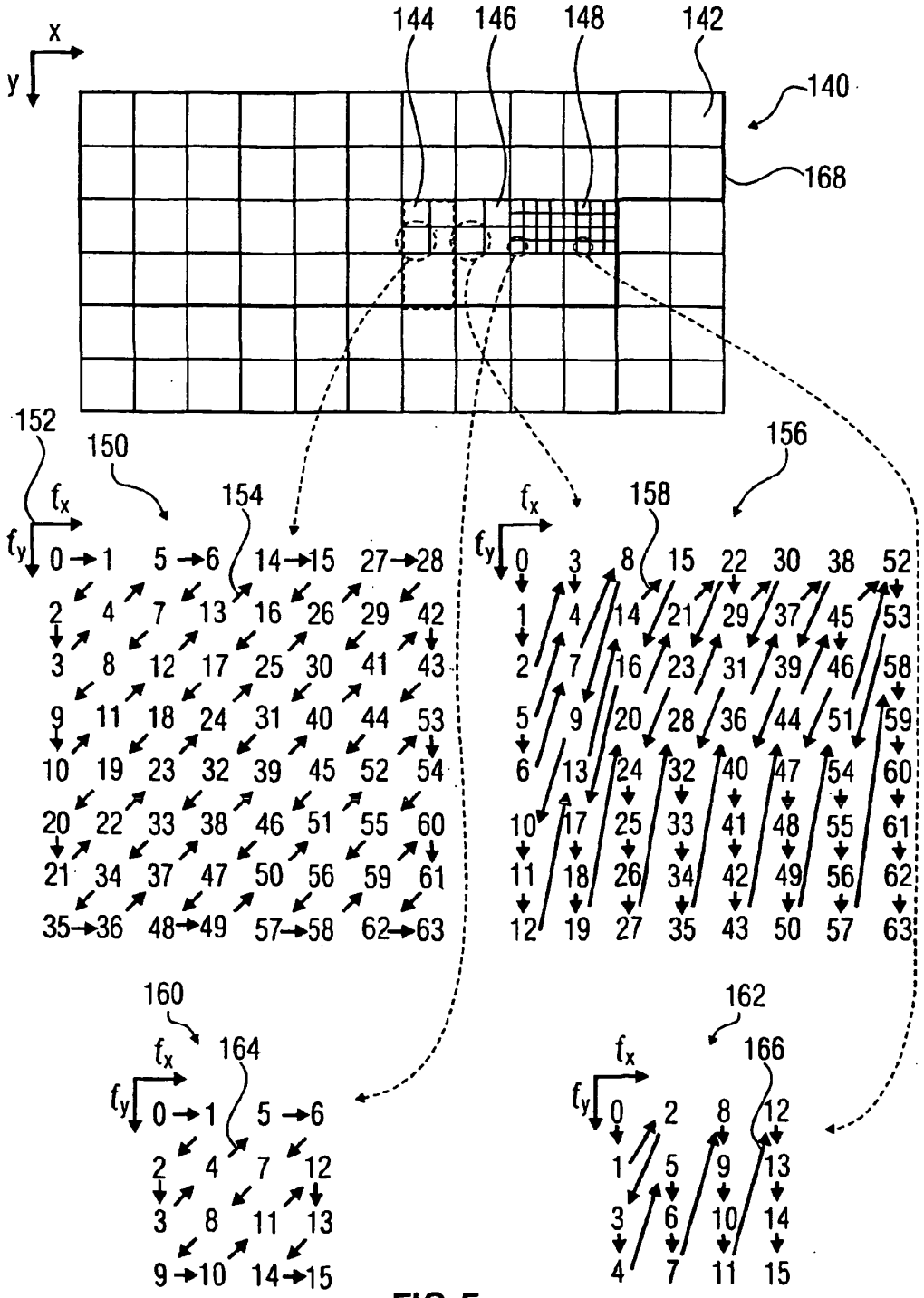

x
y
144
146
148
142
140
168
152
150
154
156
158
160
162
164
166
f_x
f_y
0→1 5→6 14→15 27→28
2 4 7 13 16 26 29 42
3 8 12 17 25 30 41 43
9 11 18 24 31 40 44 53
10 19 23 32 39 45 52 54
20 22 33 38 46 51 55 60
21 34 37 47 50 56 59 61
35→36 48→49 57→58 62→63
0 3 8 15 22 30 38 52
1 4 14 21 29 37 45 53
2 7 16 23 31 39 46 58
5 9 20 28 36 44 51 59
6 13 24 32 40 47 54 60
10 17 25 33 41 48 55 61
11 18 26 34 42 49 56 62
12 19 27 35 43 50 57 63
0→1 5→6
2 4 7 12
3 8 11 13
9→10 14→15
0 2 8 12
1 5 9 13
3 6 10 14
4 7 11 15

