

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 634 493**

51 Int. Cl.:

H04N 19/577 (2014.01)

H04N 19/70 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **03.09.2004** **E 12158085 (6)**

97 Fecha y número de publicación de la concesión europea: **26.04.2017** **EP 2466895**

54 Título: **Codificación predictiva bidireccional avanzada de vídeo entrelazado**

30 Prioridad:

07.09.2003 US 501081 P
29.06.2004 US 882135

45 Fecha de publicación y mención en BOPI de la
traducción de la patente:
28.09.2017

73 Titular/es:

MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)
One Microsoft Way
Redmond, Washington 98052, US

72 Inventor/es:

HOLCOMB, THOMAS W. y
MUKERJEE, KUNAL

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 634 493 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Codificación predictiva bidireccional avanzada de vídeo entrelazado

Información de la solicitud relacionada

Esta solicitud reivindica el beneficio de la Solicitud de Patente Provisional de Estados Unidos N.º 60/501.081, titulada "Video Encoding and Decoding Tools and Techniques", presentada el 7 de septiembre de 2003, que se incorpora en el presente documento por referencia.

Las siguientes solicitudes de patente de Estados Unidos en trámite junto con la presente se refieren a la presente solicitud y se incorporan en el presente documento por referencia: 1) Solicitud de Patente de Estados Unidos con N.º de Serie 10/622.378, titulada, "Advanced Bi-Directional Predictive Coding of Video Frames", presentada el 18 de julio de 2003; 2) Solicitud de Patente de Estados Unidos con N.º de Serie 10/622.284, titulada, "Intraframe and Interframe Interlace Coding and Decoding", presentada el 18 de julio de 2003; 3) Solicitud de Patente de Estados Unidos con N.º de Serie 10/622.841, titulada, "Coding of Motion Vector Information", presentada el 18 de julio de 2003; y 4) Solicitud de Patente de Estados Unidos con N.º de Serie 10/857.473, titulada, "Predicting Motion Vectors for Fields of Forward-predicted Interlaced Video Frames", presentada el 27 de mayo de 2004.

Campo técnico

Se describen técnicas y herramientas para codificación y decodificación de vídeo entrelazado. Por ejemplo, un codificador de vídeo codifica macrobloques previstos bidireccionalmente en vídeo entrelazado.

Antecedentes

El vídeo digital consume grandes cantidades de capacidad de almacenamiento y transmisión. Una secuencia de vídeo digital sin procesar típica incluye 15 o 30 imágenes por segundo. Cada imagen puede incluir decenas o cientos de miles de píxeles (también denominados *pe/s*). Cada píxel representa un elemento diminuto de la imagen. En forma sin procesar, un ordenador comúnmente representa un píxel con 24 bits o más. Por lo tanto, el número de bits por segundo, o tasa de bits, de una secuencia de vídeo digital sin procesar típica puede ser de 5 millones de bits/segundo o más.

Muchos ordenadores y redes de ordenadores carecen de los recursos para procesar vídeo digital sin procesar. Por esta razón, los ingenieros usan compresión (también denominada codificación o decodificación) para reducir la tasa de bits de vídeo digital. La compresión puede ser sin pérdidas, en la que la calidad del vídeo no sufre pero las disminuciones en la tasa de bits están limitadas por la complejidad del vídeo. O, la compresión puede ser con pérdidas, en la que la calidad del vídeo sufre pero las disminuciones en la tasa de bits son más drásticas. La descompresión invierte la compresión.

En general, las técnicas de compresión de vídeo incluyen "intra" compresión e "inter" compresión o compresión predictiva. Las técnicas de intra compresión comprimen imágenes individuales, típicamente denominadas fotogramas I o fotogramas clave. Las técnicas de compresión comprimen fotogramas con referencia fotogramas precedentes y/o siguientes, y los fotogramas inter-comprimidos se denominan típicamente fotogramas previstos, fotogramas P o fotogramas B.

I. Inter compresión en Windows Media Video, versiones 8 y 9

Windows Media Video de Microsoft Corporation, versión 8 ["WMV8"] incluye un codificador de vídeo y un decodificador de vídeo. El codificador de WMV8 usa intra e inter compresión, y el decodificador de WMV8 usa intra e inter descompresión. Windows Media Video, versión 9 ["WMV9"] usa una arquitectura similar para muchas operaciones.

La inter compresión en el codificador de WMV8 usa codificación de predicción de movimiento compensado basada en bloques seguida por codificación por transformación del error residual. Las Figuras 1 y 2 ilustran la inter compresión basada en bloques para un fotograma previsto en el codificador de WMV8. En particular, la Figura 1 ilustra estimación de movimiento para un fotograma 110 previsto y la Figura 2 ilustra compresión de una predicción residual para un bloque de movimiento compensado de un fotograma previsto.

Por ejemplo, en la Figura 1, el codificador de WMV8 calcula un vector de movimiento para un macrobloque 115 en el fotograma 110 previsto. Para calcular el vector de movimiento, el codificador busca en un área 135 de búsqueda de un fotograma 130 de referencia. Dentro del área 135 de búsqueda, el codificador compara el macrobloque 115 desde el fotograma 110 previsto a diversos macrobloques candidatos para hallar un macrobloque candidato que sea una buena coincidencia. El codificador emite información que especifica el vector de movimiento (codificado por entropía) para el macrobloque coincidente.

Puesto que un valor de vector de movimiento a menudo está correlacionado con los valores de vectores de movimiento espacialmente circundantes, puede conseguirse la compresión de los datos usada para transmitir la información del vector de movimiento seleccionando un predictor de vector de movimiento desde macrobloques

vecinos y predecir el vector de movimiento para el macrobloque actual usando el predictor. El codificador puede codificar el diferencial entre el vector de movimiento y el predictor. Después de reconstruir el vector de movimiento añadiendo el diferencial al predictor, un decodificador usa el vector de movimiento para calcular un macrobloque de predicción para el macrobloque 115 usando información desde el fotograma 130 de referencia, que es un fotograma previamente reconstruido disponible en el codificador y el decodificador. La predicción raramente es perfecta, por lo que el codificador normalmente codifica bloques de diferencias de píxel (también denominado el error o bloques residuales) entre el macrobloque de predicción y el mismo macrobloque 115.

La Figura 2 ilustra un ejemplo de cálculo y codificación de un bloque 235 de error en el codificador WMV8. El bloque 235 de error es la diferencia entre el bloque 215 previsto y el bloque 225 actual original. El codificador aplica una transformada de coseno discreta ["DCT"] 240 al bloque 235 de error, que da como resultado un bloque 245 de 8x8 de coeficientes. El codificador a continuación cuantifica 250 los coeficientes de DCT, dando como resultado un bloque de 8x8 de coeficientes 255 de DCT cuantificados. El codificador explora 260 el bloque 255 de 8x8 en una matriz 265 unidimensional de manera que los coeficientes están generalmente ordenados desde la frecuencia más baja a la frecuencia más alta. El codificador codifica por entropía los coeficientes explorados usando una variación de codificación 270 por longitud de serie. El codificador selecciona un código de entropía desde una o más tablas 275 de serie/nivel/duración y emite el código de entropía.

La Figura 3 muestra un ejemplo de un procedimiento 300 de decodificación correspondiente para un bloque inter-codificado. En el resumen de la Figura 3, un decodificador decodifica (310, 320) información codificada por entropía que representa una predicción residual usando decodificación 310 de longitud variable con una o más tablas 315 de serie/nivel/duración y decodificación 320 por longitud de serie. El decodificador explora 330 a la inversa una matriz 325 unidimensional que almacena la información decodificada por entropía en un bloque 335 bidimensional. El decodificador cuantifica a la inversa y aplica la transformada de coseno discreta inversa (juntas, 340) a los datos, dando como resultado un bloque 345 de error reconstruido. En una trayectoria de compensación de movimiento separada, el decodificador calcula un bloque 365 previsto usando información 355 de vector de movimiento para desplazamiento desde un fotograma de referencia. El decodificador combina 370 el bloque 365 previsto con el bloque 345 de error reconstruido para formar el bloque 375 reconstruido.

La cantidad de cambio entre los fotogramas original y reconstruido es la distorsión y el número de bits requeridos para codificar el fotograma indica la tasa para el fotograma. La cantidad de distorsión es aproximadamente inversamente proporcional a la tasa.

II. Vídeo entrelazado y vídeo progresivo

Un fotograma de vídeo contiene líneas de información espacial de una señal de vídeo. Para vídeo progresivo, estas líneas contienen muestras que empiezan desde un instante de tiempo y continúan a través de líneas sucesivas hasta la parte inferior del fotograma. Un fotograma I progresivo es un fotograma de vídeo progresivo intra-codificado. Un fotograma P progresivo es un fotograma de vídeo progresivo codificado usando predicción hacia delante, y un fotograma B progresivo es un fotograma de vídeo progresivo codificado usando predicción bidireccional.

Un fotograma de vídeo entrelazado típico consiste en dos campos explorados empezando en diferentes tiempos. Por ejemplo, haciendo referencia a la Figura 4, un fotograma 400 de vídeo entrelazado incluye el campo 410 superior y el campo 420 inferior. Típicamente, las líneas con número par (campo superior) se exploran empezando en un tiempo (por ejemplo, tiempo t) y las líneas con número impar (campo inferior) se exploran empezando en un tiempo (por ejemplo, tiempo $t + 1$) diferente (típicamente posterior). Esta temporización puede crear características similares a dientes de sierra en regiones de un fotograma de vídeo entrelazado donde está presente el movimiento puesto que los dos campos se exploran empezando en diferentes tiempos. Por esta razón, los fotogramas de vídeo entrelazado pueden reorganizarse de acuerdo con una estructura de campo, con las líneas impares agrupadas juntas en un campo, y las líneas pares agrupadas juntas en otro campo. Esta disposición, conocida como codificación de campo, es útil en imágenes de movimiento elevado para reducción de tales artefactos de bordes de sierra. Por otra parte, en regiones estáticas, el detalle de imagen en el fotograma de vídeo entrelazado puede conservarse más eficazmente sin una reorganización de este tipo. Por consiguiente, la codificación de fotograma a menudo se usa en fotogramas de vídeo entrelazado estáticos o de bajo movimiento, en los que se conserva la disposición de línea de campo alternante original.

Un fotograma de vídeo progresivo consiste en un fotograma de contenido con líneas no alternantes. En contraste a vídeo entrelazado, el vídeo progresivo no divide fotogramas de vídeo en campos separados, y se explora un fotograma completo de izquierda a derecha, de la parte superior a la parte inferior empezando en un único tiempo.

III. Codificación y decodificación de fotograma P y en un codificador y decodificador de WMV anterior

Un codificador y decodificador de WMV anterior usa codificación y decodificación progresiva y entrelazada en fotogramas P. En fotogramas P progresivos y entrelazados, se codifica un vector de movimiento en el codificador calculando un diferencial entre el vector de movimiento y un predictor de vector de movimiento, que se calcula basándose en vectores de movimiento vecinos. Y, en el decodificador, el vector de movimiento se reconstruye añadiendo el vector de movimiento diferencial al predictor de vector de movimiento, que se calcula de nuevo (esta

vez en el decodificador) basándose en vectores de movimiento vecinos. Se selecciona un predictor para el macrobloque actual o se selecciona un campo del macrobloque actual basándose en los predictores candidatos, y se calcula un vector de movimiento diferencial basándose en el predictor. El vector de movimiento puede reconstruirse añadiendo el vector de movimiento diferencial al predictor de vector de movimiento seleccionado en cualquiera del lado del codificador o del decodificador. Típicamente, los vectores de movimiento de luminancia se reconstruyen a partir de la información de movimiento codificada, y se derivan vectores de movimiento de crominancia desde los vectores de movimiento de luminancia reconstruidos.

A. Codificación y decodificación de fotograma P progresivo

Por ejemplo, en un codificador y decodificador de WMV anterior, los fotogramas P progresivos pueden contener macrobloques codificados en modo de un vector de movimiento (1 MV) o en modo de cuatro vectores de movimiento (4MV), o macrobloques saltados, con una decisión realizada generalmente en una base macrobloque a macrobloque. Los fotogramas P con únicamente macrobloques de 1 MV (y, potencialmente, macrobloques saltados) se denominan como fotogramas P de 1 MV, y los fotogramas P tanto con macrobloques de 1 MV y de 4MV (y, potencialmente, macrobloques saltados) se denominan como fotogramas P de MV mixto. Un vector de movimiento está asociado con cada macrobloque de 1 MV, y cuatro vectores de movimiento están asociados con cada macrobloque de 4MV (uno para cada bloque).

Las Figuras 5A y 5B son diagramas que muestran las localizaciones de macrobloques considerados para predictores de vector de movimiento candidato para un macrobloque en un fotograma P progresivo de 1 MV. Los predictores candidatos se toman de los macrobloques izquierdo, superior y superior-derecho, excepto en el caso donde el macrobloque es el último macrobloque en la fila. En este caso, el Predictor B se toma desde el macrobloque superior izquierdo en lugar del superior-derecho. Para el caso especial donde el fotograma es de un macrobloque de ancho, el predictor es siempre el Predictor A (el predictor superior). Cuando el Predictor A está fuera de los límites puesto que el macrobloque está en la fila superior, el predictor es el Predictor C. Diversas otras reglas tratan otros casos especiales tales como predictores intra-codificados.

Las Figuras 6A-10 muestran las localizaciones de los bloques o macrobloques considerados para los hasta tres vectores de movimiento candidatos para un vector de movimiento para un macrobloque de 1 MV o de 4MV en un fotograma de MV mixto. En las siguientes figuras, los cuadrados más grandes son límites de macrobloque y los cuadrados más pequeños son límites de bloque. Para el caso especial donde el fotograma es de un macrobloque de ancho, el predictor es siempre el Predictor A (el predictor superior). Diversas otras reglas tratan otros casos especiales tales como bloques de fila superior para macrobloques de 4MV de fila superior, macrobloques de 1 MV de fila superior y predictores intra-codificados.

Las Figuras 6A y 6B son diagramas que muestran localizaciones de bloques considerados para predictores de vector de movimiento candidato para un macrobloque actual de 1 MV en un fotograma de MV mixto. Los macrobloques vecinos pueden ser macrobloques de 1 MV o de 4MV. Las Figuras 6A y 6B muestran las localizaciones para los vectores de movimiento candidatos suponiendo que los vecinos son de 4MV (es decir, el Predictor A es el vector de movimiento para el bloque 2 en el macrobloque por encima del macrobloque actual, y el predictor C es el vector de movimiento para el bloque 1 en el macrobloque inmediatamente a la izquierda del macrobloque actual). Si cualquiera de los vecinos es un macrobloque de 1 MV, entonces el predictor de vector de movimiento mostrado en las Figuras 5A y 5B se toma para que sea el predictor de vector de movimiento para el macrobloque completo. Como muestra la Figura 6B, si el macrobloque es el último macrobloque en la fila, entonces el Predictor B es desde el bloque 3 del macrobloque superior izquierdo en lugar de desde el bloque 2 en el macrobloque superior-derecho como es el caso de otra manera.

Las Figuras 7A-10 muestran las localizaciones de bloques considerados para predictores de vector de movimiento candidato para cada uno de los 4 bloques de luminancia en un macrobloque de 4MV. Las Figuras 7A y 7B son diagramas que muestran las localizaciones de bloques considerados para predictores de vector de movimiento candidato para un bloque en la posición 0; Las Figuras 8A y 8B son diagramas que muestran las localizaciones de bloques considerados para predictores de vector de movimiento candidato para un bloque en la posición 1; La Figura 9 es un diagrama que muestra las localizaciones de bloques considerados para predictores de vector de movimiento candidato para un bloque en la posición 2; y la Figura 10 es un diagrama que muestra las localizaciones de bloques considerados para predictores de vector de movimiento candidato para un bloque en la posición 3. De nuevo, si un vecino es un macrobloque de 1 MV, se usa el predictor de vector de movimiento para el macrobloque para los bloques del macrobloque.

Para el caso donde el macrobloque es el primer macrobloque en la fila, el Predictor B para el bloque 0 se maneja de manera diferente que el bloque 0 para los macrobloques restantes en la fila (véanse las Figuras 7A y 7B). En este caso, el Predictor B se toma desde el bloque 3 en el macrobloque inmediatamente por encima del macrobloque actual en lugar de desde el bloque 3 en el macrobloque por encima y a la izquierda del macrobloque actual, como es el caso de otra manera. De manera similar, para el caso donde el macrobloque es el último macrobloque en la fila, el Predictor B para el bloque 1 se maneja de manera diferente (Figuras 8A y 8B). En este caso, el predictor se toma desde el bloque 2 en el macrobloque inmediatamente por encima del macrobloque actual en lugar de desde el bloque 2 en el macrobloque por encima y a la derecha del macrobloque actual, como es el caso de otra manera. En

general, si el macrobloque está en la primera columna del macrobloque, entonces el Predictor C para los bloques 0 y 2 se establece igual a 0.

B. Codificación y decodificación de fotograma P entrelazado en un codificador y decodificador de WMV anterior

- 5 Un codificador y decodificador de WMV anterior usa un formato de macrobloque 4:1:1 para fotogramas P entrelazados, que pueden contener macrobloques codificados en modo de campo o en modo de fotograma, o macrobloques saltados, con una decisión realizada generalmente en una base macrobloque a macrobloque. Dos vectores de movimiento están asociados con cada macrobloque codificado por campo (un vector de movimiento por campo), y un vector de movimiento está asociado con cada macrobloque codificado por fotograma. Un codificador
- 10 codifica conjuntamente información de movimiento, que incluye componentes de vector diferencial de movimiento horizontal y vertical, potencialmente junto con otra información de señalización.

- Las Figuras 11 y 12A-B muestran ejemplos de predictores candidatos para predicción de vector de movimiento para macrobloques codificados por fotograma 4:1:1 y macrobloques codificados por campo 4:1:1, respectivamente, en fotogramas P entrelazados en un codificador y decodificador de WMV anterior. La Figura 11 muestra predictores
- 15 candidatos A, B y C para un macrobloque codificado por fotograma actual 4:1:1 en una posición interior en un fotograma P entrelazado (no el primer o último macrobloque en una fila de macrobloques, ni en la fila superior). Los predictores pueden obtenerse desde diferentes direcciones candidatas distintas de las etiquetadas A, B y C (por ejemplo, en casos especiales tales como cuando el macrobloque actual es el primer macrobloque o último macrobloque en una fila, o en la fila superior, puesto que ciertos predictores no están disponibles para tales casos).
- 20 Para un macrobloque codificado por fotograma actual, los candidatos de predictor se calculan de manera diferente dependiendo de si los macrobloques vecinos están codificados por campo o codificados por fotograma. Para un macrobloque codificado por fotograma vecino, el vector de movimiento se toma simplemente como el candidato de predictor. Para un macrobloque codificado por campo vecino, el vector de movimiento candidato se determina promediando los vectores de movimiento de campo superior e inferior.

- 25 Las Figuras 12A-B muestran predictores candidatos A, B y C para un campo actual en un macrobloque codificado por campo 4:1:1 en una posición interior en el campo. En la Figura 12A, el campo actual es un campo inferior, y se usan los vectores de movimiento de campo inferior en los macrobloques vecinos como predictores candidatos. En la Figura 12B, el campo actual es un campo superior, y se usan los vectores de movimiento de campo superior en los macrobloques vecinos como predictores candidatos. Por lo tanto, para cada campo en un macrobloque codificado
- 30 por campo actual, el número de candidatos de predictor de vector de movimiento para cada campo es como máximo tres, proviniendo cada candidato desde el mismo tipo de campo (por ejemplo, superior o inferior) que el campo actual. De nuevo, se aplican diversos casos especiales (no mostrados) cuando el macrobloque actual es el primer macrobloque o último macrobloque en una fila, o en la fila superior, puesto que ciertos predictores no están disponibles para tales casos.