FIG 5

FIG 6A

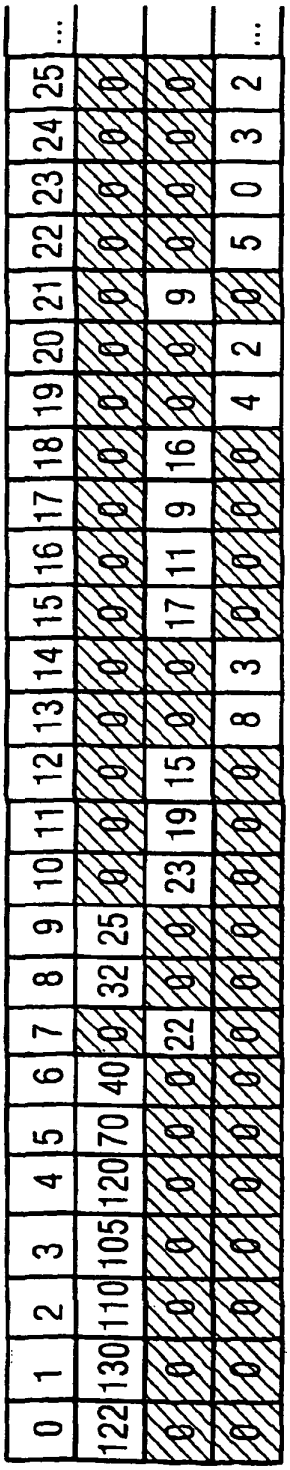


FIG 6E

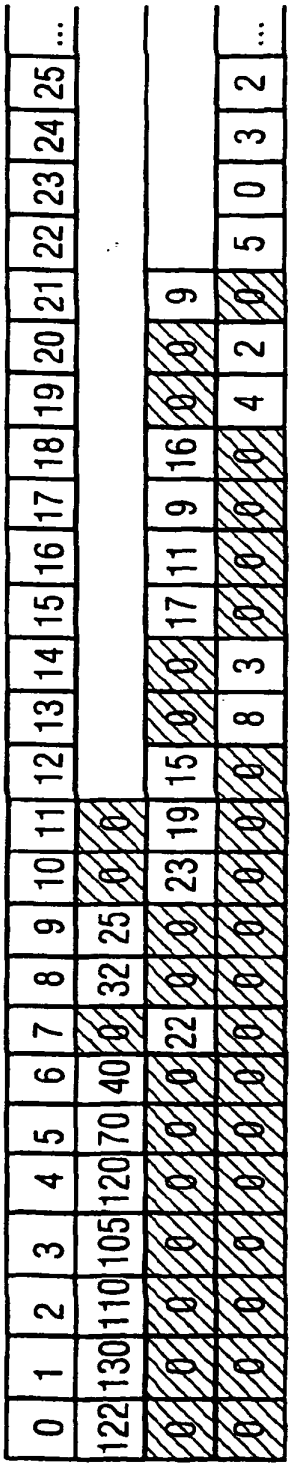


FIG 6D

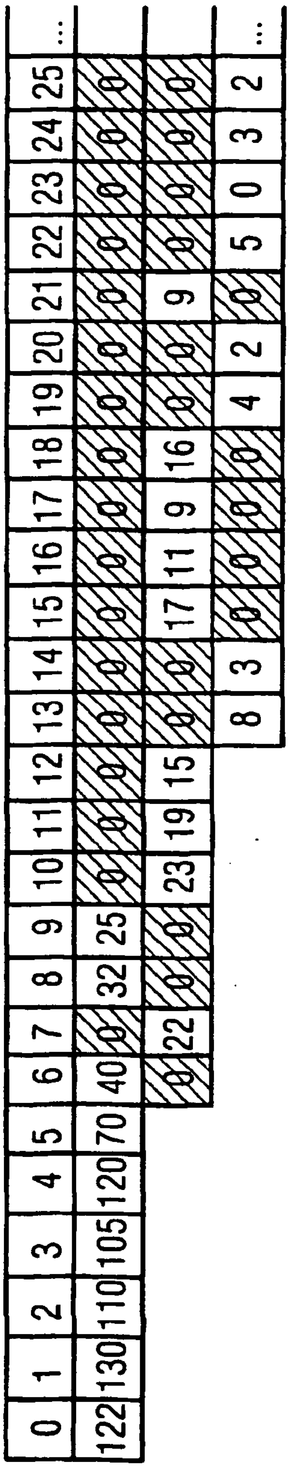

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 ...
122 130 110 105 120 70 40 0 32 25 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 22 0 0 23 19 15 0 0 17 11 9 16 0 0 9 0 0 0 0
8 3 0 0 0 0 4 2 0 5 0 3 2 ...

FIG 6C

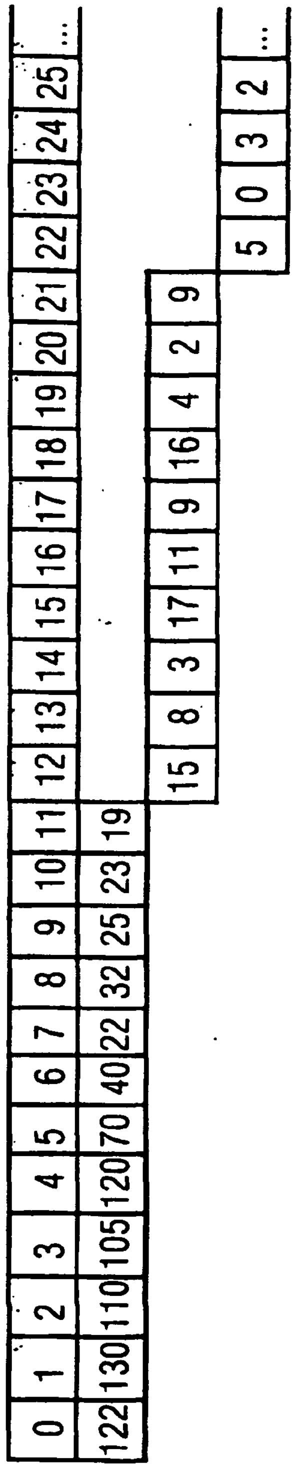

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 ...
122 130 110 105 120 70 40 22 32 25 23 19
15 8 3 17 11 9 16 4 2 9
5 0 3 2 ...

FIG 6B

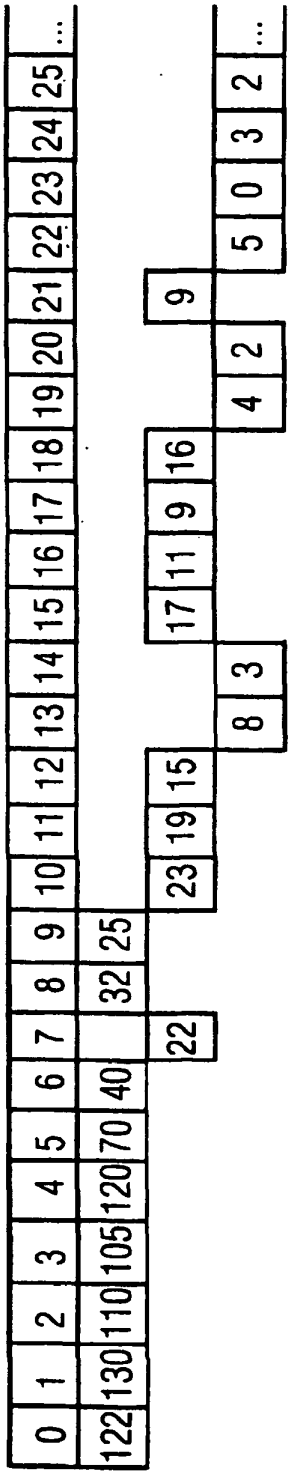

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 ...
122 130 110 105 120 70 40 22 32 25 23 19 15 8 3 17 11 9 16 4 2 9 5 0 3 2 ...

FIG 6F

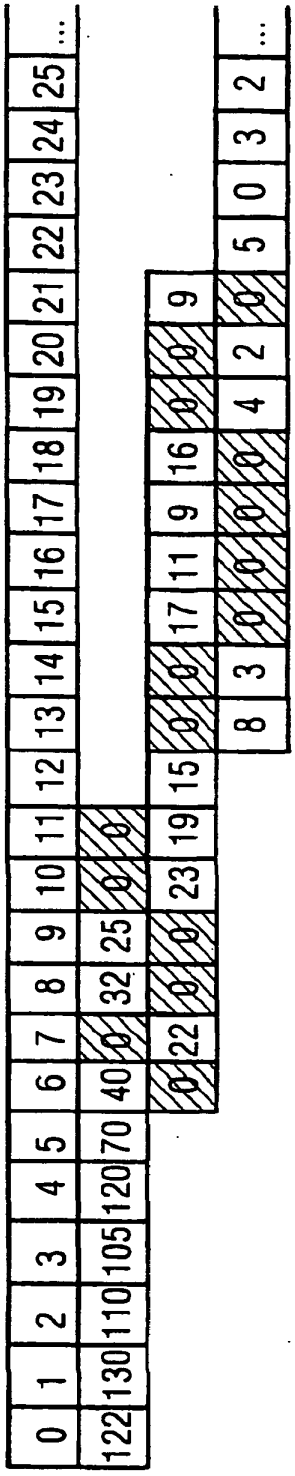
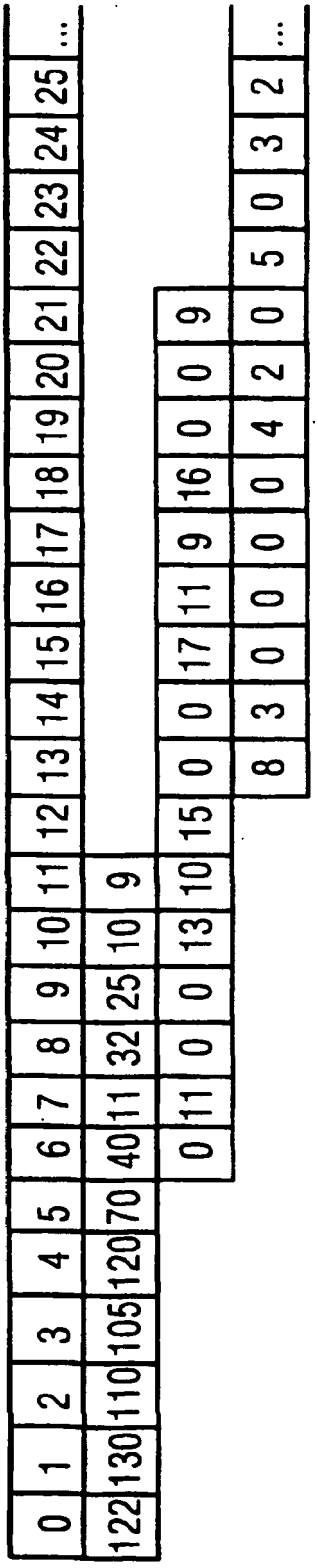