- 35 Para seleccionar un predictor de un conjunto de candidatos de predictor, el codificador y decodificador de WMV anterior en cuestión usan diferentes algoritmos de selección, tal como un algoritmo de mediana de tres o un algoritmo de mediana de cuatro. Un procedimiento para predicción de mediana de tres se describe en el pseudocódigo 1300 en la Figura 13. Un procedimiento para predicción de mediana de cuatro se describe en el pseudocódigo 1400 en la Figura 14.

IV. Predicción bidireccional

- Los fotogramas previstos bidireccionalmente (o fotogramas B) usan dos fotogramas del vídeo de origen como fotogramas de referencia (o ancla) en lugar del ancla usada en fotogramas P. Entre los fotogramas de ancla para un fotograma B típico, un fotograma de ancla proviene del pasado temporal y un fotograma de ancla proviene del futuro temporal. Haciendo referencia a la Figura 15, un fotograma B 1510 en una secuencia de vídeo tiene un fotograma
- 45 1520 de referencia temporalmente anterior y un fotograma 1530 de referencia temporalmente futuro. El uso de fotogramas B proporciona beneficios para compresión eficaz en términos de mayor economía de tasa de bits (por ejemplo, en presencia de movimiento de ciertos tipos, tal como oclusión). Los flujos de bits codificados con fotogramas B típicamente usan menos bits que los flujos de bits codificados sin fotogramas B, mientras proporcionan calidad visual similar. Los fotogramas B también proporcionan más opciones y flexibilidad cuando se usan en un espacio de dispositivo pequeño. Por ejemplo, un decodificador puede adaptar las restricciones de espacio y tiempo optando por no decodificar o visualizar fotogramas B, puesto que los fotogramas B en general no se usan como fotogramas de referencia. Las estimaciones para la mejora de tasa-distorsión en secuencias de vídeo que usan fotogramas B varían de 0 a aproximadamente el 50 %.

V. Codificación y decodificación de fotograma B en un codificador y decodificador de WMV anterior

- 55 Un codificador y decodificador de WMV anterior usa fotogramas B. Mientras los macrobloques en fotogramas previstos hacia delante (por ejemplo, fotogramas P) tienen únicamente un modo de predicción direccional (hacia delante, desde fotogramas I o P anteriores), los macrobloques en los fotogramas B pueden predecirse usando cinco diferentes modos de predicción: hacia delante, hacia atrás, directa, interpolada e intra. El codificador selecciona y

señaliza diferentes modos de predicción en el flujo de bits. Por ejemplo, el codificador WMV anterior en cuestión envía un plano de bits comprimido en el nivel de fotograma que indica una decisión de modo directo/no directo para cada macrobloque de un fotograma B, mientras que se indican modos no directos (tales como modos hacia delante, hacia atrás e interpolado) en el nivel de macrobloque.

- 5 El modo directo es similar a la predicción de fotograma P convencional. En modo directo, se deriva un macrobloque desde un ancla temporalmente anterior. En modo hacia atrás, se deriva un macrobloque desde un ancla temporalmente posterior. Los macrobloques previstos en modos directo o interpolado usan tanto anclas hacia delante como hacia atrás para predicción. Los modos directo e interpolado usan promedio de redondeo hacia arriba para combinar los valores de píxeles de las dos referencias en un conjunto de píxeles de macrobloque de acuerdo con la siguiente ecuación:

$$\text{Valor de píxel promedio} = (\text{valor interpolado hacia delante} + \text{valor interpolado hacia atrás} + 1) \gg 1$$

A. Codificación de fracción y cambio de escala de vectores de movimiento co-localizados

- 15 En el codificador y decodificador de WMV anterior en cuestión, el codificador deriva implícitamente vectores de movimiento de modo directo cambiando de escala un vector de movimiento co-localizado para el ancla hacia delante. La operación de cambio de escala depende de la posición temporal del fotograma B actual con relación a sus anclas. Para codificar la posición temporal de una imagen de referencia, el codificador usa codificación de fracción.

- 20 En codificación de fracción, el codificador codifica explícitamente una posición temporal para un fotograma B actual como una fracción de la distancia entre sus dos anclas. La variable BFRACTION se usa para representar diferentes fracciones y se envía en el nivel de fotograma. La fracción toma un conjunto limitado de valores discretos entre 0 y 1. Para vectores de movimiento de modo directo, el codificador y decodificador usan esta fracción para cambiar de escala un vector de movimiento (MV) co-localizado en un fotograma de referencia, derivando de esta manera vectores de movimiento de modo directo (MV_F y MV_B) implicados para el fotograma B actual implementando las siguientes operaciones de cambio de escala:

$$25 \quad MV_F = \text{Fracción} * MV$$

$$MV_B = (\text{Fracción} - 1) * MV$$

- 30 La Figura 16 muestra cómo la codificación de fracción permite que el codificador cambie de escala de manera arbitraria el movimiento entre fotogramas de referencia circundantes. Para derivar MV_F y MV_B para el macrobloque 1610 actual que se está codificando en el fotograma B 1620, el codificador y decodificador cambian de escala el vector de movimiento (MV) del macrobloque correspondiente en el fotograma 1630 de referencia futuro usando codificación de fracción. En el ejemplo mostrado en la Figura 16, para las fracciones p y q, p + q = 1. El codificador y decodificador usan los dos vectores de movimiento implicados para tratar los macrobloques en el fotograma 1640 de referencia anterior y en el fotograma 1630 de referencia futuro, y usan el promedio de estos para predecir el macrobloque 1610 actual. Por ejemplo, en la Figura 16, MV_F = (dx*p, dy*p) y MV_B = (-dx*q, -dy*q).

- 35 La tabla 1700 en la Figura 17 es una tabla de códigos de longitud variable (VLC) para el elemento de flujo de bits BFRACTION. En el ejemplo mostrado en la tabla 1700, las palabras de código de 3 bits son las palabras de código "corto", y las palabras de código de 7 bits son las palabras de código "largo". El decodificador halla un factor de cambio de escala basándose en el numerador y denominador de la fracción de acuerdo con el pseudocódigo 1800 mostrado en la Figura 18.

- 40 Una vez que se ha determinado el factor de cambio de escala, el decodificador lo usa para cambiar de escala los elementos x e y del vector de movimiento para el macrobloque co-localizado. Dado que el fotograma de ancla posterior era un fotograma P (para fotogramas I, todos los vectores de movimiento se supone que son (0, 0)) y que el macrobloque co-localizado contiene un vector de movimiento (MV_X, MV_Y), el decodificador deriva dos vectores de movimiento, haciendo uno referencia (MV_{X_F}, MV_{Y_F}) al fotograma de ancla hacia delante (previo), y haciendo el otro referencia (MV_{X_B}, MV_{Y_B}) al de ancla hacia atrás (posterior).

- 45 El decodificador realiza el cambio de escala de acuerdo con el pseudocódigo 1900 mostrado en la Figura 19. En la función *Scale_Direct_MV* en el pseudocódigo 1900, las entradas MV_X y MV_Y son los elementos x e y del vector de movimiento desde el macrobloque co-localizado de la imagen de referencia futura, y las salidas MV_{X_F}, MV_{Y_F}, MV_{X_B} y MV_{Y_B} son los elementos x e y de los vectores de movimiento que apuntan hacia delante y hacia atrás para el macrobloque que se está decodificando.

B. Fotogramas B/I

- 55 El codificador y decodificador de WMV anterior en cuestión también usa fotogramas B intra ("fotogramas B/I") en codificación y decodificación progresiva. Los fotogramas B/I se codifican como los fotogramas I, en que no dependen de fotogramas de referencia. Pero a diferencia de los fotogramas I, los fotogramas B/I no son fotogramas clave; otros fotogramas no están permitidos a usar fotogramas B/I como anclas.

C. Fotogramas B entrelazados

El codificador y decodificador de WMV anterior en cuestión también usa fotogramas B entrelazados. Los macrobloques en fotogramas B entrelazados pueden codificarse por campo o codificarse por fotograma. Los macrobloques codificados por fotograma pueden tener uno, dos (por ejemplo, vectores de movimiento hacia delante y hacia atrás para el modo interpolado, vectores de movimiento hacia delante y hacia atrás derivados para modo directo), o ningún vector de movimiento, y los macrobloques codificados por campo pueden tener hasta cuatro vectores de movimiento, dependiendo del modo de predicción. Por ejemplo, en un macrobloque codificado por campo de modo directo, se derivan cuatro vectores de movimiento implícitos: un vector de movimiento hacia delante y hacia atrás para el campo superior, y un vector de movimiento hacia delante y hacia atrás para el campo inferior.

Aunque el codificador y decodificador de WMV anterior en cuestión usa fotogramas B entrelazados, están limitados de varias maneras importantes. Por ejemplo, únicamente se permite un modo de predicción de macrobloque (por ejemplo, modo directo, modo hacia delante, etc.) por macrobloque, no se usa codificación de 4MV (es decir, un vector de movimiento para cada bloque de un macrobloque), y ninguna parte de ningún fotograma B puede ser una referencia para compensación de movimiento para cualquier fotograma. Como otro ejemplo, la codificación y decodificación entrelazada en el codificador y decodificador de WMV anterior en cuestión (incluyendo fotogramas B entrelazados) se realiza usando únicamente un formato de macrobloque 4:1:1.

VI. Normas para compresión y descompresión de vídeo.

Aparte de los codificadores y decodificadores de WMV anteriores, varias normas internacionales se refieren a compresión y descompresión de vídeo. Estas normas incluyen las normas del Grupo de Expertos de Imágenes en Movimiento ["MPEG"] 1, 2 y 4 y las normas H.261, H.262, H.263 y H.264 de la Unión Internacional de Telecomunicación ["ITU"]. Uno de los procedimientos principales para conseguir compresión de datos de secuencias de vídeo digital en las normas internacionales es reducir la redundancia temporal entre imágenes. Estos esquemas de compresión conocidos (MPEG-1, MPEG-2, MPEG-4, H.261, H.263, etc.) usan estimación y compensación de movimiento. Por ejemplo, un fotograma actual se divide en regiones cuadradas uniformes (por ejemplo, bloques y/o macrobloques). Una región coincidente para cada región actual se especifica enviando información de vector de movimiento para la región. El vector de movimiento indica la localización de la región en un fotograma previamente codificado (y reconstruido) que se ha de usar como un predictor para la región actual. Se deriva una diferencia de pixel a pixel, denominada la señal de error, entre la región actual y la región en el fotograma de referencia. Esta señal de error normalmente tiene entropía inferior que la señal original. Por lo tanto, la información puede codificarse a una tasa de bits inferior. Como en codificadores y decodificadores de WMV anteriores, puesto que un valor de vector de movimiento a menudo está correlacionado con vectores de movimiento espacialmente circundantes, la compresión de los datos usados para representar la información del vector de movimiento puede conseguirse codificando el diferencial entre el vector de movimiento actual y un predictor basándose en vectores de movimiento vecinos previamente codificados.

Algunas normas internacionales describen estimación y compensación de movimiento en fotogramas de vídeo entrelazados. La norma H.262 permite que un fotograma de vídeo entrelazado se codifique como un único fotograma o como dos campos, donde la codificación por fotograma o la codificación por campo puede seleccionarse de manera adaptativa en una base fotograma a fotograma. La norma H.262 describe predicción basada en campo, que es un modo de predicción que usa únicamente un campo de un fotograma de referencia. La norma H.262 también describe predicción de primos dobles, que es un modo de predicción en el que se promedian dos predicciones basadas en campo hacia delante para un bloque de 16x16 en una imagen P entrelazada. La sección 7.6 de la norma H.262 describe "predicción de campo", que incluye seleccionar entre dos campos de referencia para usar compensación de movimiento para un macrobloque de un campo actual de un fotograma de vídeo entrelazado. La sección 7.6.3 describe predicción de vector de movimiento y reconstrucción, en la que un vector de movimiento reconstruido para un macrobloque dado se hace el predictor de vector de movimiento para un macrobloque codificado/decodificado posteriormente. Tal predicción de vector de movimiento falla al predecir adecuadamente vectores de movimiento para macrobloques de campos de fotogramas de vídeo entrelazados en muchos casos.

Además, la sección 7.6 de la norma H.262 describe "predicción de campo" y "predicción de fotograma" de imágenes B. En predicción de campo y predicción de fotograma, la predicción se realiza para imágenes B usando los dos fotogramas de referencia reconstruidos más recientemente (omitiendo otras imágenes B intermedias), que pueden haberse codificado como cualquiera de dos campos o como un único fotograma.

Dada la importancia crítica de la compresión y descompresión de vídeo para vídeo digital, no es sorprendente que la compresión y descompresión de vídeo sean campos altamente desarrollados. Sean cuales sean los beneficios de las técnicas de compresión y descompresión de vídeo anteriores, sin embargo, no tienen las ventajas de las siguientes técnicas y herramientas.

Louis Kerofsky y col: "Adaptive Syntax for MTYPE", 13th VCEG meeting, 2 de abril de 2001 a 4 de abril de 2001, Austin, Texas, Estados Unidos, Videocoding Experts Group of ITU-T SG. 16, n.º VCEG-M14, 28 de marzo de 2001 se refiere a una técnica sencilla para conseguir ganancia de codificación a bajas tasas de bits dentro del modelo de

codificación UVLC. Se centra en la codificación de MBtype. Se usan modelos sencillos para la distribución de probabilidad MBtype y se proponen modificaciones menores a la sintaxis para conseguir ganancia significativa en secuencias de Bajo Movimiento, y consigue mejoras ligeras a través de otras secuencias.

- 5 Gary Sullivan y col: "Meeting report of the thirteenth meeting (meeting M) of the ITU-T Q.6/16 Video Coding Experts Group (VCEG) - Austin, Texas, 2 a 4 de abril de 2001", 13th VCEG meeting, 2 de abril de 2001 a 4 de abril de 2001, Austin, Texas, Estados Unidos, Videocoding Experts Group of ITU-T SG.16, n.º VCEG-M82d1, 15 de junio de 2001, se reconoce por la presente.

- 10 El documento WO 03/047272 A2 se refiere a un procedimiento y sistema de codificación y decodificación de contenido de vídeo digital. El contenido de vídeo digital comprende un flujo de imágenes que pueden ser imágenes intra, previstas o bi-previstas. Cada una de las imágenes comprende macrobloques que pueden dividirse adicionalmente en bloques más pequeños. El procedimiento conlleva codificar y decodificar cada uno de los bloques más pequeños en cada imagen en dicho flujo de imágenes en cualquiera del modo de fotograma o modo de campo.

- 15 El documento del Equipo Mixto de Vídeo (JVT): "B pictures in JVT", 4th JVT meeting, 61st MPEG meeting, 22 de julio de 2002 a 26 de julio de 2002, Klagenfurt, AT, Joint video team of ISO/IEC JTC1/SC29/WG11 e ITU-T SG. 16, n.º JVT-D155, 26 de julio de 2002, páginas 1 a 19 se refiere a modificaciones necesarias a la descripción actual de imágenes B dentro de la norma del JVT de acuerdo con las decisiones de 4th JVT en Klagenfurt, Austria. Todos los asuntos relacionados, incluyendo predicción de vector de movimiento, predicción bi-predictiva adaptativa y modo directo, se describen en detalle.

- 20 El documento WO 98/44743 se refiere a un procedimiento para decodificar un flujo de bits que representa una imagen que se ha codificado. El procedimiento incluye las etapas de: realizar una decodificación por entropía del flujo de bits para formar una pluralidad de coeficientes de transformación y una pluralidad de vectores de movimiento; realizar una transformación inversa en la pluralidad de coeficientes de transformación para formar una pluralidad de bloques de error; determinar una pluralidad de bloques previstos basándose en estimación de movimiento bidireccional que emplea los vectores de movimiento, en el que la estimación de movimiento bidireccional incluye un modo de predicción directa y un segundo modo de predicción; y, añadir la pluralidad de bloques de error a la pluralidad de bloques previstos para formar la imagen. El segundo modo de predicción puede incluir modos de predicción hacia delante, hacia atrás e interpolado.

Sumario

- 30 Es el objeto de la presente invención proporcionar técnicas y herramientas para codificar y decodificar bidireccionalmente imágenes previstas.

Este objeto se resuelve mediante la materia objeto de las reivindicaciones independientes.

Se proporcionan realizaciones en las reivindicaciones dependientes.

Con respecto a la materia objeto reivindicada, se hace referencia en particular a las secciones VI, VIII, XI y XIV, mientras que las otras figuras y partes de la descripción son útiles para entender la invención.

- 35 En resumen, la descripción detallada se refiere a diversas técnicas y herramientas para codificar y decodificar bidireccionalmente fotogramas vídeo entrelazado bidireccionalmente previstos (por ejemplo, campos B entrelazados, fotogramas B entrelazados). Las técnicas y herramientas descritas mejoran el rendimiento de tasa/distorsión y facilitan mejor soporte para dispositivos con recursos de CPU inferiores (por ejemplo, en dispositivos con factores de forma más pequeños).

- 40 Las realizaciones descritas implementan una o más de las técnicas y herramientas descritas para codificar y/o decodificar imágenes B entrelazadas incluyendo, pero sin limitación, lo siguiente:

- 45 En un aspecto, para fotogramas B entrelazados, un codificador/decodificador conmuta modos de predicción entre los campos en un macrobloque codificado por campo de un fotograma B entrelazado. Por ejemplo, el codificador/decodificador conmuta entre un modo de predicción hacia delante para el campo superior y un modo hacia atrás para el campo inferior en el macrobloque codificado por campo. Conmutar entre modos de predicción hacia delante y hacia atrás dentro del mismo macrobloque codificado por campo permite más flexibilidad para hallar modos de predicción eficaces para diferentes porciones de fotogramas B entrelazados.

- 50 En otro aspecto, para fotogramas B entrelazados, un codificador/decodificador calcula vectores de movimiento de modo directo para un macrobloque actual seleccionando como máximo un vector de movimiento representativo para cada uno de los campos superior e inferior del macrobloque co-localizado del ancla previamente decodificada temporalmente posterior. Por ejemplo, la selección se realiza basándose al menos en parte en el modo de codificación del macrobloque del fotograma B entrelazado actual (por ejemplo, modo de 1MV, modo de 2 MV de campo, etc.).

En otro aspecto, para campos B entrelazados o fotogramas B entrelazados, un codificador/decodificador usa

codificación de 4MV. Por ejemplo, se usa 4MV en modos de predicción de una dirección (modos hacia delante o hacia atrás), pero no en otros modos de predicción disponibles (por ejemplo, directo, interpolado). Usar 4MV permite compensación de movimiento más precisa para campos B entrelazados y fotogramas B entrelazados; limitar 4MV a modos hacia delante y hacia atrás reduce la tara de codificación y evita la complejidad de decodificación asociada con combinar 4MV con modos tales como directo e interpolado.