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 ...
122 130 110 105 120 70 40 0 32 25 0 0 15 0 8 3 0 17 11 9 16 0 4 2 0 9 5 0 3 2 ...

FIG 6G


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 ...
122 130 110 105 120 70 40 11 32 25 10 9
0 11 0 0 13 10 15 0 0 17 11 9 16 0 0 9
8 3 0 0 0 0 4 2 0 5 0 3 2 ...

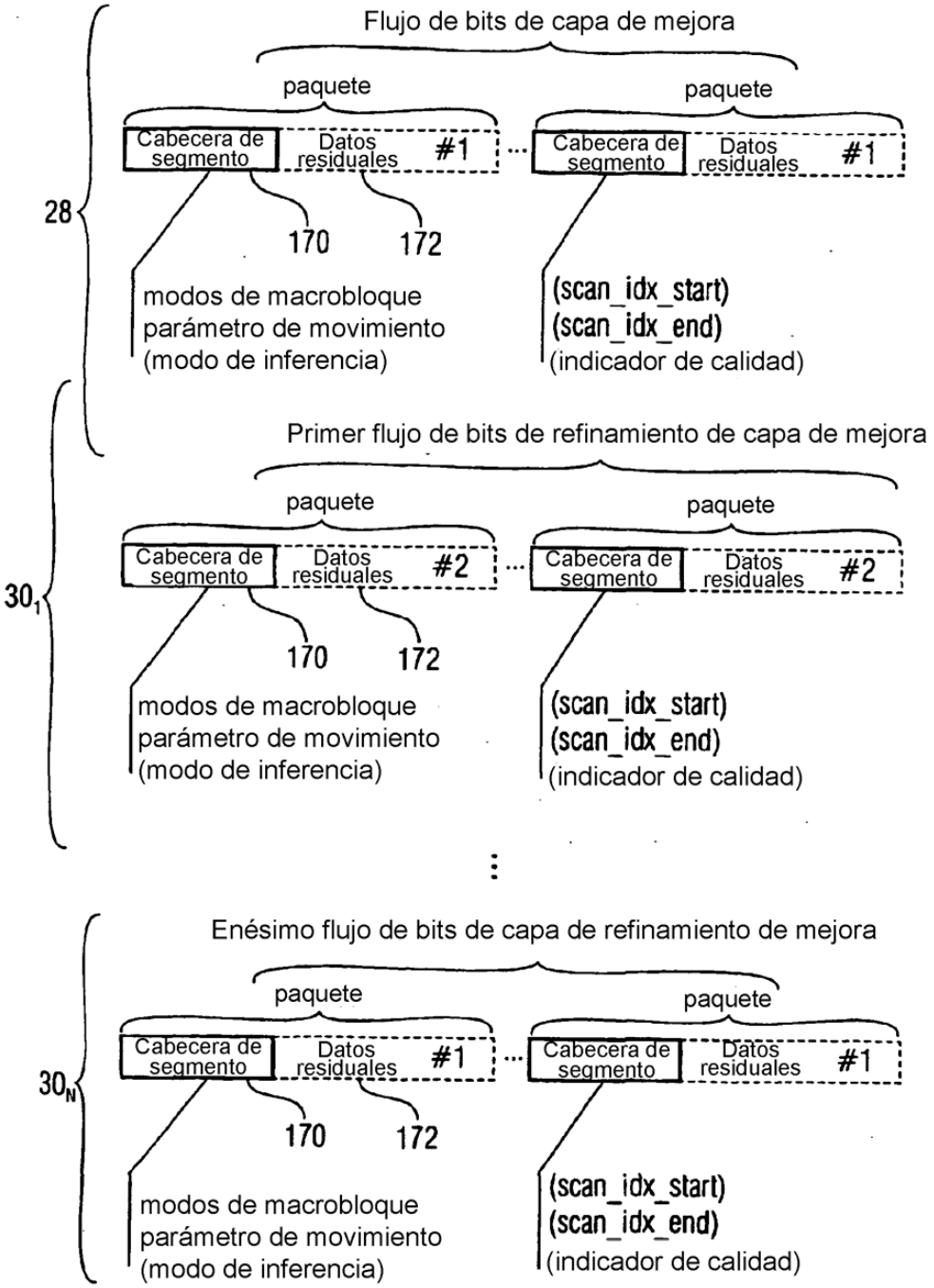

Flujo de bits de capa de mejora
paquete
paquete
Cabecera de segmento
Datos residuales
#1
Cabecera de segmento
Datos residuales
#1
28
170
172
modos de macrobloque
parámetro de movimiento
(modo de inferencia)
(scan_idx_start)
(scan_idx_end)
(indicador de calidad)
Primer flujo de bits de refinamiento de capa de mejora
paquete
paquete
Cabecera de segmento
Datos residuales
#2
Cabecera de segmento
Datos residuales
#2
30₁
170
172
modos de macrobloque
parámetro de movimiento
(modo de inferencia)
(scan_idx_start)
(scan_idx_end)
(indicador de calidad)
Enésimo flujo de bits de capa de refinamiento de mejora
paquete
paquete
Cabecera de segmento
Datos residuales
#1
Cabecera de segmento
Datos residuales
#1
30N
170
172
modos de macrobloque
parámetro de movimiento
(modo de inferencia)
(scan_idx_start)
(scan_idx_end)
(indicador de calidad)

FIG 7

```
residual_block_cavlc( coeffLevel, startIdx, endIdx, maxNumCoeff) {
  for(i=0; i<maxNumCoeff; i++)                                    }
    coeffLevel[ i ]                                               } 280