En otro aspecto, para campos B entrelazados o fotogramas B entrelazados, se predicen vectores de movimiento hacia delante mediante un codificador/decodificador usando vectores de movimiento hacia delante previamente reconstruidos (o estimados) desde una memoria intermedia de vector de movimiento hacia delante, y se predicen vectores de movimiento hacia atrás usando vectores de movimiento hacia atrás previamente reconstruidos (o estimados) desde una memoria intermedia de vector de movimiento hacia atrás. Los vectores de movimiento resultantes se añaden a la correspondiente memoria intermedia. Los huecos en las memorias intermedias de vectores de movimiento pueden rellenarse con valores de vector de movimiento estimados. Por ejemplo, para fotogramas B entrelazados, cuando se usa predicción hacia delante para predecir un vector de movimiento y el vector de movimiento se añade a la memoria intermedia de vector de movimiento hacia delante, se rellena la posición correspondiente en una memoria intermedia de vector de movimiento hacia atrás ("relleno de huecos") con lo que sería el vector de movimiento previsto usando únicamente vectores de movimiento hacia atrás como predictores. Como otro ejemplo, para campos B entrelazados, para elegir entre diferentes vectores de movimiento de polaridad (por ejemplo, "misma polaridad" o "polaridad opuesta") para relleno de huecos, un codificador/decodificador selecciona un vector de movimiento de campo de polaridad dominante. La distancia entre las anclas y fotogramas actuales se calcula usando diversos elementos de sintaxis, y la distancia calculada se usa para cambiar de escala vectores de movimiento de campo de referencia. Las memorias intermedias de vector de movimiento separadas y el relleno de huecos en las memorias intermedias de vector de movimiento separada permite predicción de vector de movimiento más precisa para campos B entrelazados y fotogramas B entrelazados.

En otro aspecto, para campos B entrelazados, un codificador/decodificador usa fotogramas de "auto-referencia". Por ejemplo, un segundo campo B en un fotograma actual hace referencia al primer campo B desde el fotograma actual en predicción de movimiento compensado. Permitir al primer campo B en un fotograma que actúe como una referencia para el segundo campo B en el fotograma permite predicción más precisa del segundo campo B, mientras conserva también los beneficios de escalabilidad temporal de tener campos B en el fotograma actual.

En otro aspecto, para campos B entrelazados, un codificador envía información binaria que indica si un modo de predicción es hacia delante o no hacia delante para uno o más macrobloques en un campo B entrelazado. Por ejemplo, el codificador envía información de decisión hacia delante/no hacia delante a nivel de campo B en un plano de bits comprimido. Enviar información de decisión de modo de predicción hacia delante/no hacia delante en un plano de bits comprimido a nivel de campo B puede reducir tara de codificación para codificación de modo de predicción. Un decodificador realiza la decodificación correspondiente.

En otro aspecto, para campos B entrelazados, un codificador/decodificador selecciona vectores de movimiento de modo directo usando lógica que favorece la polaridad dominante si el macrobloque correspondiente en el campo correspondiente de la siguiente imagen de ancla se codificó usando cuatro vectores de movimiento. Por ejemplo, si los vectores de movimiento de la misma polaridad del macrobloque correspondiente superan en número sus vectores de movimiento de polaridad opuesta, el codificador/decodificador calcula la mediana de los vectores de movimiento de la misma polaridad para obtener un vector de movimiento para derivar vectores de movimiento de modo directo. Este procedimiento de selección permite la derivación de vectores de movimiento de modo directo precisos para campos B entrelazados que tienen anclas con macrobloques de 4MV.

Las diversas técnicas y herramientas pueden usarse en combinación o de manera independiente.

Se harán evidentes características y ventajas adicionales a partir de la siguiente descripción detallada de diferentes realizaciones que continúa con referencia a los dibujos adjuntos.

Breve descripción de los dibujos

La Figura 1 es un diagrama que muestra estimación de movimiento en un codificador de vídeo de acuerdo con la técnica anterior.

La Figura 2 es un diagrama que muestra compresión basada en bloques para un bloque de 8x8 de residuos de predicción en un codificador de vídeo de acuerdo con la técnica anterior.

La Figura 3 es un diagrama que muestra descompresión basada en bloques para un bloque de 8x8 de residuos de predicción en un codificador de vídeo de acuerdo con la técnica anterior.

La Figura 4 es un diagrama que muestra un fotograma entrelazado de acuerdo con la técnica anterior.

Las Figuras 5A y 5B son diagramas que muestran localizaciones de macrobloques para predictores de vector de movimiento candidato para un macrobloque de 1 MV en un fotograma P progresivo de acuerdo con la técnica anterior.

Las Figuras 6A y 6B son diagramas que muestran localizaciones de bloques para predictores de vector de movimiento candidato para un macrobloque de 1 MV en un fotograma P progresivo de 1 MV/4MV mixto de acuerdo con la técnica anterior.

Las Figuras 7A, 7B, 8A, 8B, 9 y 10 son diagramas que muestran las localizaciones de bloques para predictores de vector de movimiento candidato para un bloque en diversas posiciones en un macrobloque de 4MV en un fotograma P progresivo de 1 MV/4MV mixto de acuerdo con la técnica anterior.

La Figura 11 es un diagrama que muestra predictores de vector de movimiento candidato para un macrobloque codificado por fotograma actual en un fotograma P entrelazado de acuerdo con la técnica anterior.

Las Figuras 12A-12B son diagramas que muestran predictores de vector de movimiento candidato para un macrobloque codificado por campo actual en un fotograma P entrelazado de acuerdo con la técnica anterior.

Las Figuras 13 y 14 son diagramas de código que muestran pseudocódigo para realizar cálculos de mediana de 3 y mediana de 4, respectivamente, de acuerdo con la técnica anterior.

La Figura 15 es un diagrama que muestra un fotograma B con fotogramas de referencia pasados y futuros de acuerdo con la técnica anterior.

La Figura 16 es un diagrama que muestra la predicción de modo directo con codificación de fracción de acuerdo con la técnica anterior.

La Figura 17 muestra una tabla de VLC para el elemento de flujo de bits BFRACTION de acuerdo con la técnica anterior.

Las Figuras 18 son un listado de código que muestra pseudocódigo para hallar un factor de cambio de escala para cambiar de escala un vector de movimiento para un macrobloque co-localizado en predicción de modo directo de acuerdo con la técnica anterior.

Las Figuras 19 son un listado de código que muestra pseudocódigo para cambiar de escala los elementos x e y de un vector de movimiento en un macrobloque co-localizado de acuerdo con un factor de cambio de escala de acuerdo con la técnica anterior.

La Figura 20 es un diagrama de bloques de un entorno informático adecuado en conjunto con el que pueden implementarse varias realizaciones descritas.

La Figura 21 es un diagrama de bloques de un sistema de codificador de vídeo generalizado en conjunto con el que pueden implementarse varias realizaciones descritas.

La Figura 22 es un diagrama de bloques de un sistema de decodificador de vídeo generalizado en conjunto con el que pueden implementarse varias realizaciones descritas.

La Figura 23 es un diagrama de un formato de macrobloque usado en varias realizaciones descritas.

La Figura 24A es un diagrama de parte de un fotograma de vídeo entrelazado, que muestra líneas alternas de un campo superior y un campo inferior. La Figura 24B es un diagrama del fotograma de vídeo entrelazado organizado para codificar/decodificar como un fotograma, y la Figura 24C es un diagrama del fotograma de vídeo entrelazado organizado para codificar/decodificar como campos.

Las Figuras 25 y 26 son diagramas que muestran campos P entrelazados que tienen dos campos de referencia.

Las Figuras 27 y 28 son diagramas que muestran campos P entrelazados que usan el campo de referencia permisible más reciente.

Las Figuras 29 y 30 son diagramas que muestran campos P entrelazados que usan el segundo campo de referencia permisible más reciente.

La Figura 31 es un diagrama que muestra relaciones entre componentes verticales de vectores de movimiento y una localización espacial correspondiente para diferentes combinaciones de polaridades de campo actual y de referencia.

La Figura 32 es un diagrama que muestra dos conjuntos de tres predictores de vector de movimiento candidato para un macrobloque actual.

Las Figuras 33A-33F son listados de código que muestran pseudocódigo para calcular predictores de vector de movimiento en campos P o B entrelazados de campo de dos referencias.

Las Figuras 34A-34B son listados de código que muestran pseudocódigo para cambiar de escala un predictor desde un campo para derivar un predictor de otro campo.

Las Figuras 35 y 36 son tablas que muestran valores de operación de cambio de escala asociados con diferentes distancias de fotograma de referencia.

La Figura 37 es un diagrama que muestra vectores de movimiento para bloques de luminancia y vectores de movimiento derivados para bloques de crominancia en un macrobloque de MV de 2 campos de un fotograma P entrelazado.

La Figura 38 es un diagrama que muestra diferentes vectores de movimiento para cada uno de cuatro bloques de luminancia, y vectores de movimiento derivados para cada uno de cuatro subbloques de crominancia, en un macrobloque de MV de 4 fotogramas de un fotograma P entrelazado.

La Figura 39 es un diagrama que muestra vectores de movimiento para bloques de luminancia y vectores de movimiento derivados para bloques de crominancia en un macrobloque de MV de 4 campos de un fotograma P entrelazado.

Las Figuras 40A-40B son diagramas que muestran predictores candidatos para un macrobloque actual de un fotograma P entrelazado.

La Figura 41 es un diagrama de flujo que muestra una técnica para predecir vectores de movimiento para campos individuales en un macrobloque codificado por campo en un fotograma B entrelazado usando diferentes modos de predicción.

La Figura 42 es un diagrama de flujo que muestra una técnica para calcular vectores de movimiento de modo directo para macrobloques de fotogramas B entrelazados.

La Figura 43 es un diagrama que muestra vectores de movimiento almacenados en memoria intermedia para los bloques de un macrobloque co-localizado de un fotograma de ancla previamente decodificado temporalmente

posterior para uso al calcular vector o vectores de movimiento de modo directo para un macrobloque de un fotograma B entrelazado.

La Figura 44 es un diagrama de flujo que muestra una técnica para predecir vectores de movimiento para un macrobloque actual en una imagen B entrelazada usando una memoria intermedia de vector de movimiento hacia delante y/o hacia atrás.

La Figura 45 es un diagrama que muestra vectores de movimiento en una memoria intermedia de vector de movimiento hacia delante y una memoria intermedia de vector de movimiento hacia atrás para predecir vectores de movimiento para macrobloques.

La Figura 46 es un diagrama que muestra vectores de movimiento de campo superior e inferior para un macrobloque reconstruido en una memoria intermedia de vector de movimiento hacia delante y una memoria intermedia de vector de movimiento hacia atrás.

La Figura 47 es un listado de código que muestra pseudocódigo que describe el procedimiento de selección de polaridad para almacenamiento en memoria intermedia de valor real y relleno de huecos en predicción de vector de movimiento de campo B entrelazado.

Las Figuras 48A-48B son listados de código que muestran pseudocódigo para cambiar de escala un predictor desde un campo para derivar un predictor desde otro campo para un campo B entrelazado previsto hacia atrás.

La Figura 49 es una tabla que muestra valores de operación de cambio de escala asociados con diferentes distancias de fotograma de referencia para un primer campo B entrelazado.

Las Figuras 50A y 50B son diagramas que muestran campos de referencia para un campo B entrelazado.

La Figura 51 es un diagrama de flujo que muestra una técnica para codificar información de decisión de modo de predicción hacia delante/no hacia delante para macrobloques de un campo B entrelazado en un codificador de vídeo que tiene uno o más modos de codificación de plano de bits.

La Figura 52 es un diagrama de flujo que muestra una técnica para decodificar información de decisión de modo de predicción hacia delante/no hacia delante para macrobloques de un campo B entrelazado, que se codificó mediante un codificador de vídeo que tiene uno o más modos de codificación de plano de bits.

La Figura 53 es un listado de código que muestra pseudocódigo que describe un procedimiento de selección para un vector de movimiento para usar como una base para vectores de movimiento de modo directo en campos B entrelazados.

La Figura 54 es un diagrama que muestra una sintaxis de flujo de bits de capa de fotograma para campo B entrelazado o campos BI en una implementación combinada.

La Figura 55 es un diagrama que muestra una sintaxis de flujo de bits de capa de campo para campos B entrelazados en una implementación combinada.

La Figura 56 es un diagrama que muestra una sintaxis de flujo de bits de capa de campo para campos BI entrelazados en una implementación combinada.

La Figura 57 es un diagrama que muestra una sintaxis de flujo de bits de capa de macrobloque para macrobloques de campos B entrelazados en una implementación combinada.

La Figura 58 es un diagrama que muestra una sintaxis de flujo de bits de capa de macrobloque para macrobloques de campos BI entrelazados en una implementación combinada.

La Figura 59 es un diagrama que muestra una sintaxis de flujo de bits de capa de fotograma para fotogramas B entrelazados en una implementación combinada.

La Figura 60 es un diagrama que muestra una sintaxis de flujo de bits de capa de macrobloque para macrobloques de fotogramas B entrelazados en una implementación combinada.

Las Figuras 61A-61B son listados de código que muestran pseudocódigo para decodificar información de predictor de vector de movimiento diferencial y dominante/no dominante en una implementación combinada.

Las Figuras 62A-62F son listados de código que muestran pseudocódigo para calcular predictores de vector de movimiento en campos P entrelazados de dos referencias en una implementación combinada.

La Figura 63 es un listado de código que muestra pseudocódigo para determinar un campo de referencia para un campo B entrelazado en una implementación combinada.

La Figura 64 es un listado de código que muestra pseudocódigo para recopilar vectores de movimiento candidatos para macrobloques de 1 MV en un fotograma P entrelazado en una implementación combinada.

Las Figuras 65, 66, 67 y 68 son listados de código que muestran pseudocódigo para recopilar vectores de movimiento candidatos para macrobloques de MV de 4 tramas en un fotograma P entrelazado en una implementación combinada.

Las Figuras 69 y 70 son listados de código que muestran pseudocódigo para recopilar vectores de movimiento candidatos para macrobloques de MV de 2 campos en un fotograma P entrelazado en una implementación combinada.

Las Figuras 71, 72, 73 y 74 son listados de código que muestran pseudocódigo para recopilar vectores de movimiento candidatos para macrobloques de MV de 4 campos en un fotograma P entrelazado en una implementación combinada.

La Figura 75 es un listado de código que muestra pseudocódigo para calcular predictores de vector de movimiento para vectores de movimiento de fotograma en un fotograma P entrelazado en una implementación combinada.

La Figura 76 es un listado de código que muestra pseudocódigo para calcular predictores de vector de movimiento para vectores de movimiento de campo en un fotograma P entrelazado en una implementación combinada.

La Figura 77A y 77B son listados de código que muestran pseudocódigo para decodificar un vector de

movimiento diferencial para unos fotogramas P entrelazados y fotogramas B en una implementación combinada. La Figura 78 es un listado de código que muestra pseudocódigo para derivar un vector de movimiento de crominancia en un fotograma P entrelazado en una implementación combinada.

5 Las Figuras 79A - 79C son diagramas que muestran piezas para modos de codificación de plano de bits Norm-6 y Dif-6 para información de decisión de modo de predicción hacia delante/no hacia delante para macrobloques de un campo B entrelazado.

Descripción detallada

10 La presente solicitud se refiere a técnicas y herramientas para compresión y descompresión eficaz de vídeo entrelazado. En diversas realizaciones descritas, un codificador y decodificador de vídeo incorpora técnicas para codificar y decodificar bidireccionalmente fotogramas de vídeo previstos entrelazados, y técnicas de señalización correspondientes para uso con un formato de flujo de bits o sintaxis que comprende diferentes capas o niveles (por ejemplo, nivel de secuencia, nivel de fotograma, nivel de campo, nivel de macrobloque y/o nivel de bloque).

15 Son posibles diversas alternativas a las implementaciones descritas en el presente documento. Por ejemplo, las técnicas descritas con referencia a diagramas de flujo pueden modificarse cambiando la ordenación de las etapas mostradas en los diagramas de flujo, repitiendo u omitiendo ciertas etapas, etc. Como otro ejemplo, aunque algunas implementaciones se describen con referencia a formatos de macrobloque específicos, pueden usarse también otros formatos. Además, las técnicas y herramientas descritas con referencia a predicción bidireccional pueden ser aplicables también a otros tipos de predicción.

20 Las diversas técnicas y herramientas pueden usarse en combinación o de manera independiente. Diferentes realizaciones implementan una o más de las técnicas y herramientas descritas. Algunas técnicas y herramientas descritas en el presente documento pueden usarse en un codificador o decodificador de vídeo, o algún otro sistema no específicamente limitado a codificación o decodificación de vídeo.

I. Entorno informático

25 La Figura 20 ilustra un ejemplo generalizado de un entorno 2000 informático adecuado en el que pueden implementarse varias de las realizaciones descritas. El entorno 2000 informático no se pretende para sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad, ya que las técnicas y herramientas pueden implementarse en diversos entornos informáticos de fin general o de fin especial.

30 Con referencia a la Figura 20, el entorno 2000 informático incluye al menos una unidad 2010 de procesamiento y memoria 2020. En la Figura 20, esta configuración 2030 más básica se incluye con una línea discontinua. La unidad 2010 de procesamiento ejecuta instrucciones ejecutables por ordenador y puede ser un procesador real o uno virtual. En un sistema de multiprocesamiento, múltiples unidades de procesamiento ejecutan instrucciones ejecutables por ordenador para aumentar la potencia de procesamiento. La memoria 2020 puede ser memoria volátil (por ejemplo, registros, caché, RAM), memoria no volátil (por ejemplo, ROM, EEPROM, memoria flash, etc.), o alguna combinación de las dos. La memoria 2020 almacena software 2080 que implementa un codificador o decodificador de vídeo con predicción bidireccional de fotogramas de vídeo entrelazados.

35 Un entorno informático puede tener características adicionales. Por ejemplo, el entorno 2000 informático incluye el almacenamiento 2040, uno o más dispositivos 2050 de entrada, uno o más dispositivos 2060 de salida y una o más conexiones 2070 de configuración. Un mecanismo de interconexión (no mostrado) tal como un bus, controlador o red interconecta los componentes del entorno 2000 informático. Típicamente, el software de sistema operativo (no mostrado) proporciona un entorno operativo para que otro software se ejecute en el entorno 2000 informático, y coordine actividades de los componentes del entorno 2000 informático.

40 El almacenamiento 2040 puede ser extraíble o no extraíble, e incluye discos magnéticos, cintas o casetes magnéticas, CD-ROM, DVD o cualquier otro medio que pueda usarse para almacenar información y que pueda accederse en el entorno 2000 informático. El almacenamiento 2040 almacena instrucciones para que el software 2080 implemente el codificador o decodificador de vídeo.

45 El dispositivo o dispositivos 2050 de entrada puede ser un dispositivo de entrada táctil tal como un teclado, ratón, lapicero, o bola de mando, un dispositivo de entrada de voz, un dispositivo de exploración u otro dispositivo que proporcione entrada al entorno 2000 informático. Para codificación de audio o de vídeo, el dispositivo o dispositivos 2050 de entrada pueden ser una tarjeta de sonido, tarjeta de vídeo, tarjeta sintonizadora de TV o dispositivo similar que acepte entrada de audio o de vídeo en forma analógica o digital, o un CD-ROM o CD-RW que lea muestras de audio o de vídeo en el entorno 2000 informático. El dispositivo o dispositivos 2060 de salida pueden ser una pantalla, impresora, altavoz, grabadora de CD u otro dispositivo que proporcione salida desde el entorno 2000 informático.

50 La conexión o conexiones 2070 de comunicación posibilitan comunicación a través de un medio de comunicación a otra entidad informática. El medio de comunicación transporta información tal como instrucciones ejecutables por ordenador, entrada o salida de audio o vídeo, u otros datos en una señal de datos modulada. Una señal de datos modulada es una señal que tiene una o más de sus características establecidas o cambiadas de tal manera para codificar información en la señal. A modo de ejemplo, y no como limitación, medio de comunicación incluye técnicas

alámbricas o inalámbricas implementadas con una portadora eléctrica, óptica, de RF, infrarrojos, acústica u otra.

Las técnicas y herramientas pueden describirse en el contexto general de medio legible por ordenador. Medio legible por ordenador puede ser cualquier medio disponible que pueda accederse en un entorno informático. A modo de ejemplo, y no como limitación, con el entorno 2000 informático, medio legible por ordenador incluye memoria 2020, almacenamiento 2040, medio de comunicación y combinaciones de cualquiera de lo anterior.

Las técnicas y herramientas pueden describirse en el contexto general de instrucciones ejecutables por ordenador, tal como aquellas incluidas en módulos de programa, que se ejecutan en un entorno informático en un procesador real o virtual. En general, los módulos de programa incluyen rutinas, programas, bibliotecas, objetos, clases, componentes, estructuras de datos, etc., que realizan tareas particulares o implementan tipos de datos abstractos particulares. La funcionalidad de los módulos de programa puede combinarse o dividirse entre módulos de programa según se desee en diversas realizaciones. Instrucciones ejecutables por ordenador para módulos de programa pueden ejecutarse en un entorno informático local o distribuido.

Para fines de presentación, la descripción detallada usa términos como “estimar”, “compensar”, “predecir” y “aplicar” para describir operaciones informáticas en un entorno informático. Estos términos son abstracciones de alto nivel para operaciones realizadas mediante un ordenador, y no deberían confundirse con actos realizados por un ser humano. Las operaciones informáticas reales que corresponden a estos términos varían dependiendo de la implementación.

II. Codificador y decodificador de vídeo generalizado

La Figura 21 es un diagrama de bloques de un codificador 2100 de vídeo generalizado en conjunto con el que pueden implementarse algunas realizaciones descritas. La Figura 22 es un diagrama de bloques de un decodificador 2200 de vídeo generalizado en conjunto con el que pueden implementarse algunas realizaciones descritas.

Las relaciones mostradas entre los módulos en el codificador 2100 y decodificador 2200 indican flujos generales de información en el codificador y decodificador; otras realizaciones no se muestran por motivos de simplicidad. En particular, las Figuras 21 y 22 normalmente no muestran información secundaria que indica los ajustes, modos, tablas, etc., del codificador usados para una secuencia, imagen, macrobloque, bloque, etc., de vídeo. Tal información secundaria se envía en el flujo de bits de salida, típicamente después de la codificación por entropía de la información secundaria. El formato del flujo de bits de salida puede ser un formato de Windows Media Video versión 9 u otro formato.

El codificador 2100 y el decodificador 2200 procesan imágenes de vídeo, que pueden ser fotogramas de vídeo, campos de vídeo o combinaciones de fotogramas y campos. La sintaxis y semántica de flujo de bits en los niveles de imagen y macrobloque puede depender de si se usan fotogramas o campos. Puede haber cambios a la organización de macrobloque y temporización global también. El codificador 2100 y el decodificador 2200 están basados en bloques y usan un formato de macrobloque 4:2:0 para fotogramas, incluyendo cada macrobloque cuatro bloques de luminancia de 8x8 (a veces tratados como un macrobloque de 16x16) y dos bloques de crominancia de 8x8. Para campos, puede usarse la misma o una organización y formato de macrobloques diferente. Los bloques de 8x8 pueden subdividirse adicionalmente en diferentes etapas, por ejemplo, en las etapas de transformación de frecuencia y codificación por entropía. Se describen organizaciones de fotograma de vídeo ejemplares en más detalle a continuación. Como alternativa, el codificador 2100 y el decodificador 2200 están basados en objeto, usan un formato de macrobloque o bloque diferente, o realizan operaciones en conjuntos de píxeles de tamaño o configuración diferente a los bloques de 8x8 y macrobloques de 16x16.

Dependiendo de la implementación y el tipo de compresión deseados, pueden añadirse módulos del codificador o decodificador, omitirse, dividirse en múltiples módulos, combinarse con otros módulos y/o sustituirse con módulos similares. En realizaciones alternativas, codificadores o decodificadores con diferentes módulos y/u otras configuraciones de módulos realizan una o más de las técnicas descritas.

A. Organizaciones de fotograma de vídeo

En algunas implementaciones, el codificador 2100 y el decodificador 2200 procesan fotogramas de vídeo organizados como sigue. Un fotograma contiene líneas de información espacial de una señal de vídeo. Para vídeo progresivo, estas líneas contienen muestras que comienzan desde un instante de tiempo y continúan a través de líneas sucesivas hasta la parte inferior del fotograma. Un fotograma de vídeo progresivo se divide en macrobloques tal como el macrobloque 2300 mostrado en la Figura 23. El macrobloque 2300 incluye cuatro bloques de luminancia de 8x8 (Y1 a Y4) y dos bloques de crominancia de 8x8 que están co-localizados con los cuatro bloques de luminancia pero a la mitad de resolución horizontal y verticalmente, siguiendo el formato de macrobloque 4:2:0 convencional. Los bloques de 8x8 pueden subdividirse adicionalmente en diferentes etapas, por ejemplo, en las etapas de transformación de frecuencia (por ejemplo, DCT de 8x4, 4x8 o 4x4) y codificación por entropía. Un fotograma I progresivo es un fotograma de vídeo progresivo intra-codificado. Un fotograma P progresivo es un fotograma de vídeo progresivo codificado usando predicción hacia delante, y un fotograma B progresivo es un fotograma de vídeo progresivo codificado usando predicción bidireccional. Los fotogramas P y B progresivos pueden incluir macrobloques intra-codificados así como diferentes tipos de macrobloques previstos.

Un fotograma de vídeo entrelazado consiste en dos exploraciones de un fotograma, una que comprende las líneas pares del fotograma (el campo superior) y la otra que comprende las líneas impares del fotograma (el campo inferior). Los dos campos pueden representar dos periodos de tiempo diferentes o pueden proceder del mismo periodo de tiempo. La Figura 24A muestra parte de un fotograma 2400 de vídeo entrelazado, que incluye líneas alternas del campo superior y campo inferior en la parte superior izquierda de parte del fotograma 2400 de vídeo entrelazado.

La Figura 24B muestra el fotograma 2400 de vídeo entrelazado de la Figura 24A organizado para codificar/decodificar como un fotograma 2430. El fotograma 2400 de vídeo entrelazado se ha particionado en macrobloques tal como los macrobloques 2431 y 2432, que usan un formato 4:2:0 como se muestra en la Figura 23. En el plano de luminancia, cada macrobloque 2431, 2432 incluye 8 líneas desde el campo superior que se alternan con 8 líneas desde el campo inferior para 16 líneas en total, y cada línea es de 16 píxeles de longitud. (La organización y colocación real de los bloques de luminancia y bloques de crominancia en los macrobloques 2431, 2432 no se muestra, y de hecho puede variar para diferentes decisiones de codificación). Dentro de un macrobloque dado, la información de campo superior y la información de campo inferior pueden codificarse conjuntamente o por separado en cualquiera de diversas fases. Un fotograma I entrelazado son dos campos intra-codificados de un fotograma de vídeo entrelazado, donde un macrobloque incluye información para los dos campos. Un fotograma P entrelazado son dos campos de un fotograma de vídeo entrelazado codificado usando predicción hacia delante, y un fotograma B entrelazado son dos campos de un fotograma de vídeo entrelazado codificado usando predicción bidireccional, donde un macrobloque incluye información para los dos campos. Los fotogramas P y fotogramas B pueden incluir macrobloques intra-codificados así como diferentes tipos de macrobloques previstos.

La Figura 24C muestra el fotograma 2400 de vídeo entrelazado de la Figura 24A organizado para codificar/decodificar como campos 2460. Cada uno de los dos campos del fotograma 2400 de vídeo entrelazado está particionado en macrobloques. El campo superior está particionado en macrobloques tal como el macrobloque 2461, y el campo inferior está particionado en macrobloques tal como el macrobloque 2462. (De nuevo, los macrobloques usan un formato 4:2:0 como se muestra en la Figura 23, y la organización y colocación de los bloques de luminancia y bloques de crominancia en los macrobloques no se muestra). En el plano de luminancia, el macrobloque 2461 incluye 16 líneas desde el campo superior y el macrobloque 2462 incluye 16 líneas desde el campo inferior, y cada línea es de 16 píxeles de longitud. Un campo I entrelazado es un único campo representado de manera separada de un fotograma de vídeo entrelazado. Un campo P entrelazado es un único campo representado de manera separada de un fotograma de vídeo entrelazado codificado usando predicción hacia delante, y un campo B entrelazado es un único campo representado de manera separada de un fotograma de vídeo entrelazado codificado usando predicción bidireccional. Los campos P y B entrelazados pueden incluir macrobloques intra-codificados así como diferentes tipos de macrobloques previstos.

El término imagen se refiere en general a datos de imagen de origen, codificados o reconstruidos. Para vídeo progresivo, una imagen es un fotograma de vídeo progresivo. Para vídeo entrelazado, una imagen puede hacer referencia a un fotograma de vídeo entrelazado, el campo superior del fotograma, o el campo inferior del fotograma, dependiendo del contexto.

Como alternativa, el codificador 2100 y el decodificador 2200 están basados en objeto, usan un formato de macrobloque o bloque diferente, o realizan operaciones en conjuntos de píxeles de tamaño o configuración diferente a los bloques de 8x8 y macrobloques de 16x16.

B. Codificador de vídeo

La Figura 21 es un diagrama de bloques de un sistema 2100 de codificador de vídeo generalizado. El sistema 2100 de codificador recibe una secuencia de imágenes de vídeo que incluyen una imagen 2105 actual (por ejemplo, fotograma de vídeo progresivo, fotograma de vídeo entrelazado o campo de un fotograma de vídeo entrelazado), y produce información 2195 de vídeo comprimida como salida. Las realizaciones particulares de los codificadores de vídeo típicamente usan una variación o versión complementada del codificador 2100 generalizado.

El sistema 2100 de codificador comprime imágenes previstas e imágenes clave. Para fines de presentación, la Figura 21 muestra una trayectoria para imágenes clave a través del sistema 2100 de codificador y una trayectoria para imágenes previstas. Muchos de los componentes del sistema 2100 de codificador se usan para comprimir tanto imágenes clave como imágenes previstas. Las operaciones exactas realizadas mediante estos componentes pueden variar dependiendo del tipo de información que se esté comprimiendo.

Una imagen prevista (por ejemplo, imagen p o imagen b progresiva, campo p o campo b entrelazado, o fotograma p o fotograma b entrelazado) se representa en términos de predicción (o diferencia) desde una o más otras imágenes (que se denominan típicamente como imágenes o anclas de referencia). Una predicción residual es la diferencia entre lo que se predice y la imagen original. En contraste, una imagen clave (por ejemplo, fotograma 1 progresivo, campo I entrelazado o fotograma I entrelazado) se comprime sin referencia a otras imágenes.

Si la imagen 2105 actual es una imagen prevista hacia delante, un estimador 2110 de movimiento estima el movimiento de macrobloques u otros conjuntos de píxeles de la imagen 2105 actual con respecto a una o más

imágenes de referencia, por ejemplo, la imagen 2125 previa reconstruida almacenada en memoria intermedia en el almacenamiento 2120 de imágenes. Si la imagen 2105 actual es una imagen bidireccionalmente prevista, un estimador 2110 de movimiento estima el movimiento en la imagen 2105 actual con respecto a hasta cuatro imágenes de referencia reconstruidas (para un campo B entrelazado, por ejemplo). Típicamente, un estimador de movimiento estima el movimiento en una imagen B con respecto a una o más imágenes de referencia temporalmente anteriores y una o más imágenes de referencia temporalmente futuras. Por consiguiente, el sistema 2100 de codificador puede usar los almacenamientos 2120 y 2122 separados para múltiples imágenes de referencia. Para más información sobre fotogramas B progresivos, véase la Solicitud de Patente de Estados Unidos con N.º de Serie 10/622.378, titulada, "Advanced Bi-Directional Predictive Coding of Video Frames", presentada el 18 de julio de 2003.

El estimador 2110 de movimiento puede estimar movimiento en incrementos de píxel, $\frac{1}{2}$ píxel, $\frac{1}{4}$ píxel u otros incrementos, y puede conmutar la resolución de la estimación de movimiento en una base imagen a imagen u otra base. El estimador 2110 de movimiento (y el compensador 2130) también pueden conmutar entre tipos de interpolación de píxel de imagen de referencia (por ejemplo, entre bicúbica y bilinear) en una base por fotograma u otra. La resolución de la estimación de movimiento puede ser la misma o diferente horizontal y verticalmente. El estimador 2110 de movimiento emite como información secundaria información 2115 de movimiento tal como información de vector de movimiento diferencial. El codificador 2100 codifica la información 2115 de movimiento calculando, por ejemplo, uno o más predictores para vectores de movimiento, calculando diferenciales entre los vectores de movimiento y predictores, y codificando por entropía los diferenciales. Para reconstruir un vector de movimiento, un compensador 2130 de movimiento combina un predictor con información de vector de movimiento diferencial. Se describen a continuación diversas técnicas para calcular predictores de vector de movimiento, calcular vectores de movimiento diferencial y reconstruir vectores de movimiento para campos B entrelazados y fotogramas B entrelazados.

El compensador 2130 de movimiento aplica el vector de movimiento reconstruido a la imagen o imágenes 2125 reconstruidas para formar una imagen 2135 actual de movimiento compensado. Sin embargo, la predicción raramente es perfecta, y la diferencia entre la imagen 2135 actual de movimiento compensado y la imagen 2105 actual original es la predicción residual 2145. Durante la última reconstrucción de la imagen, la predicción residual 2145 se añade a la imagen 2135 actual de movimiento compensado para obtener una imagen reconstruida que está más cerca de la imagen 2105 actual original. En compresión con pérdidas, sin embargo, aún se pierde alguna información de la imagen 2105 actual original. Como alternativa, un estimador de movimiento y compensador de movimiento aplica otro tipo de estimación/compensación de movimiento.

Un transformador 2160 de frecuencia convierte la información de vídeo de dominio espacial en datos de dominio de la frecuencia (es decir, espectral). Para imágenes de vídeo basadas en bloques, el transformador 2160 de frecuencia aplica una DCT, variante de DCT u otra transformación de bloque para transformar bloques de los datos de píxel o datos de predicción residual, produciendo bloques de coeficientes de transformación de frecuencia. Como alternativa, el transformador 2160 de frecuencia aplica otra transformación de frecuencia convencional tal como una transformada de Fourier o usa análisis de ondícula o de subbanda. El transformador 2160 de frecuencia puede aplicar una transformación de frecuencia de tamaño de 8x8, 8x4, 4x8, 4x4 u otro.

Un cuantificador 2170 a continuación cuantifica los bloques de coeficientes de datos espectrales. El cuantificador aplica cuantificación escalar uniforme a los datos espectrales con un tamaño gradual que varía en una base imagen a imagen u otra base. Como alternativa, el cuantificador aplica otro tipo de cuantificación a los coeficientes de datos espectrales, por ejemplo, una cuantificación no uniforme, vectorial o no adaptativa, o directamente cuantifica datos de dominio espacial en un sistema codificador que no usa transformaciones de frecuencia. Además de cuantificación adaptativa, el codificador 2100 puede usar pérdida de fotogramas, filtración adaptativa u otras técnicas para control de tasa.

El codificador 2100 puede usar señalización especial para un macrobloque saltado, que es un macrobloque que no tiene información de ciertos tipos (por ejemplo, ninguna información de movimiento para el macrobloque y ninguna información residual).

Cuando una imagen actual reconstruida es necesaria para estimación/compensación de movimiento posterior, un cuantificador 2176 inverso realiza cuantificación inversa en los coeficientes de datos espectrales cuantificados. Un transformador 2166 de frecuencia inverso a continuación realiza la inversa de las operaciones del transformador 2160 de frecuencia, produciendo una predicción reconstruida residual (para una imagen prevista) o una imagen clave reconstruida. Si la imagen 2105 actual era una imagen de clave, la imagen de clave reconstruida se toma como la imagen actual reconstruida (no mostrada). Si la imagen 2105 actual era una imagen prevista, la predicción reconstruida residual se añade a la imagen 2135 actual de movimiento compensado para formar la imagen actual reconstruida. Uno o ambos de los almacenamientos 2120, 2122 de imagen almacenan en memoria intermedia la imagen actual reconstruida para uso en predicción de movimiento compensado. En algunas realizaciones, el codificador aplica un filtro de desbloqueo al fotograma reconstruido para suavizar de manera adaptativa discontinuidades y otros artefactos en la imagen.

El codificador 2180 por entropía comprime la salida del cuantificador 2170 así como cierta información secundaria

(por ejemplo, información 2115 de movimiento, tamaño de etapa de cuantificación). Las técnicas de codificación por entropía típicas incluyen codificación aritmética, codificación diferencial, codificación Huffman, codificación por longitud de serie, codificación LZ, codificación de diccionario y combinaciones de las anteriores. El codificador 2180 por entropía típicamente usa diferentes técnicas de codificación para diferentes clases de información (por ejemplo, coeficientes de DC, coeficientes de AC, diferentes clases de información secundaria), y puede elegir de entre múltiples tablas de código dentro de una técnica de codificación particular.

El codificador 2180 por entropía proporciona información 2195 de vídeo comprimida al multiplexor ["MUX"] 2190. El MUX 2190 puede incluir una memoria intermedia, y un indicador de nivel de memoria intermedia puede realimentarse a los módulos adaptativos de tasa de bits para control de tasas. Antes o después del MUX 2190, la información 2195 de vídeo comprimida puede codificarse por canal para transmisión a través de la red. La codificación de canal puede aplicar datos de detección y corrección de errores a la información 2195 de vídeo comprimida.

C. Decodificador de vídeo

La Figura 22 es un diagrama de bloques de un sistema 2200 de decodificador de vídeo general. El sistema 2200 de decodificador recibe información 2295 para una secuencia comprimida de imágenes de vídeo y produce salida que incluye una imagen 2205 reconstruida (por ejemplo, fotograma de vídeo progresivo, fotograma de vídeo entrelazado o campo de un fotograma de vídeo entrelazado). Realizaciones particulares de decodificadores de vídeo típicamente usan una variación o versión complementada del decodificador 2200 generalizado.

El sistema 2200 de decodificador descomprime imágenes previstas e imágenes clave. Para fines de presentación, la Figura 22 muestra una trayectoria para imágenes clave a través del sistema 2200 de decodificador y una trayectoria para imágenes previstas hacia delante. Muchos de los componentes del sistema 2200 de decodificador se usan para descomprimir tanto imágenes clave como imágenes previstas. Las operaciones exactas realizadas por estos componentes pueden variar dependiendo del tipo de información que se esté descomprimiendo.

Un DEMUX 2290 recibe la información 2295 para la secuencia de vídeo comprimido y pone a disposición la información recibida al decodificador 2280 por entropía. El DEMUX 2290 puede incluir una memoria intermedia de fluctuación y otras memorias intermedias también. Antes o después del DEMUX 2290, la información de vídeo comprimida puede codificarse por canal y procesarse para detección y corrección de errores.

El decodificador 2280 por entropía decodifica por entropía datos cuantificados codificados por entropía así como información secundaria codificada por entropía (por ejemplo, información 2215 de movimiento, tamaño de etapa de cuantificación), aplicando típicamente la inversa de la codificación por entropía realizada en el codificador. Las técnicas de decodificación por entropía incluyen decodificación aritmética, decodificación diferencial, decodificación Huffman, decodificación por longitud de serie, decodificación LZ, decodificación de diccionario y combinaciones de las anteriores. El decodificador 2280 por entropía típicamente usa diferentes técnicas de decodificación para diferentes tipos de información (por ejemplo, coeficientes de DC, coeficientes de AC, diferentes tipos de información secundaria), y puede elegir de entre múltiples tablas de código dentro de una técnica de decodificación particular.

El decodificador 2200 decodifica la información 2215 de movimiento calculando, por ejemplo, uno o más predictores para vectores de movimiento, decodificación por entropía vectores de movimiento diferencial, y combinando vectores de movimiento diferencial decodificados con predictores para reconstruir vectores de movimiento. Se describen a continuación diversas técnicas para calcular predictores de vector de movimiento, calcular vectores de movimiento diferencial y reconstruir vectores de movimiento para campos B entrelazados y fotogramas B entrelazados.