  coeff_token                                          ←— 240
  if( total_coeff( coeff_token) >0){                   ←— 242
    for( i=0;i<total_coef(coeff_token); i++)           ←— 244
      if(i<trailing_ones(coeff_token)){                ←— 248
        trailing_ones_sign_flag                        ←— 246
        level[ i ] =1 - 2 * trailing_onessign_flag     ←— 250
      } else {
        coeff_level                                    ←— 252
        level [ i ]=coeff_level                        ←— 254
      }
    if( total_coeff( coeff_token )<endIdx-startIdx+1){ ←— 256
      total_zeros                                      ←— 258
      zerosLeft=total_zeros                            ←— 260
    } else
      zerosLeft=0
    for(i=0;i<total_coeff(coeff_token)-1;i++){         ←— 262
      if( zerosLeft>0){                                ←— 270
        run_before                                     ←— 264
        run[ i ]=run_before                            ←— 266
      }else
        run[ i ]= 0                                    ←— 272
      zerroLeft=zerosLeft-run[ i ]                     ←— 268
    }
    run[total_coeff(coeff_token)-1]=zerosLeft          ←— 274
    coeffNum=-1                                        ←— 278
    for(i=total_coeff(coeff_token)-1;i>=0;i--) {                  }
      coeffNum+=run[i] + 1                                        } 276
      coeffLevel[startIdx+coeffNum]=level[i]                      }
    }
  }
}
```