Un compensador 2230 de movimiento aplica información 2215 de movimiento a una o más imágenes 2225 de referencia para formar una predicción 2235 de la imagen 2205 que se está reconstruyendo. Por ejemplo, el compensador 2230 de movimiento usa uno o más vectores de movimiento de macrobloque para hallar el macrobloque o macrobloques en la imagen o imágenes 2225 de referencia. Uno o más almacenamientos de imágenes (por ejemplo, los almacenamientos 2220, 2222 de imágenes) almacenan imágenes reconstruidas anteriores para uso como imágenes de referencia. Típicamente, las imágenes B tienen más de una imagen de referencia (por ejemplo, al menos una imagen de referencia temporalmente anterior y al menos una imagen de referencia temporalmente futura). Por consiguiente, el sistema 2200 de decodificador puede usar almacenamientos 2220 y 2222 de imágenes separados para múltiples imágenes de referencia. El compensador 2230 de movimiento puede compensar movimiento a incrementos de píxel, $\frac{1}{2}$ píxel, $\frac{1}{4}$ píxel u otros incrementos, y puede conmutar la resolución de la compensación de movimiento en una base imagen a imagen u otra. El compensador 2230 de movimiento también puede conmutar entre tipos de interpolación de píxel de imagen de referencia (por ejemplo, entre bicúbica y bilinear) en una base por fotograma u otra. La resolución de la compensación de movimiento puede ser la misma o diferente horizontal y verticalmente. Como alternativa, un compensador de movimiento aplica otro tipo de compensación de movimiento. La predicción mediante el compensador de movimiento raramente es perfecta, por lo que el decodificador 2200 también reconstruye residuos de predicción.

Un cuantificador 2270 inverso cuantifica a la inversa datos decodificados por entropía. En general, el cuantificador inverso aplica cuantificación inversa escalar uniforme a los datos decodificados por entropía con un tamaño gradual

que varía en una base imagen a imagen u otra. Como alternativa, el cuantificador inverso aplica otro tipo de cuantificación inversa a los datos, por ejemplo, para reconstruir después de una cuantificación no uniforme, vectorial o no adaptativa, o cuantifica a la inversa directamente datos de dominio espacial en un sistema de decodificador que no usa transformaciones de frecuencia inversa.

- 5 Un transformador 2260 de frecuencia inverso convierte los datos de dominio de frecuencia cuantificados en información de vídeo de dominio espacial. Para imágenes de vídeo basadas en bloques, el transformador 2260 de frecuencia inverso aplica una DCT inversa ["IDCT"], variante de IDCT, u otra transformación de bloque inversa a bloques de los coeficientes de transformación de frecuencia, produciendo datos de píxeles o datos de predicción residual para imágenes clave o imágenes previstas, respectivamente. Como alternativa, el transformador 2260 de frecuencia inverso aplica otra transformación de frecuencia inversa convencional tal como una transformada de Fourier inversa o usa síntesis de ondícula o de subbanda. El transformador 2260 de frecuencia inverso puede aplicar una transformación de frecuencia inversa de 8x8, 8x4, 4x8, 4x4 u otro tamaño.

- 15 Para una imagen prevista, el decodificador 2200 combina la predicción reconstruida residual 2245 con la predicción 2235 de movimiento compensado para formar la imagen 2205 reconstruida. Cuando el decodificador necesita una imagen 2205 reconstruida para compensación de movimiento posterior, uno o ambos de los almacenamientos de la imagen (por ejemplo, el almacenamiento 2220 de imagen) almacenan en memoria intermedia la imagen 2205 reconstruida para uso al predecir la siguiente imagen. En algunas realizaciones, el decodificador 2200 aplica un filtro de desbloqueo a la imagen reconstruida para suavizar de manera adaptativa discontinuidades y otros artefactos en la imagen.

20 **III. Campos P entrelazados y fotogramas P entrelazados**

- Un fotograma de vídeo entrelazado típico consiste en dos campos (por ejemplo, un campo superior y un campo inferior) explorados en diferentes tiempos. En general, es más eficaz codificar regiones estáticas de un fotograma de vídeo entrelazado codificando campos juntos (codificación de "modo de fotograma"). Por otra parte, a menudo es más eficaz codificar regiones en movimiento de un fotograma de vídeo entrelazado codificando campos de manera separada (codificación de "modo de campo"), puesto que los dos campos tienden a tener diferente movimiento. Un fotograma de vídeo entrelazado previsto hacia delante puede codificarse como dos campos previstos hacia delante separados, campos P entrelazados. Codificar campos de manera separada para un fotograma de vídeo entrelazado previsto hacia delante puede ser eficaz, por ejemplo, cuando hay elevado movimiento a través de todo el fotograma de vídeo entrelazado, y por lo tanto mucha diferencia entre los campos.

- 30 O, un fotograma de vídeo entrelazado previsto hacia delante puede codificarse usando una mezcla de codificación de campo y codificación de fotograma, como un fotograma P entrelazado. Para un macrobloque de un fotograma P entrelazado, el macrobloque incluye líneas de píxeles para los campos superior e inferior, y las líneas pueden codificarse de manera colectiva en un modo de codificación de fotograma o de manera separada en un modo de codificación de campo.

35 **A. Campos P entrelazados**

- Un campo P entrelazado hace referencia a uno o más campos previamente decodificados. Por ejemplo, en algunas implementaciones, un campo P entrelazado hace referencia a cualquiera de uno o dos campos previamente decodificados, mientras que los campos B entrelazados hacen referencia a hasta dos campos de referencia anteriores y dos futuros (es decir, hasta un total de cuatro campos de referencia). (Se describen en detalle a continuación técnicas de codificación y decodificación para campos B entrelazados).

- 45 Las Figuras 25 y 26 muestran ejemplos de campos P entrelazados que tienen dos campos de referencia. En la Figura 25, el campo 2510 actual hace referencia a un campo 2520 superior y campo 2530 inferior en un fotograma de vídeo entrelazado temporalmente anterior. Puesto que los campos 2540 y 2550 son campos B entrelazados, no se usan como campos de referencia. En la Figura 26, el campo 2610 actual hace referencia a un campo 2620 superior y al campo 2630 inferior en un fotograma de vídeo entrelazado inmediatamente anterior al fotograma de vídeo entrelazado que contiene el campo 2610 actual. Para más información sobre campos P entrelazados de dos referencias, véase la Solicitud de Patente de Estados Unidos con N.º de Serie xx/yyy.zzz, titulada, "Predicting Motion Vectors for Fields of Forward-predicted Interlaced Video Frames", presentada el 27 de mayo de 2004.

- 50 Las Figuras 27 y 28 muestran ejemplos de campos P entrelazados que tienen un campo de referencia, el campo de referencia permisible temporalmente más reciente. En la Figura 27, el campo 2710 actual hace referencia a un campo 2730 inferior en un fotograma de vídeo entrelazado temporalmente anterior, pero no hace referencia al campo 2720 superior menos reciente en el fotograma de vídeo entrelazado. En el ejemplo mostrado en la Figura 27, los campos 2740 y 2750 son campos B entrelazados y no son campos de referencia permisibles. En la Figura 28, el campo 2810 actual hace referencia al campo 2830 inferior en un fotograma de vídeo entrelazado inmediatamente anterior al fotograma de vídeo entrelazado que contiene el campo 2810 actual, en lugar del campo 2820 superior menos reciente.

Las Figuras 29 y 30 muestran ejemplos de campos P entrelazados que usan el segundo campo de referencia permisible más reciente. En la Figura 29, el campo 2910 actual hace referencia a un campo 2920 superior en un

fotograma de vídeo entrelazado temporalmente anterior, pero no hace referencia al campo 2930 inferior más reciente. En el ejemplo mostrado en la Figura 29, los campos 2940 y 2950 son campos B entrelazados y no son campos de referencia permisibles. En la Figura 30, el campo 3010 actual hace referencia al campo 3020 superior en lugar de al campo 3030 inferior más reciente.

- 5 En una implementación, todos los escenarios mostrados en las Figuras 25-30 están permitidos en una sintaxis de campo P entrelazado. Son posibles otras implementaciones. Por ejemplo, las imágenes pueden usar campos de otras imágenes de diferentes tipos o posiciones temporales como campos de referencia.

1. Sistema de coordenadas de imagen de campo y polaridades de campo

- 10 Los vectores de movimiento representan desplazamientos horizontales y verticales en unidades de cuarto de píxel. Por ejemplo, si el componente vertical de un vector de movimiento indica un desplazamiento de 6 unidades de cuarto de píxel, esto indica que el bloque de referencia está una línea y media de campo por debajo de la posición real del bloque ($6 * \frac{1}{4} = 1 \frac{1}{2}$).

- 15 La Figura 31 muestra una relación entre componentes verticales de vectores de movimiento y localizaciones espaciales en una implementación. El ejemplo mostrado en la Figura 31 muestra tres diferentes escenarios 3110, 3120 y 3130 para tres combinaciones diferentes de tipos de campo actual y de referencia (por ejemplo, superior e inferior). Si los tipos de campo son diferentes para los campos de referencia y actual, la polaridad es "opuesta". Si los tipos de campo son el mismo, la polaridad es la "misma". Para cada escenario, la Figura 31 muestra una columna vertical de píxeles en un campo actual y una segunda columna vertical de píxeles en un campo de referencia. En realidad, las dos columnas están alienadas horizontalmente. Un círculo representa una posición de píxel de número entero real y una X representa una posición de medio o cuarto de píxel interpolado. Los valores de componente horizontal (no mostrados) no necesitan contabilizarse para ningún desplazamiento debido al entrelazado, ya que los respectivos campos están alineados horizontalmente. Los valores negativos indican desplazamientos más hacia arriba, y en la dirección opuesta, como lo muestran los desplazamientos verticales de valor positivo.

- 25 En el escenario 3110, la polaridad es "opuesta". El campo actual es un campo superior y el campo de referencia es un campo inferior. Con relación al campo actual, la posición del campo de referencia está desplazada por medio píxel en la dirección hacia abajo debido al entrelazado. Un valor de componente de vector de movimiento vertical de 0 es el desplazamiento de "ningún movimiento vertical" y representa una posición en el campo de referencia en el mismo nivel vertical (en términos absolutos) que la localización en el campo de referencia, un valor de componente de vector de movimiento vertical de +2 representa una posición en el campo de referencia que está desplazada medio píxel (en términos absolutos) por debajo de la localización en el campo actual, que es un valor real en el campo de referencia, y un valor de componente vertical de +4 representa un desplazamiento de posición de un píxel completo (en términos absolutos) por debajo de la localización en el campo actual, que es un valor interpolado en el campo de referencia.

- 35 En el escenario 3120, la polaridad es también "opuesta." El campo actual es un campo inferior y el campo de referencia es un campo superior. Con relación al campo actual, la posición del campo de referencia está desplazada por medio píxel en la dirección hacia arriba debido al entrelazado. Un componente de vector de movimiento vertical de -2 representa una posición en el campo de referencia que está medio píxel (en términos absolutos) por encima de la localización en el campo actual, un valor de componente vertical de 0 representa una posición al mismo nivel (en términos absolutos) que la localización en el campo actual, y un componente vertical de +2 representa un desplazamiento de posición de medio píxel por debajo (en términos absolutos) de la localización en el campo actual.

- 40 En el escenario 3130, la polaridad es la "misma". Con relación al campo actual, la posición del campo de referencia es la misma en la dirección vertical. Un valor de componente de vector de movimiento vertical de 0 es el desplazamiento de "ningún movimiento vertical" y representa una posición en el campo de referencia en el mismo nivel vertical (en términos absolutos) que la localización en el campo de referencia, un valor de componente de vector de movimiento vertical de +2 representa una posición en el campo de referencia que está desplazada por medio píxel (en términos absolutos) por debajo de la localización en el campo actual, que es un valor interpolado en el campo de referencia, y un valor de componente vertical de +4 representa un desplazamiento de posición en un píxel completo (en términos absolutos) por debajo de la localización en el campo actual, que es un valor real en el campo de referencia.

Como alternativa, los desplazamientos para vectores de movimiento se expresan de acuerdo con una convención diferente.

2. Predicción de vector de movimiento en campos P entrelazados de campo de dos referencias

- 55 Los campos P entrelazados de campo de dos referencias hacen referencia a dos campos en la misma dirección temporal (por ejemplo, los dos campos de referencia anteriores más recientes). Se calculan dos predictores de vector de movimiento para cada macrobloque. En algunas implementaciones, un predictor proviene de un campo de referencia de la misma polaridad, y el otro proviene de un campo de referencia de polaridad opuesta. Son posibles también otras combinaciones de polaridades. (Se describen a continuación campos B entrelazados que usan

campos de dos referencias por dirección. En algunas implementaciones, tales campos B entrelazados usan las mismas técnicas que los campos P entrelazados para calcular predictores de vector de movimiento).

En algunas realizaciones, un codificador/decodificador calcula un predictor de vector de movimiento para un bloque o macrobloque actual hallando un predictor de campo impar y un predictor de campo par, y seleccionando uno de los predictores para procesar el macrobloque. Por ejemplo, un codificador/decodificador determina un predictor de vector de movimiento de campo impar y predictor de vector de movimiento de campo par. Uno de los predictores de vector de movimiento por lo tanto tiene la misma polaridad que el campo actual, y el otro predictor de vector de movimiento tiene la polaridad opuesta. El codificador/decodificador selecciona un predictor de vector de movimiento de entre el predictor de vector de movimiento de campo impar y el predictor de vector de movimiento de campo par. Por ejemplo, el codificador selecciona entre los predictores de vector de movimiento basándose en el que proporciona mejor predicción. El codificador señala qué predictor de vector de movimiento usar usando una señal de selección sencilla o usando señalización más compleja que incorpora información contextual para mejorar la eficacia de codificación. La información contextual puede indicar cuál del campo impar o el campo par, o cuál del campo de la misma polaridad o campo de polaridad opuesta, se ha usado predominantemente en las cercanías alrededor del bloque o macrobloque. El decodificador selecciona qué predictor de vector de movimiento usar basándose en la señal de selección y/o la información contextual. A continuación, el codificador/decodificador procesa el vector de movimiento usando el predictor de vector de movimiento seleccionado. Por ejemplo, el codificador codifica un diferencial entre el vector de movimiento y el predictor de vector de movimiento. O, el decodificador decodifica el vector de movimiento combinando el vector de movimiento diferencial y el predictor de vector de movimiento.

Como alternativa, el codificador y/o el decodificador puede saltar la determinación del predictor de vector de movimiento de campo impar o determinar el predictor de vector de movimiento de campo par. Por ejemplo, si el codificador determina que el campo impar se usará para compensación de movimiento para un bloque o macrobloque particular, el codificador determina únicamente el predictor de vector de movimiento de campo impar. O, si el decodificador determina a partir de información contextual y/o señalizada que se usará el campo impar para compensación de movimiento, el decodificador determina únicamente el predictor de vector de movimiento de campo impar. De esta manera, el codificador y el decodificador pueden evitar operaciones innecesarias.

Un decodificador puede emplear la siguiente técnica para determinar predictores de vector de movimiento para un campo P entrelazado actual:

Para cada bloque o macrobloque con un vector de movimiento en un campo P entrelazado, se obtienen dos conjuntos de tres predictores de vector de movimiento candidatos. Las posiciones de los macrobloques vecinos a partir de los cuales se obtienen estos predictores de vector de movimiento candidato con relación a un macrobloque 3200 actual se muestran en la Figura 32. Tres de los candidatos provienen del campo de referencia par y tres provienen del campo de referencia impar. Puesto que los macrobloques vecinos en cada dirección candidata (A, B o C) estarán intra-codificados o tendrán un vector de movimiento real que hace referencia a cualquiera del campo par o el campo impar, existe una necesidad de derivar el vector de movimiento del otro campo (o derivar ambos candidatos de vector de movimiento de tanto el campo impar como el campo para un macrobloque intra-codificado). Por ejemplo, para un macrobloque dado, supóngase que el predictor A tiene un vector de movimiento que hace referencia al campo impar. En este caso, el candidato de predictor A de “campo par” se deriva desde el vector de movimiento del candidato de predictor A de “campo impar”. Esta derivación se consigue usando una operación de cambio de escala. (Véase, por ejemplo, la explicación de las Figuras 34A y 34B a continuación). Como alternativa, la derivación se consigue de otra manera.

Una vez que se han obtenido los tres predictores de vector de movimiento candidatos de campo impar, se usa una operación de mediana para derivar un predictor de vector de movimiento de campo impar desde los tres candidatos de campo impar. De manera similar, una vez que se han obtenido los tres predictores de vector de movimiento candidatos de campo par, se usa una operación de mediana para derivar un predictor de vector de movimiento de campo par a partir de los tres candidatos de campo par. Como alternativa, se usa otro mecanismo para seleccionar el predictor de vector de movimiento de campo basándose en los predictores de vector de movimiento de campo candidatos. El decodificador decide si usar el campo par o el campo impar como el predictor de vector de movimiento candidato (por ejemplo, seleccionando el predictor dominante), y se usa el predictor de vector de movimiento impar o par para reconstruir el vector de movimiento.

El pseudocódigo 3300 en las Figuras 33A-33F ilustra un procedimiento usado para generar predictores de vector de movimiento desde los predictores A, B y C como están dispuestos en la Figura 32. Mientras que la Figura 32 muestra unas cercanías para un macrobloque típico en el medio del campo P entrelazado actual, el pseudocódigo 3300 de las Figuras 33A-33F trata diversos casos especiales para las localizaciones de macrobloque. Además, el pseudocódigo 3300 puede usarse para calcular un predictor de vector de movimiento para el vector de movimiento de un bloque en diversas localizaciones.

En el pseudocódigo 3300, los términos “mismo campo” y “campo opuesto” se han de entender con relación al campo que se está codificando o decodificando actualmente. Si el campo actual es un campo par, por ejemplo, el “mismo campo” es el campo de referencia y el “campo opuesto” es el campo de referencia impar. Las variables

samefieldpred_x y *samefieldpred_y* en el pseudocódigo 3300 representan los componentes horizontales y verticales del predictor de vector de movimiento desde el mismo campo, y las variables *oppositefieldpred_x* y *oppositefieldpred_y* representan los componentes horizontales y verticales del predictor de vector de movimiento desde el campo opuesto. Las variables *samecount* y *oppositecount* rastrean cuántos de los vectores de movimiento para los vecinos del bloque o macrobloque actual hacen referencia al campo de referencia de la “misma” polaridad para el campo actual y cuántos hacen referencia al campo de referencia de polaridad “opuesta”, respectivamente. Las variables *samecount* y *oppositecount* se inicializan a 0 en el comienzo del pseudocódigo.

Las operaciones de cambio de escala *scaleforsame()* y *scaleforopposite()* mencionadas en el pseudocódigo 3300 se usan para derivar candidatos de predictor de vector de movimiento para el “otro” campo desde los valores de vector de movimiento reales de los vecinos. Las operaciones de cambio de escala son dependientes de la implementación. Se describen a continuación operaciones de cambio de escala con referencia a las Figuras 34A, 34B, 35 y 36. Como alternativa, se usan otras operaciones de cambio de escala, por ejemplo, para compensar desplazamientos verticales tales como aquellos mostrados en la Figura 31. (Se describen en detalle a continuación operaciones de cambio de escala específicamente usadas para campos B entrelazados).