FIG 8

```
residual_block_cabac( coeffLevel, startIdx, endIdx, maxNumCoeff) {
  if( maxNumCoeff==64)
    coded_block_flag=1
  else
    coded_block_flag
  for(i=0; i <maxNumCoeff;i++)
    coeffLevel[ i ]=0                                   } 322
  if(coded_block_flag){
    numCoeff=maxNumCoeffendIdx+1
    i=0starrtIdx
    do {
      significant_coeff_flag[ i ]                       <-- 312
      if(significant_coeff_flag[ i ]){                  <-- 314
        last_significant_coeff_flag[ i ]                <-- 316
        if (last_significant_coeff_flag[ i ]){          <-- 318
          numCoeff=i+1                                  <-- 320
        }
      }
      i++                                               <-- 310
    } while( i < numCoeff-1)
    coeff_abs_level_minus1[numCoeff-1]
    coeff_sign_flag[numCoeff-1]                         } 324
    coeffLevel[ numCoeff-1 ]=
      (coeff_abs_level_minus1[numCoeff-1]+1) *          } 326
      (1-2*coeff_sign_flag[numCoeff-1])
    for( i=numCoeff-2;i>=0;i--)                         <-- 332
      if(significant_coeff_flag[ i ]) {                 <-- 330
        coeff_abs_level_minus1[ i ]
        coeff_sign_flag[ i ]
        coeffLevel[ i ]=(coeff_abs_level_minus1[ i ]+1)*  } 328
                        (1-2*coeff_sign_flag[ i ])
      }
    }
  }
```