Las Figuras 33A y 33B muestran pseudocódigo para calcular un predictor de vector de movimiento para un bloque o macrobloque típico en una posición interior dentro de un fotograma. Los vectores de movimiento para “intra” vecinos se establecen a 0. Para cada vecino, se establecen el predictor de vector de movimiento de mismo campo y el predictor de vector de movimiento de campo opuesto, donde uno se establece desde el valor real del vector de movimiento para el vecino y el otro se deriva a desde el mismo. La mediana de los candidatos se calcula para el predictor de vector de movimiento de mismo campo y para el predictor de vector de movimiento de campo opuesto, y el predictor “dominante” se determina desde *samecount* y *oppositecount*. La variable *dominantpredictor* indica qué campo contiene el predictor de vector de movimiento dominante. Un predictor de vector de movimiento es dominante si tiene la misma polaridad que la mayoría de los tres predictores candidatos. (El valor señalado *predictor_flag*, que se decodifica junto con los datos de vector de movimiento diferencial, indican si se usa el predictor dominante o no dominante).

El pseudocódigo en la Figura 33C trata la situación de un macrobloque en un campo P entrelazado con únicamente un macrobloque por fila, para el que no hay vecinos B o C. El pseudocódigo en las Figuras 33D y 33E trata la situación de un bloque o macrobloque en el borde izquierdo de un campo P entrelazado, para el que no hay vecino C. En este punto, un predictor de vector de movimiento es dominante si tiene la misma polaridad que más de los dos predictores candidatos, siendo el predictor de vector de movimiento de campo opuesto dominante en el caso de un empate. Finalmente, el pseudocódigo en la Figura 33F trata, por ejemplo, los casos de macrobloque en la fila superior de un campo P entrelazado.

3. Cambio de escala para derivación de predictor de vector de movimiento de un campo desde otro predictor de vector de movimiento de campo

En una implementación, un codificador/decodificador deriva un predictor de vector de movimiento de campo desde otro predictor de vector de movimiento de campo usando la operación de cambio de escala ilustrada en el pseudocódigo 3400 de las Figuras 34A y 34B. Los valores de *SCALEOPP*, *SCALESAME1*, *SCALESAME2*, *SCALEZONE1_X*, *SCALEZONE1_Y*, *ZONE1OFFSET_X* y *ZONE1OFFSET_Y* son dependientes de la implementación. Se muestran dos posibles conjuntos de valores en la tabla 3500 en la Figura 35 para el caso donde el campo actual es el primer campo en el fotograma de vídeo entrelazado, y en la tabla 3600 en la Figura 36 para el caso donde el campo actual es el segundo campo en el fotograma de vídeo entrelazado. Para fotogramas P, la distancia de fotograma de referencia se define como el número de fotogramas B (es decir, un fotograma de vídeo que contiene dos campos B) entre el fotograma P actual y su fotograma de referencia. Si no están presentes fotogramas B, la distancia de referencia es 0. Por ejemplo, un codificador usa un elemento de sintaxis con tamaño variable (por ejemplo, el elemento de sintaxis *REFDIST*, que se describe en detalle en la sección XIV a continuación) para codificar la distancia de fotograma de referencia.

En los ejemplos mostrados en las tablas 3500 y 3600, el valor de N (un multiplicador para los valores *SCALEZONE1_X*, *SCALEZONE1_Y*, *ZONE1OFFSET_X* y *ZONE1OFFSET_Y* en las tablas) depende de un intervalo de vector de movimiento. Por ejemplo, un intervalo de vector de movimiento extendido puede señalizarse mediante el elemento de sintaxis *EXTENDED_MV* = 1. Si *EXTENDED_MV* = 1, el elemento de sintaxis *MVRANGE* está presente en el encabezamiento de la imagen y señala el intervalo del vector de movimiento. Si *EXTENDED_MV* = 0 entonces se usa un intervalo de vector de movimiento por defecto. La Tabla 1 a continuación muestra la relación entre N y *MVRANGE*.

Tabla 1: Derivación de N en Figuras 35 y 36

MVRANGE	N
0 o por defecto	1
10	2
110	8

(continuación)

MVRANGE	N
111	16

Los valores mostrados en la tablas 3500 y 3600 pueden modificarse dependiendo de la implementación.

Como alternativa, N se supone que es 1 (es decir, el cambio de escala no depende de N), o el cambio de escala puede realizarse de alguna otra manera.

B. Fotogramas P entrelazados

En algunas implementaciones, los macrobloques en fotogramas P entrelazados pueden ser uno de cinco tipos: 1MV, MV de 2 campos, MV de 4 fotogramas, MV de 4 campos e intra.

En un macrobloque de 1 MV, el desplazamiento de los cuatro bloques de luminancia en el macrobloque se representa mediante un único vector de movimiento. Un correspondiente vector de movimiento de crominancia puede derivarse desde el vector de movimiento de luminancia para representar los desplazamientos de cada uno de los dos bloques de crominancia de 8x8 para el vector de movimiento. Por ejemplo, haciendo referencia de nuevo a la disposición de macrobloque mostrada en la Figura 23, un macrobloque 2300 de 1 MV incluye cuatro bloques de luminancia de 8x8 y dos bloques de crominancia de 8x8. El desplazamiento de los bloques de luminancia (Y1 a Y4) se representa mediante un único vector de movimiento, y puede derivarse un correspondiente vector de movimiento de crominancia desde el vector de movimiento de luminancia para representar los desplazamientos de cada uno de los dos bloques de crominancia (U y V).

En un macrobloque de MV de 2 campos, el desplazamiento de cada campo para los cuatro bloques de luminancia en el macrobloque se describe mediante un vector de movimiento diferente. Por ejemplo, la Figura 37 muestra que un vector de movimiento de campo superior describe el desplazamiento de las líneas pares de todos los cuatro bloques de luminancia y que un vector de movimiento de campo inferior describe el desplazamiento de las líneas impares de todos los cuatro bloques de luminancia. Usando el vector de movimiento de campo superior, un codificador puede derivar un correspondiente vector de movimiento de crominancia de campo superior que describe el desplazamiento de las líneas pares de los bloques de crominancia. De manera similar, un codificador puede derivar un vector de movimiento de crominancia de campo inferior que describe los desplazamientos de las líneas impares de los bloques de crominancia.

Haciendo referencia a la Figura 38, en un macrobloque de MV de 4 fotogramas, el desplazamiento de cada uno de los cuatro bloques de luminancia se describe mediante un vector de movimiento diferente (MV1, MV2, MV3 y MV4). Cada bloque de crominancia puede compensarse en movimiento usando cuatro vectores de movimiento de crominancia derivados (MV1', MV2', MV3' y MV4') que describen el desplazamiento de cuatro sub-bloques de crominancia de 4x4. Un vector de movimiento para cada sub-bloque de crominancia de 4x4 puede derivarse desde el vector de movimiento para el bloque de luminancia espacialmente correspondiente.

Haciendo referencia a la Figura 39, en un macrobloque de MV de 4 campos, el desplazamiento de cada campo en los bloques de luminancia se describe mediante dos vectores de movimiento diferentes. Las líneas pares de los bloques de luminancia se subdividen verticalmente para formar dos regiones de 8x8. Para las líneas pares, el desplazamiento de la región izquierda se describe mediante el vector de movimiento de bloque de campo superior izquierdo y el desplazamiento de la región derecha se describe mediante el vector de movimiento de bloque de campo superior derecho. Las líneas impares en los bloques de luminancia se subdividen también verticalmente para formar dos regiones de 8x8. El desplazamiento de la región izquierda se describe mediante el vector de movimiento de bloque de campo inferior izquierdo y el desplazamiento de la región derecha se describe mediante el vector de movimiento de bloque de campo inferior derecho. Cada bloque de crominancia puede partitionarse también en cuatro regiones de la misma manera que los bloques de luminancia, y cada región de bloque de crominancia puede compensarse en movimiento usando un vector de movimiento derivado.

Para macrobloques intra, el movimiento se supone que es cero.

En general, el procedimiento de calcular el predictor o predictores de vector de movimiento para un macrobloque actual en un fotograma P entrelazado consiste en dos etapas. En primer lugar, se recopilan tres vectores de movimiento candidatos para el macrobloque actual desde sus macrobloques vecinos. Por ejemplo, en una implementación, se recopilan vectores de movimiento candidatos basándose en la disposición mostrada en las Figuras 40A-40B (y diversos casos especiales para macrobloques de fila superior, etc.). Como alternativa, pueden recopilarse vectores de movimiento candidatos en algún otro orden o disposición. En segundo lugar, el predictor o predictores de vector de movimiento para el macrobloque actual se calcula a partir del conjunto de vectores de movimiento candidatos. Por ejemplo, el predictor puede calcularse usando predicción de mediana de 3 o mediante algún otro procedimiento.

Para detalles adicionales sobre cálculo de predictores y derivación de vectores de movimiento de crominancia para macrobloques de fotogramas P entrelazados, véase la Solicitud de Patente Provisional de Estados Unidos N.º

60/501.081, titulada "Video Encoding and Decoding Tools and Techniques", presentada el 7 de septiembre de 2003, y la sección XIV, a continuación.

IV. Predicción bidireccional de fotogramas de vídeo progresivos

Como se ha explicado anteriormente, los macrobloques en fotogramas B progresivos pueden predecirse usando cinco modos de predicción diferentes: hacia delante, hacia atrás, directo, interpolado e intra. Un codificador selecciona y señala los diferentes modos de predicción en el flujo de bits a nivel de macrobloque o algún otro nivel. En modo directo, un macrobloque en un fotograma B progresivo actual se deriva desde un ancla temporalmente anterior. En modo hacia atrás, un macrobloque en el fotograma B progresivo actual se deriva desde un ancla temporalmente posterior. Los macrobloques previstos en modos directo o interpolado usan anclas tanto hacia delante como hacia atrás para predicción. Puesto que hay dos fotogramas de referencia en modos directo e interpolado, hay típicamente al menos dos vectores de movimiento (codificados explícitamente o derivados) para cada macrobloque. (Diversos aspectos de la codificación, señalización y decodificación usadas para fotogramas B progresivos pueden usarse también para fotogramas B entrelazados, como se describe a continuación).

En algunas implementaciones, el codificador deriva implícitamente vectores de movimiento en modo directo cambiando de escala un vector de movimiento co-localizado para el ancla hacia delante usando un valor de fracción. La fracción puede reflejar la posición temporal relativa de un fotograma B progresivo actual dentro del intervalo formado por sus anclas, pero no refleja necesariamente verdaderas distancias inter-fotograma. Por lo tanto, el codificador no supone necesariamente velocidad constante. Esto permite al codificador un grado adicional de libertad para describir de manera precisa y económica el movimiento verdadero entre las anclas y el fotograma B progresivo actual cambiando la fracción desde la posición temporal "real" para mejorar la predicción de movimiento compensado. La variable BFRATION representa diferentes fracciones que pueden enviarse en un flujo de bits (por ejemplo, a nivel de imagen o algún otro nivel) para indicar su posición temporal relativa. Las diferentes fracciones son de un conjunto limitado de valores discretos entre 0 y 1.

Haciendo referencia de nuevo a la Figura 17, la tabla 1700 es una tabla de códigos de longitud variable (VLC) para el elemento de flujo de bits BFRATION. No hay restricción en la singularidad de BFRATION entre fotogramas B progresivos entre las mismas dos anclas; diferentes fotogramas B progresivos con anclas idénticas pueden tener el mismo valor BFRATION. Los códigos en la tabla 1700 pueden modificarse o reorganizarse para representar diferentes fracciones con diferentes códigos. Otros posibles códigos no mostrados en la tabla 1700 (por ejemplo, 1111110 o 1111111) pueden considerarse códigos inválidos, o pueden usarse para otros fines. Por ejemplo, la entrada 1111110 puede usarse para codificar explícitamente BFRATION en un formato de punto fijo. Como otro ejemplo, la entrada 1111111 puede usarse para señalar un tipo de fotograma específico (por ejemplo, un fotograma B progresivo intra-codificado).

Haciendo referencia de nuevo a la Figura 18, el decodificador halla un factor de cambio de escala de acuerdo con el pseudocódigo 1800. Haciendo referencia de nuevo a la Figura 19, el decodificador usa el factor de cambio de escala para cambiar de escala los elementos x e y del vector de movimiento para el macrobloque co-localizado en la imagen de referencia posterior. La función *Scale_Direct_MV* en el pseudocódigo 1900 toma las entradas MV_X y MV_Y y deriva dos vectores de movimiento en modo directo, haciendo referencia uno a la imagen ancla hacia delante (anterior), (MV_X_F, MV_Y_F) y haciendo referencia el otro a la imagen de ancla (MV_X_B, MV_Y_B) hacia atrás (posterior).

Una señal de macrobloque "saltado" en un fotograma B progresivo indica que no está presente error de predicción de vector de movimiento para un macrobloque dado. Los vectores de movimiento previstos serán exactamente los que usa el codificador/decodificador al reconstruir el macrobloque (es decir, no se aplica error de predicción de vector de movimiento). El codificador aún señala un modo de predicción para el macrobloque puesto que el macrobloque puede saltarse usando predicción directa, hacia delante, hacia atrás o interpolada.

V. Vista general de innovaciones en codificación/decodificación predictiva de imágenes B entrelazadas

Las realizaciones descritas incluyen técnicas y herramientas para codificar y decodificar imágenes B entrelazadas (por ejemplo, campos B entrelazados, fotogramas B entrelazados). Las realizaciones descritas implementan una o más de las técnicas y herramientas descritas para codificar y/o decodificar imágenes entrelazadas previstas bidireccionalmente incluyendo, pero sin limitación, lo siguiente:

1. Para fotogramas B entrelazados, un codificador/decodificador conmuta modos de predicción entre el campo superior e inferior en un macrobloque de un fotograma B entrelazado.
2. Para fotogramas B entrelazados, un codificador/decodificador calcula vectores de movimiento de modo directo para un macrobloque actual seleccionando un vector de movimiento representativo para cada uno de los campos superior e inferior del macrobloque co-localizado del ancla previamente decodificada temporalmente posterior. La selección puede realizarse basándose al menos en parte en el modo de codificación de macrobloque del fotograma B entrelazado actual (por ejemplo, modo de 1 MV, modo de MV de 2 campos, etc.).
3. Para campos B entrelazados o fotogramas B entrelazados, un codificador/decodificador usa codificación de 4MV. Por ejemplo, puede usarse 4MV en modos de predicción de una dirección (modos hacia delante o hacia

atrás), pero no en otros modos de predicción disponibles (por ejemplo, directo, interpolado).

4. Para campos B entrelazados o fotogramas B entrelazados, se predicen vectores de movimiento hacia delante usando vectores de movimiento hacia delante previamente reconstruidos (o estimados) desde una memoria intermedia de vector de movimiento hacia delante, y se predicen vectores de movimiento hacia atrás usando vectores de movimiento hacia atrás previamente reconstruidos (o estimados) desde una memoria intermedia de vector de movimiento hacia atrás. Los vectores de movimiento resultantes se añaden a la correspondiente memoria intermedia, y se rellenan huecos en las memorias intermedias de vector de movimiento con valores de vector de movimiento estimados.

a. Para fotogramas B entrelazados, cuando se usa predicción hacia delante para predecir un vector de movimiento y el vector de movimiento se añade a la memoria intermedia hacia delante, la posición correspondiente en la memoria intermedia hacia atrás se rellena ("relleno de huecos") con lo que sería el vector de movimiento previsto usando únicamente vectores de movimiento hacia atrás como predictores. De manera similar, cuando se usa predicción hacia atrás para predecir un vector de movimiento y el vector de movimiento se añade a la memoria intermedia hacia atrás, la posición correspondiente en una memoria intermedia hacia delante se rellena con lo que sería el vector de movimiento previsto usando únicamente vectores de movimiento hacia delante como predictores.

b. Para campos B entrelazados, para elegir entre diferentes vectores de movimiento de polaridad (por ejemplo, "misma polaridad" o "polaridad opuesta") para el relleno de huecos, un codificador/decodificador selecciona un vector de movimiento de campo de polaridad dominante. La distancia entre anclas y fotogramas actuales se calcula usando diversos elementos de sintaxis, y la distancia calculada se usa para cambiar de escala vectores de movimiento de campo de referencia.

5. Para campos B entrelazados, un codificador/decodificador usa fotogramas de "auto referencia". Por ejemplo, un segundo campo B en un fotograma actual hace referencia al primer campo B desde el fotograma actual en predicción de movimiento compensado.

6. Para campos B entrelazados, un codificador envía información binaria (por ejemplo, a nivel de campo B en un plano de bits comprimido) que indica si un modo de predicción es hacia delante o no hacia delante para uno o más macrobloques en el campo B entrelazado. Un decodificador realiza la decodificación correspondiente.

7. Para campos B entrelazados, un codificador/decodificador selecciona vectores de movimiento de modo directo usando lógica que favorece la polaridad dominante si el macrobloque correspondiente en el campo correspondiente de la siguiente imagen de ancla se codificó usando cuatro vectores de movimiento.

8. Campos intra-codificados: cuando no es posible buena compensación de movimiento para un campo B, puede codificarse como un campo B ("campo BI") intra (es decir, no previsto).

Las técnicas y herramientas descritas pueden usarse en combinación unas con las otras o con otras técnicas y herramientas, o pueden usarse de manera independiente.

VI. Conmutar modos de predicción en macrobloques codificados por campo en fotogramas B entrelazados

En algunas implementaciones, un codificador realiza conmutación de modo de predicción en macrobloques en fotogramas B entrelazados. Por ejemplo, un codificador permite la conmutación de modo de predicción de hacia delante a hacia atrás, o de hacia atrás a hacia delante, al pasar desde el campo superior al campo inferior en un macrobloque en un fotograma B entrelazado. En lugar de codificar un macrobloque completo con un único modo de dirección de predicción, se usa una combinación de modos de dirección de predicción para codificar un único macrobloque. La capacidad para cambiar los modos de dirección de predicción entre campos individuales en un macrobloque conduce a codificación más eficaz de fotogramas B entrelazados en muchos escenarios.

La Figura 41 muestra una técnica 4100 para predecir vectores de movimiento para campos individuales en un macrobloque codificado por campo en un fotograma B entrelazado usando diferentes modos de predicción. En 4110, en un fotograma B entrelazado, un codificador/decodificador predice un vector de movimiento para un primer campo en el macrobloque codificado por campo usando un primer modo de predicción. En algunas implementaciones, el "primer campo" puede ser cualquiera del campo superior o el campo inferior, la decisión para lo cual se señala de manera separada. En 4120, el codificador/decodificador predice un vector de movimiento para un segundo campo en el mismo macrobloque usando un modo de predicción diferente.

Por ejemplo, para un macrobloque codificado por campo usando dos vectores de movimiento, el campo superior puede predecirse hacia delante (es decir, el vector de movimiento de campo superior hace referencia a una imagen de ancla anterior), y el campo inferior puede predecirse hacia atrás (es decir, el campo inferior hace referencia a una imagen de ancla posterior). En algunas implementaciones, los macrobloques codificados por campo en fotogramas B entrelazados no se codifican usando cuatro vectores de movimiento. Como alternativa, si el macrobloque se codifica por campo usando cuatro vectores de movimiento (por ejemplo, dos vectores de movimiento para cada campo), entonces los dos vectores de movimiento para el campo superior harían referencia a un ancla (hacia delante o hacia atrás), y los vectores de movimiento de campo inferior harían referencia al otro ancla.

Esta conmutación de modo de predicción requiere únicamente un bit adicional en el caso donde el tipo de macrobloque no es directo o interpolado para comenzar con, como se ilustra adicionalmente en el siguiente

pseudocódigo para fotogramas B entrelazados:

```
Si MB está codificado por campo Y MB Type es hacia delante o hacia atrás entonces
  Si MVSwitch = 1 entonces el modo de predicción conmuta (de hacia delante a hacia
  atrás o viceversa) entre campo superior e inferior
```

- 5 Limitar el modo de predicción que conmute a modos hacia delante y hacia atrás por lo tanto evita la necesidad de más bits para señalar el segundo modo puesto que el segundo modo está implicado desde el primer modo (previamente señalado) y el valor de conmutación.