FIG 9

FIG 10

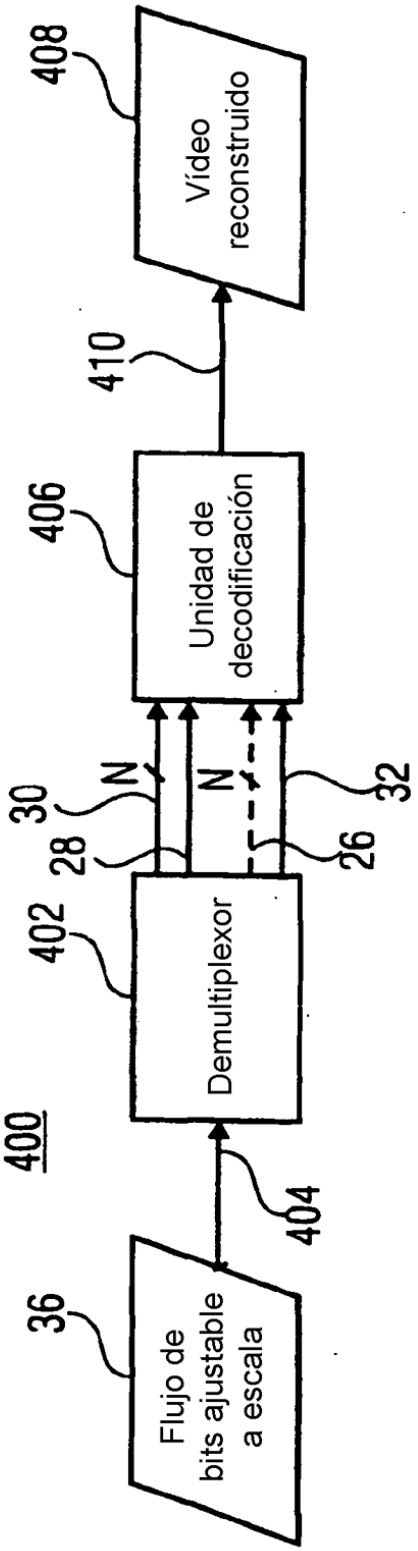
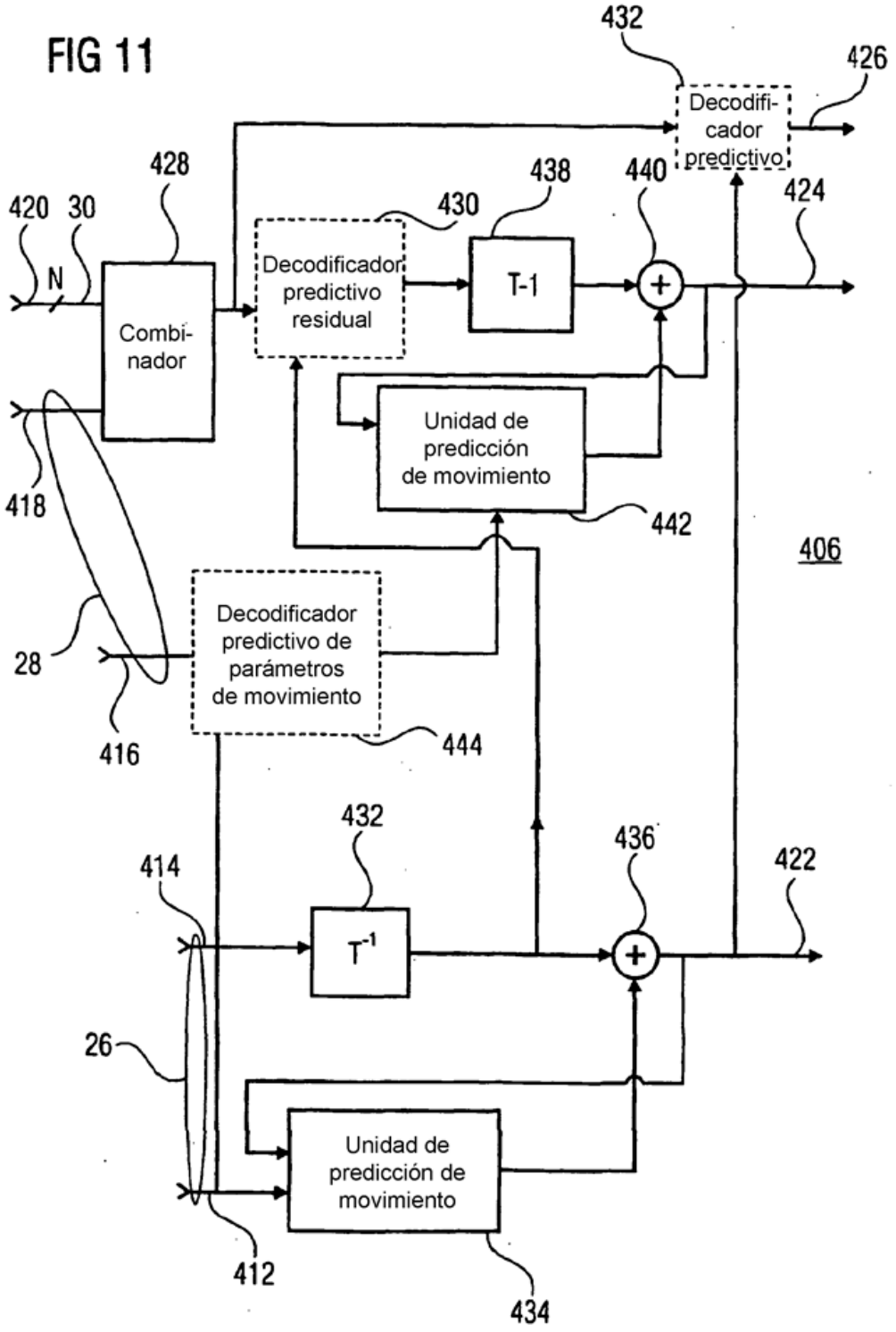

408
Vídeo reconstruido
410
406
Unidad de decodificación
30
N
28
N
26
32
402
Demultiplexor
400
404
36
Flujo de bits ajustable a escala

FIG 11


432
426
Decodificador predictivo
428
438
440
424
420
30
430
N
Decodificador predictivo residual
T-1
Combinador
Unidad de predicción de movimiento
418
442
406
Decodificador predictivo de parámetros de movimiento
28
416
444
432
436
422
414
T-1
26
Unidad de predicción de movimiento
412
434