10 Si hay movimiento elevado en el área cubierta por un macrobloque de un fotograma B entrelazado, el macrobloque es probable que esté codificado en modo de campo. Bajo estas circunstancias, cualquiera de la predicción hacia delante o hacia atrás es más probable que proporcione un resultado de compensación de movimiento más preciso que los modos directo o interpolado, que implica promedio de píxeles. Debido a que promediar resultados al suavizar (por ejemplo, pérdida de elementos de alta frecuencia que acompañan movimiento elevado), los modos directo e interpolado pueden no ser la mejor manera para codificar tales macrobloques. Resultados experimentales indican la ineficacia a través de la tara aumentada de señalización de todos los cuatro modos de predicción como

15 opciones de conmutación en el nivel de campo en macrobloques codificados por campo.

Como alternativa, un codificador puede conmutar entre más de dos modos de predicción en macrobloques codificados por campo de fotogramas B entrelazados, o puede conmutar entre diferentes modos de predicción.

VII. Calcular vectores de movimiento de modo directo en fotogramas B entrelazados

20 En algunas implementaciones, un codificador/decodificador almacena en memoria intermedia vectores de movimiento desde un fotograma I o fotograma P de ancla previamente decodificado (que es el fotograma de referencia que está hacia delante en el tiempo, usado como el fotograma de referencia de predicción hacia atrás) y selecciona uno o más de los vectores de movimiento almacenados en memoria intermedia para uso al calcular

25 vectores de movimiento de modo directo para un macrobloque actual en un fotograma B entrelazado. Por ejemplo, el codificador/decodificador almacena en memoria intermedia un vector de movimiento representativo para cada uno de los campos superior e inferior desde cada macrobloque del fotograma de ancla, y usa uno o más de los vectores de movimiento almacenados en memoria intermedia para calcular vectores de movimiento para el macrobloque de modo directo actual. La selección se realiza basándose al menos en parte en el modo de codificación del macrobloque actual (por ejemplo, modo de 1 MV, modo de MV de 2 campos, etc.).

30 La Figura 42 muestra una técnica 4200 para calcular vectores de movimiento de modo directo para macrobloques en fotogramas B entrelazados en una implementación. En 4210, un codificador/decodificador almacena en memoria intermedia diversos vectores de movimiento por macrobloque de un macrobloque co-localizado en un fotograma de ancla temporalmente futuro previamente reconstruido. Si el macrobloque co-localizado únicamente tiene un vector de movimiento, ese vector de movimiento se almacena en memoria intermedia como el valor de vector de movimiento para los diversos bloques del macrobloque co-localizado, si fuera necesario. En 4220, el

35 codificador/decodificador selecciona uno o más de los vectores de movimiento almacenados en memoria intermedia del macrobloque co-localizado para predicción de modo directo del macrobloque actual en el fotograma B entrelazado, dependiendo en parte del número de vectores de movimiento necesarios para el macrobloque actual.

40 En una implementación, el decodificador almacena en memoria intermedia dos vectores de movimiento en el macrobloque co-localizado, o la mitad del número máximo posible de vectores de movimiento de luminancia decodificados, desde el fotograma de ancla futuro. Los macrobloques en el fotograma de ancla pueden codificarse de diferentes maneras, hasta con cuatro vectores de movimiento por macrobloque, pero únicamente se almacenan en memoria intermedia hasta dos vectores de movimiento, como se describe a continuación. Y, el número de pares de vectores de movimiento hacia delante/hacia atrás generados para el macrobloque actual depende del modo de codificación del macrobloque actual, en lugar de solamente del modo de codificación del macrobloque co-localizado

45 en el fotograma de ancla futuro previamente decodificado.

Por ejemplo, si el macrobloque de modo directo actual está codificado con 1 MV, un decodificador toma el vector de movimiento almacenado en memoria intermedia desde el campo superior del macrobloque co-localizado en el fotograma de ancla, y genera un par de vectores de movimiento directo - uno para cada una de las direcciones hacia delante y hacia atrás. Si el macrobloque de modo directo actual está codificado por campo, un decodificador toma

50 tanto los vectores de movimiento de campo inferior como de campo superior almacenados en memoria intermedia desde el macrobloque co-localizado en el fotograma de ancla y genera dos pares de vectores de movimiento, proporcionando un total de cuatro vectores de movimiento para el macrobloque de modo directo actual, uno para cada campo en ambas direcciones hacia delante y hacia atrás.

55 La Figura 43 muestra los vectores de movimiento MV1, MV2, MV3 y MV4 para los bloques de un macrobloque 4300 co-localizado de un fotograma de ancla temporalmente futuro previamente decodificado. Si el macrobloque co-localizado es un macrobloque de 1 MV, MV1, MV2, MV3 y MV4 son todos iguales. Si el macrobloque co-localizado es un macrobloque de MV de 2 campos, MV1 y MV2 son todos iguales a un valor y MV3 y MV4 son iguales a otro. Si el macrobloque co-localizado del fotograma de ancla es un macrobloque de MV de 4 campos o MV de 4

fotogramas, MV1, MV2, MV3 y MV4 pueden todos ser diferentes valores. Incluso si MV1, MV2, MV3 y MV4 están todos disponibles, no obstante, el decodificador almacena en memoria intermedia únicamente MV1 y MV3.

En el ejemplo mostrado en la Figura 43, el decodificador almacena en memoria intermedia MV1 y MV3. Si el macrobloque actual usa el modo 1 MV, el decodificador elige MV1 para calcular los vectores de movimiento de modo directo hacia delante y hacia atrás para el macrobloque actual, e ignora MV3. Si el macrobloque actual usa modo de MV de 2 campos, el decodificador usa tanto MV1 como MV3 para calcular los cuatro vectores de movimiento de modo directo. Esta operación produce una buena representación del movimiento para los campos superior e inferior del macrobloque actual.

Cuando se han seleccionado los vectores de movimiento desde el macrobloque co-localizado en el fotograma de ancla, el decodificador aplica lógica de cambio de escala para derivar los correspondientes vectores de movimiento que apuntan hacia delante y hacia atrás para la predicción de modo directo del macrobloque de fotograma B. Por ejemplo, el decodificador puede aplicar la función *Scale_Direct_MV* en la Figura 19. O, el decodificador aplica una función de cambio de escala diferente.

Como alternativa, un codificador/decodificador puede almacenar en memoria intermedia cuatro vectores de movimiento por macrobloque de un fotograma de ancla. Por ejemplo, si el macrobloque actual está codificado con 1 MV, el codificador/decodificador puede tomar el vector de movimiento superior izquierdo del macrobloque co-localizado en el fotograma de ancla y generar un par de vectores de movimiento directo, o tomar el promedio de los cuatro vectores de movimiento para el macrobloque del fotograma de ancla. Si el macrobloque actual está codificado por campo, el codificador/decodificador puede tomar los vectores de movimiento superior izquierdo e inferior izquierdo y generar dos pares (uno para cada campo), o tomar el promedio de los vectores de movimiento superiores y el promedio de los vectores de movimiento inferiores para el macrobloque del fotograma de ancla.

Los vectores de movimiento de modo directo se tratan como (0, 0) cuando el macrobloque co-localizado en el fotograma de ancla es intra, o cuando el fotograma de ancla es un fotograma 1.

VIII. Codificación de 4MV en campos B entrelazados y fotogramas B entrelazados

En algunas implementaciones, un codificador codifica campos B entrelazados y fotogramas B entrelazados usando un modo de codificación de cuatro vectores de movimiento (4MV). La codificación de 4MV puede permitir representación más precisa de trayectorias de movimiento complejo que la codificación de un vector de movimiento (1 MV) (por ejemplo, permitiendo que se predigan los cuatro bloques de luminancia en un macrobloque y se compensen en movimiento de manera independiente). El uso de 4MV puede limitarse a ciertos modos de predicción. Por ejemplo, en algunas implementaciones, un codificador usa 4MV para los modos hacia delante y hacia atrás (ambas variedades de campo y fotograma) pero no los modos directo o interpolado. Esto se diferencia de los modos de codificación progresiva donde no se usa 4MV en fotogramas B progresivos.

Los modos directo e interpolado implican promedio de píxeles cuando se calcula una predicción de movimiento compensado, que tiende a suavizar sobre detalles precisos. Si tal suavizado es aceptable, es probable que el modo de 1 MV pueda usarse de manera eficaz en lugar del modo 4MV, puesto que 1 MV es más económico de codificar y puede usarse para describir de manera precisa trayectorias de movimiento suave. La experimentación ha mostrado un beneficio para usar el modo de 4MV para macrobloques de campos B entrelazados y fotogramas B entrelazados, pero restringiendo el modo 4MV a macrobloques previstos hacia delante y hacia atrás. Otro factor en favorecer la restricción de 4MV a modos hacia delante y hacia atrás es que combinar 4MV con modos directo o interpolado daría como resultado un total de ocho vectores de movimiento en cada caso. La tara de señalización (para el modo interpolado) y la implementación y la complejidad de decodificación asociada con ocho vectores de movimiento generalmente desplaza los beneficios de precisión. Además, a menudo no es práctica para codificar imágenes B entrelazadas con ocho vectores de movimiento cuando las imágenes P, que normalmente se codifican a unos ajustes de calidad superiores (es decir, cuantificadas menos severamente), pueden usar normalmente únicamente uno o cuatro vectores de movimiento para compensación de movimiento.

Restringir 4MV a ciertos modos de predicción tiene otros beneficios también. Por ejemplo, si se restringe 4MV a únicamente modos de predicción hacia delante y hacia atrás, y si ya se ha señalado una decisión de modo hacia delante/no hacia delante (por ejemplo, en una técnica de codificación de plano de bits tal como la descrita a continuación en la sección XI), un codificador no necesita enviar ningún bit adicional para señalar el modo de predicción para un macrobloque de 4MV.

El siguiente pseudocódigo es aplicable a macrobloques de campos B entrelazados donde las decisiones hacia delante/no hacia delante están codificadas en plano de bits y se envían antes de cualquier información de nivel de macrobloque (por ejemplo, enviadas a nivel de imagen):

```
Si MB está codificado con 4MV Y el modo de predicción NO es hacia delante
Entonces el modo de predicción = hacia atrás (no enviar ningún bit más para
señalar el modo)
```

En algunas implementaciones, las decisiones de modo de predicción directo/no directo se envían antes de cualquier

información de nivel de macrobloque (por ejemplo, en un plano de bits comprimido a nivel de imagen). (Para más información sobre información de codificación directa/no directa, véase la Solicitud de Patente de Estados Unidos con N.º de Serie 10/622.378, titulada, "Advanced Bi-Directional Predictive Coding of Video Frames", presentada el 18 de julio de 2003.) El siguiente pseudocódigo es aplicable a macrobloques de fotogramas B entrelazados donde 4MV está restringido a modos hacia delante y hacia atrás en tales implementaciones:

```
Si MB está codificado con 4MV Y modo de predicción NO es directo
Entonces enviar un bit adicional para señalar el modo de predicción (hacia
delante o hacia atrás)
```

Como alternativa, se usa 4MV para modos de predicción distintos de o además de los modos hacia delante o hacia atrás, no se usa para el modo hacia delante, no se usa para el modo hacia atrás, o no se usa para ningún modo de predicción. Por ejemplo, en algunas implementaciones, se usa 4MV para campos B entrelazados pero no para fotogramas B entrelazados. En otras alternativas, pueden usarse otros códigos o longitudes de código para señalar modos de predicción en combinación con codificación de 4MV.

IX. Predecir vectores de movimiento en imágenes B entrelazadas usando memorias intermedias separadas de vector de movimiento hacia delante y hacia atrás

Los vectores de movimiento para imágenes B entrelazadas se predicen usando contextos de vector de movimiento hacia delante y hacia atrás separados. En general, los vectores de movimiento hacia delante se predicen usando vectores de movimiento almacenados en una memoria intermedia de vector de movimiento hacia delante, y los vectores de movimiento hacia atrás se predicen usando vectores de movimiento almacenados en una memoria intermedia de vector de movimiento hacia atrás. Los vectores de movimiento resultantes para los macrobloques actuales se almacenan a continuación en la memoria intermedia apropiada, y pueden usarse en predicción de vector de movimiento posterior para otros macrobloques. Típicamente, los espacios correspondientes en ambas memorias intermedias de vector de movimiento hacia delante y hacia atrás se rellenan para cada macrobloque, incluso si un macrobloque dado se predice únicamente con vectores de movimiento hacia delante (en el caso de un macrobloque previsto hacia delante) o únicamente vectores de movimiento hacia atrás (en el caso de macrobloque previsto hacia atrás). Las siguientes secciones describen técnicas para predecir vectores de movimiento en imágenes B entrelazadas (por ejemplo, campos B entrelazados, fotogramas B entrelazados) y para "rellenar" espacios correspondientes memorias intermedias de vector de movimiento para vectores de movimiento hacia delante o hacia atrás "faltantes".

A. Memorias intermedias hacia delante y hacia atrás

Cuando se predicen vectores de movimiento para imágenes B entrelazadas, un codificador/decodificador usa vectores de movimiento previamente reconstruidos en una memoria intermedia de vector de movimiento hacia delante y/o memoria intermedia de vector de movimiento hacia atrás. En modo hacia delante, el codificador/decodificador usa vectores de movimiento reconstruidos hacia delante desde la memoria intermedia de vector de movimiento hacia delante para predecir vectores de movimiento actuales para compensación de movimiento hacia delante. En modo hacia atrás, el codificador/decodificador usa vectores de movimiento hacia atrás reconstruidos desde la memoria intermedia de vector de movimiento hacia atrás para predecir vectores de movimiento actuales para compensación de movimiento hacia atrás. Para macrobloques de modo hacia delante o modo interpolado, el codificador/decodificador usan ambos la memoria intermedia de vector de movimiento hacia delante para predecir un componente de vector de movimiento hacia delante (o potencialmente múltiples componentes de movimiento hacia delante) y predecir la memoria intermedia de vector de movimiento hacia atrás para predecir un componente hacia atrás (o potencialmente múltiples componentes de movimiento hacia atrás).

Después de reconstruir vectores de movimiento para imágenes B entrelazadas, un codificador/decodificador almacena en memoria intermedia los vectores de movimiento hacia delante reconstruidos en una memoria intermedia de vector de movimiento hacia delante y almacena en memoria intermedia los vectores de movimiento hacia atrás reconstruidos en una memoria intermedia de vector de movimiento hacia atrás. En modo hacia delante, el codificador/decodificador almacena vectores de movimiento reconstruidos hacia delante en la memoria intermedia de vector de movimiento hacia delante. En modo hacia atrás, el codificador/decodificador almacena vectores de movimiento hacia atrás reconstruidos en la memoria intermedia de vector de movimiento hacia atrás. Para macrobloques que usan ambos modos de predicción directo o interpolado, el codificador/decodificador almacena tanto el componente o componentes de vector de movimiento hacia delante en la memoria intermedia de vector de movimiento hacia delante como el componente o componentes de vector de movimiento hacia atrás en la memoria intermedia de vector de movimiento hacia atrás.

Por ejemplo, si un codificador está codificando un macrobloque previsto hacia delante en la posición de coordenada de macrobloque (12, 13) en una imagen B entrelazada, el codificador calcula un predictor de vector de movimiento hacia delante y envía un residuo (suponiendo que el macrobloque no se "salta") para el vector de movimiento hacia delante en el flujo de bits. El decodificador decodifica el residuo (es decir, el diferencial), y reconstruye el vector de movimiento. El codificador/decodificador inserta el vector de movimiento reconstruido en la memoria intermedia de vector de movimiento hacia delante. El codificador/decodificador a continuación usa lógica de predicción de vector

de movimiento para calcular un predictor de vector de movimiento hacia atrás para rellenarse en un vector de movimiento hacia atrás, y pone el vector de movimiento hacia atrás en la memoria intermedia de vector de movimiento hacia atrás en la posición (12, 13). Por ejemplo, en un escenario de predicción de mediana de 3, el codificador/decodificador puede tomar la mediana de los vectores de movimiento hacia atrás almacenados en memoria intermedia en las posiciones (11, 13), (12, 12) y (13, 12) (los vecinos izquierdo, superior y superior derecho del macrobloque previsto hacia delante actual) para rellenar el vector de movimiento hacia atrás para (12, 13).

La Figura 44 muestra una técnica 4400 para predecir vectores de movimiento para un macrobloque actual en una imagen B entrelazada usando una memoria intermedia de vector de movimiento hacia delante y/o hacia atrás. En 4410, dependiendo de si el vector de movimiento a predecir es un vector de movimiento hacia delante o hacia atrás, el codificador/decodificador elige si usar la memoria intermedia de vector de movimiento hacia delante o hacia atrás. Si el vector de movimiento actual es un vector de movimiento hacia delante, el codificador/decodificador selecciona un conjunto de candidatos de predictor de vector de movimiento desde la memoria intermedia de vector de movimiento hacia delante en 4420. Si el vector de movimiento actual es un vector de movimiento hacia atrás, el codificador/decodificador selecciona un conjunto de candidatos de predictor de vector de movimiento desde la memoria intermedia de vector de movimiento hacia atrás en 4430. En 4440, el codificador/decodificador calcula un predictor de vector de movimiento basándose en el conjunto del conjunto de candidatos de predictor de vector de movimiento. Por ejemplo, el codificador/decodificador calcula la mediana del conjunto de candidatos de predictor de vector de movimiento. En un caso sencillo, el codificador/decodificador calcula un predictor de vector de movimiento para un macrobloque actual de 1 MV basándose en predictores donde todos eran macrobloques de 1 MV. Se describen variaciones más complejas a continuación en las que el macrobloque actual y/o macrobloques vecinos tienen diferentes modos.

La Figura 45 muestra vectores de movimiento en una memoria intermedia 4510 de vector de movimiento hacia delante y una memoria intermedia 4520 de vector de movimiento hacia atrás. En el ejemplo mostrado en la Figura 45, para los macrobloques 4530-4570 reconstruidos, un codificador/decodificador almacena vectores de movimiento hacia delante en la memoria intermedia 4510 de vector de movimiento hacia delante y vectores de movimiento hacia atrás en la memoria intermedia 4520 de vector de movimiento hacia atrás. Para predecir el vector de movimiento para el macrobloque 4580 actual, el codificador/decodificador usa predictores candidatos desde macrobloques vecinos. Por ejemplo, si el macrobloque 4580 actual se predice en modo hacia delante, el codificador puede predecir el vector de movimiento hacia delante usando vectores de movimiento hacia delante vecinos en la memoria intermedia de vector de movimiento hacia delante (por ejemplo, usando predicción de mediana de 3), a continuación más tarde rellenar la posición del macrobloque actual en la memoria intermedia de vector de movimiento hacia delante con el valor de vector de movimiento reconstruido. Para rellenar la posición de macrobloque actual correspondiente en la memoria intermedia 4520 de vector de movimiento hacia atrás, el codificador/decodificador puede predecir un vector de movimiento hacia atrás usando vectores de movimiento hacia atrás vecinos en la memoria intermedia de movimiento hacia atrás y poner el predictor en la posición para el macrobloque actual en la memoria intermedia de vector de movimiento hacia atrás.

B. Predicción de vector de movimiento en fotogramas B entrelazados

En algunas implementaciones, un codificador/decodificador emplea el siguiente esquema para predecir vectores de movimiento para macrobloques (incluyendo campos separados de los mismos) en fotogramas B entrelazados, que usan contextos de vector de movimiento hacia delante y hacia atrás separados. Las Figuras 40A-40B muestran macrobloques vecinos a partir de los cuales se recogen los vectores de movimiento candidatos.

Si un macrobloque de 1 MV está previsto hacia delante, un codificador/decodificador predice su vector de movimiento hacia delante desde vectores de movimiento candidatos de la memoria intermedia de vector de movimiento hacia delante (por ejemplo, usando predicción de mediana de 3 y patrones de predicción tales como aquellos mostrados en las Figuras 40A y 40B o en cualquier otro lugar). El codificador/decodificador almacena el vector de movimiento hacia delante (después de añadir el error de predicción de vector de movimiento) en la memoria intermedia de vector de movimiento hacia delante. El codificador/decodificador rellena "huecos" prediciendo un vector de movimiento hacia atrás desde vectores de movimiento candidatos de la memoria intermedia de vector de movimiento hacia atrás (por ejemplo, como en el caso de predicción hacia delante), y almacena el vector de movimiento hacia atrás (en este punto, el predictor) en la memoria intermedia de vector de movimiento hacia atrás.

Si el macrobloque de 1 MV se predice hacia atrás, un codificador/decodificador predice su vector de movimiento hacia atrás desde vectores de movimiento candidatos de la memoria intermedia de vector de movimiento hacia atrás (por ejemplo, como en el caso de predicción hacia delante). El codificador/decodificador almacena el vector de movimiento hacia atrás (después de añadir el error de predicción) en la memoria intermedia de vector de movimiento hacia atrás. El codificador/decodificador rellena huecos prediciendo un vector de movimiento hacia delante desde vectores de movimiento candidatos de la memoria intermedia de vector de movimiento hacia delante, y almacena el vector de movimiento hacia delante (en este punto, el predictor) en la memoria intermedia de vector de movimiento hacia delante.

Los vecinos que son macrobloques intra-codificados se ignoran en las memorias intermedias de vector de movimiento hacia delante y hacia atrás.

Diversos casos especiales tratan combinaciones de macrobloques de 1 MV y de 2MV codificados por campo en fotogramas B entrelazados. Si un macrobloque vecino en la posición A, B o C para un macrobloque de 1 MV actual es un macrobloque de 2MV codificado por campo, el codificador/decodificador toma el promedio de los vectores de movimiento de campo para el macrobloque de 2MV como el predictor de vector de movimiento para esa posición.

- 5 Para macrobloque de MV de 2 campos actual previsto hacia delante, por ejemplo, para cada uno de dos vectores de movimiento de campo previstos hacia delante, los vectores de movimiento candidatos desde los vecinos se recopilan desde la memoria intermedia de vector de movimiento hacia delante. El codificador/decodificador selecciona un conjunto de vectores de movimiento candidatos basándose en el modo de codificación (por ejemplo, intra, 1MV, MV de 2 campos) de los macrobloques vecinos, que se almacenan en la memoria intermedia de vector de movimiento hacia delante.
- 10 Si existe un macrobloque vecino y no está intra codificado, el codificador/decodificador busca el vector o vectores de movimiento para el macrobloque para añadir al conjunto de candidatos. En algunas realizaciones, el codificador/decodificador continúa como sigue. Para el vector de movimiento hacia delante de campo superior, si un macrobloque vecino en la posición A, B o C es un macrobloque de 1 MV, el codificador añade el vector de movimiento del macrobloque desde la posición correspondiente de la memoria intermedia de vector de movimiento hacia delante al conjunto de candidatos.
- 15 Para un macrobloque vecino en la posición A, B o C que es un macrobloque de MV de 2 campos, el codificador/decodificador añade el MV de campo superior desde la posición correspondiente de la memoria intermedia de vector de movimiento hacia delante al conjunto.

- Para el vector de movimiento hacia delante de campo inferior, si un macrobloque vecino en la posición A, B o C es un macrobloque de 1 MV, el codificador añade el vector de movimiento del macrobloque desde la posición correspondiente de la memoria intermedia de vector de movimiento hacia delante al conjunto de candidatos. Para un macrobloque vecino en la posición A, B o C que es un macrobloque de MV de 2 campos, el codificador/decodificador añade el vector de movimiento de campo inferior desde la posición correspondiente de la memoria intermedia de vector de movimiento hacia delante al conjunto.
- 20

- Para calcular predictores para vectores de movimiento de campo en macrobloques de MV de 2 campos, el codificador/decodificador a continuación calcula las medianas de los conjuntos de candidatos.
- 25

Para calcular unos vectores de movimiento previstos hacia atrás para un macrobloque de MV de 2 campos, la lógica es la misma que el caso de predicción hacia adelante, pero los vectores de movimiento candidatos desde los vecinos se recopilan desde la memoria intermedia de vector de movimiento hacia atrás.

- De nuevo, los vecinos en la posición A, B o C que están intra-codificados se ignoran para predicción de vector de movimiento.
- 30

- Después de la reconstrucción de los vectores de movimiento para un macrobloque de MV de 2 campos (por ejemplo, añadiendo información de vector de movimiento diferencial), los vectores de movimiento reales reconstruidos se ponen en la memoria intermedia de vector de movimiento hacia delante o en la memoria intermedia de vector de movimiento hacia atrás, según sea apropiado para la dirección de predicción de los vectores de movimiento reconstruidos. Los intervalos vacíos correspondientes de la memoria intermedia de vector de movimiento para la dirección faltante se rellenan calculando predictores de vector de movimiento para la dirección faltante y almacenando los predictores de vector de movimiento en los intervalos vacíos.
- 35

- Se aplica una excepción para el relleno de huecos para macrobloques codificados por campo en fotogramas B entrelazados si se está usando la conmutación de modo de predicción (véase la sección VI, anterior). En este caso, un macrobloque de 2MV codificado por campo dado tiene un vector de movimiento hacia delante y un vector de movimiento hacia atrás. Después de la reconstrucción de un macrobloque codificado por campo de un fotograma B entrelazado, donde el macrobloque codificado por campo conmuta direcciones de predicción entre los campos superior e inferior, el codificador/decodificador rellena ambos "intervalos" de vector de movimiento superior e inferior de la memoria intermedia de vector de movimiento hacia delante con el vector de movimiento hacia delante, y rellena ambos intervalos de vector de movimiento superior e inferior de la memoria intermedia hacia atrás con el vector de movimiento hacia atrás. Aunque se envía el vector de movimiento hacia delante únicamente para un campo (por ejemplo, el campo superior), el codificador pone el mismo vector de movimiento en ambos intervalos de vector de movimiento de campo superior e inferior para la memoria intermedia de vector de movimiento hacia delante. De manera similar, aunque se envía el vector de movimiento hacia atrás únicamente para el campo inferior, el codificador lo pone ambos intervalos de campo superior e inferior de la memoria intermedia de vector de movimiento hacia atrás.
- 40
- 45
- 50

- Por ejemplo, la Figura 46 muestra vectores de movimiento de campo superior e inferior para un macrobloque 4680 reconstruido en una memoria intermedia 4610 de vector de movimiento hacia delante y una memoria intermedia 4620 de vector de movimiento hacia atrás. En el ejemplo mostrado en la Figura 46, para los macrobloques 4630-4670 reconstruidos, un codificador/decodificador almacena vectores de movimiento hacia delante en la memoria intermedia 4610 de vector de movimiento hacia delante y vectores de movimiento hacia atrás en la memoria intermedia 4620 hacia atrás. El macrobloque 4680 reconstruido está codificado por campo con conmutación de predicción, y su vector de movimiento de campo superior está almacenado en las localizaciones superior e inferior en cualquiera de la memoria intermedia de vector de movimiento hacia delante o hacia atrás (dependiendo de la
- 55

dirección de predicción del vector de movimiento de campo superior). El vector de movimiento de campo inferior del macrobloque 4680 se almacena en las localizaciones superior e inferior de la otra memoria intermedia de vector de movimiento. En este ejemplo, el macrobloque 4680 reconstruido usa la conmutación de modo de predicción. Aunque el vector de movimiento hacia delante y el vector de movimiento hacia atrás se envían cada uno para únicamente un campo, el codificador pone el mismo vector de movimiento en ambos intervalos de vector de movimiento de campo superior e inferior para las respectivas memorias intermedias de vector de movimiento hacia delante y hacia atrás.

Si el macrobloque actual está interpolado, un codificador/decodificador usa la memoria intermedia de vector de movimiento hacia delante para predecir el vector de movimiento hacia delante (o vectores de movimiento hacia delante para un macrobloque de MV de 2 campos), usa la memoria intermedia de vector de movimiento hacia atrás para predecir el vector de movimiento hacia atrás (o vectores de movimiento hacia atrás para un macrobloque de MV de 2 campos), y almacena los vectores de movimiento hacia delante y hacia atrás (después de añadir los errores de predicción, una vez que estos se han calculado), en las memorias intermedias de vector de movimiento hacia delante y hacia atrás, respectivamente.

Si el macrobloque está previsto directo en un fotograma B entrelazado, un codificador/decodificador puede usar una técnica descrita en la sección VII, anterior.

En algunas implementaciones, están permitidos macrobloques 1 MV, macrobloques de MV de 2 campos, macrobloques intra para fotogramas B entrelazados (pero no otros tipos de macrobloque de MV), que simplifica la lógica para predecir vectores de movimiento puesto que necesitan tratarse menos combinaciones de modo actual/vecino. Como alternativa, están permitidos otros modos de MV y/o adicionales tales como macrobloques de MV de 4 fotogramas y macrobloques de MV de 4 campos. Por ejemplo, partes del pseudocódigo mostrado en las Figuras 64, 69 y 70 pueden usarse para tratar tales otras combinaciones en fotogramas B entrelazados.

C. Predicción de vector de movimiento para campos B entrelazados

En general, para un campo B entrelazado, previamente reconstruido (o derivado) se usan vectores de movimiento de campo hacia delante como predictores para vectores de movimiento de campo hacia delante actuales, y se usan vectores de movimiento de campo hacia atrás previamente reconstruidos (o derivados) para vectores de movimiento de campo hacia atrás actuales. En modo hacia delante o hacia atrás, los vectores de movimiento de campo hacia delante o hacia atrás actuales se añaden a la memoria intermedia de vector de movimiento apropiada, y un vector de movimiento para la otra (faltante) dirección (por ejemplo, la dirección hacia atrás en modo hacia delante, o la dirección hacia delante en modo hacia atrás) se deriva para uso posterior como un predictor.

En algunas implementaciones, se realiza selección de predicción de vector de movimiento de campo de acuerdo con la lógica de predicción de vector de movimiento de campo de dos referencias descrita en la sección III.A.2 de la descripción detallada y a continuación en la sección XIV.B.3. Por ejemplo, el pseudocódigo mostrado en las Figuras 33A-33F se usa para calcular dos predictores de vector de movimiento hacia delante de campo para un macrobloque de un campo B entrelazado, y se selecciona un predictor de vector de movimiento para uso al reconstruir el vector de movimiento de campo hacia delante. El valor de vector de movimiento reconstruido se pone a continuación en la memoria intermedia de vector de movimiento hacia delante. El pseudocódigo se usa también para calcular dos predictores de vector de movimiento hacia atrás de campo para el macrobloque, y se selecciona un predictor para uso como un valor de relleno para la memoria intermedia de vector de movimiento hacia atrás. Para campos B entrelazados, para rellenar "huecos" en una memoria intermedia de vector de movimiento de dirección faltante, un codificador/decodificador elige entre predictores de vector de movimiento de la misma polaridad y la polaridad opuesta. Esta elección entre polaridades surge puesto que se generan dos predictores en una dirección faltante dada, uno de la misma polaridad que el campo actual, y uno de la polaridad opuesta que el campo actual. Por lo tanto, en algunas implementaciones, el codificador/decodificador elige el predictor de mayoría o polaridad "dominante" para el vector de movimiento de dirección faltante. De esta manera, está presente un conjunto completo tanto de vectores de movimiento hacia delante como hacia atrás para uso en predicción de vector de movimiento. Como alternativa, se determina la polaridad dominante y se realiza en primer lugar la selección de predictor, y únicamente se calcula el predictor de vector de movimiento seleccionado.

Los procedimientos de almacenamiento en memoria intermedia de valor real y relleno de huecos seleccionando de entre predictores de vector de movimiento de campo de diferentes polaridades en una implementación se muestran en el pseudocódigo 4700 en la Figura 47. El pseudocódigo 4700 muestra que durante la predicción de relleno de huecos, no está presente vector de movimiento real para la dirección faltante, por lo que se elige el vector de movimiento de dirección faltante previsto que tiene la polaridad dominante mediante el codificador/decodificador.

En algunas implementaciones, el esquema global para predicción de vector de movimiento de campo B entrelazado es como sigue.

Si el macrobloque está previsto hacia delante, un codificador/decodificador predice su vector de movimiento hacia delante desde el mismo candidato y/u vectores de movimiento de polaridad opuesta de la memoria intermedia de vector de movimiento hacia delante (por ejemplo, usando predicción de mediana de 3 desde los vecinos izquierdo, superior y superior-derecho para la mayoría de los casos) o vectores de movimiento derivados desde los vectores de

movimiento almacenados en memoria intermedia. El codificador/decodificador almacena el vector de movimiento reconstruido hacia delante en la memoria intermedia de vector de movimiento hacia delante y calcula y almacena el predictor de vector de movimiento hacia atrás dominante (previsto de manera similar con mediana de 3 desde las cercanías espaciales de la memoria intermedia de vector de movimiento hacia atrás) en la posición correspondiente en la memoria intermedia de vector de movimiento hacia atrás.

Si el macrobloque está previsto hacia atrás, un codificador/decodificador predice su vector de movimiento hacia atrás desde el mismo candidato y/o vectores de movimiento de polaridad opuesta de la memoria intermedia de vector de movimiento hacia atrás (por ejemplo, usando predicción de mediana de 3 desde los vecinos izquierdo, superior y superior-derecho para la mayoría de los casos) o vectores de movimiento derivados desde los vectores de movimiento almacenados en memoria intermedia. El codificador/decodificador almacena el vector de movimiento hacia atrás reconstruido en la memoria intermedia de vector de movimiento hacia atrás y calcula y almacena el predictor de vector de movimiento hacia delante dominante (previsto de manera similar con mediana de 3 desde las cercanías espaciales de la memoria intermedia de vector de movimiento hacia delante) en la posición correspondiente en la memoria intermedia de vector de movimiento hacia delante.

Si el macrobloque está interpolado, un codificador/decodificador usa la memoria intermedia de vector de movimiento hacia delante para predecir el componente de vector de movimiento hacia delante, usa la memoria intermedia de vector de movimiento hacia atrás para predecir el componente de vector de movimiento hacia atrás, y almacena los vectores de movimiento hacia delante y hacia atrás reconstruidos (después de añadir errores de predicción, una vez que estos se han calculado), en las memorias intermedias de vector de movimiento hacia delante y hacia atrás, respectivamente.

Si el macrobloque está previsto directo, el codificador/decodificador calcula los vectores de movimiento de modo directo para el campo actual y almacena los componentes de vector de movimiento hacia delante y hacia atrás en las respectivas memorias intermedias de vector de movimiento.

Los vecinos que son macrobloques intra-codificados se ignoran en la predicción de vector de movimiento.

Diversos casos especiales tratan combinaciones de macrobloques de 1 MV y 4MV en campos B entrelazados. Las Figuras 6A-10 muestran patrones de predictor para predicción de vector de movimiento para fotogramas P progresivos. Estos mismos patrones muestran las localizaciones de los bloques o macrobloques considerados para los vectores de movimiento candidatos para predicción de vector de movimiento para un vector de movimiento para un macrobloque de 1 MV o 4MV campos B entrelazados de MV mixto. Para el caso especial donde el fotograma es de un macrobloque de ancho, el predictor es siempre el Predictor A (el predictor superior). Diversas otras reglas tratan otros casos especiales tales como bloques de fila superior para macrobloques de 4MV de fila superior, macrobloques de 1 MV de fila superior y predictores intra-codificados.

Los patrones de predictor mostrados en las Figuras 6A-10 se usan para predicción hacia delante usando candidatos desde posiciones en la memoria intermedia de vector de movimiento hacia delante, y también para predicción hacia atrás usando candidatos desde posiciones en la memoria intermedia de vector de movimiento hacia atrás. Además, los patrones de predictor mostrados en las Figuras 6A-10 se usan en conjunto con la lógica de predicción de vector de movimiento de dos campos de referencia analizada anteriormente para campos B entrelazados.

Las Figuras 6A y 6B muestran localizaciones de bloques considerados para predictores de vector de movimiento candidato para un macrobloque actual de 1 MV en un campo B entrelazado de MV mixto. Los macrobloques vecinos pueden ser macrobloques de 1 MV o de 4MV. Las Figuras 6A y 6B muestran las localizaciones para los vectores de movimiento candidatos suponiendo que los vecinos son 4MV (es decir, el Predictor A es el vector de movimiento para el bloque 2 en el macrobloque por encima del macrobloque actual, y el predictor C es el vector de movimiento para el bloque 1 en el macrobloque inmediatamente a la izquierda del macrobloque actual). Si cualquiera de los vecinos es un macrobloque de 1 MV, entonces el predictor de vector de movimiento mostrado en las Figuras 5A y 5B se toma para que sea el predictor de vector de movimiento para el macrobloque completo. Como muestra la Figura 6B, si el macrobloque es el último macrobloque en la fila, entonces el Predictor B es desde el bloque 3 del macrobloque superior izquierdo en lugar de desde el bloque 2 en el macrobloque superior derecho como es el caso de otra manera.

Las Figuras 7A-10 muestran las localizaciones de bloques considerados para predictores de vector de movimiento candidato para cada uno de los 4 bloques de luminancia en un macrobloque de 4MV de campo B entrelazado de MV mixto. Las Figuras 7A y 7B son diagramas que muestran las localizaciones de bloques considerados para predictores de vector de movimiento candidato para un bloque en la posición 0; Las Figuras 8A y 8B son diagramas que muestran las localizaciones de bloques considerados para predictores de vector de movimiento candidato para un bloque en la posición 1; La Figura 9 es un diagrama que muestra las localizaciones de bloques considerados para predictores de vector de movimiento candidato para un bloque en la posición 2; y la Figura 10 es un diagrama que muestra las localizaciones de bloques considerados para predictores de vector de movimiento candidato para un bloque en la posición 3. De nuevo, si un vecino es un macrobloque de 1 MV, el predictor de vector de movimiento para el macrobloque se usa para los bloques del macrobloque.

Para el caso donde el macrobloque es el primer macrobloque en la fila, el Predictor B para el bloque 0 se maneja de manera diferente que el bloque 0 para los macrobloques restantes en la fila (véanse las Figuras 7A y 7B). En este caso, el Predictor B se toma desde el bloque 3 en el macrobloque inmediatamente por encima del macrobloque actual en lugar de desde el bloque 3 en el macrobloque por encima y a la izquierda del macrobloque actual, como es el caso de otra manera. De manera similar, para el caso donde el macrobloque es el último macrobloque en la fila, el Predictor B para el bloque 1 se maneja de manera diferente (Figuras 8A y 8B). En este caso, el predictor se toma desde el bloque 2 en el macrobloque inmediatamente por encima del macrobloque actual en lugar de desde el bloque 2 en el macrobloque por encima y a la derecha del macrobloque actual, como es el caso de otra manera. En general, si el macrobloque está en la primera columna de macrobloque, entonces el Predictor C para los bloques 0 y 2 se establece igual a 0.

De nuevo, los vecinos en la posición A, B o C que están intra-codificados se ignoran para predicción de vector de movimiento.

Después de la reconstrucción de los vectores de movimiento para un macrobloque de 4MV (por ejemplo, añadiendo información de vector de movimiento diferencial), los vectores de movimiento reales reconstruidos se ponen en la memoria intermedia de vector de movimiento hacia delante o memoria intermedia de vector de movimiento hacia atrás, según sea apropiado para la dirección de predicción de los vectores de movimiento reconstruidos. Los intervalos vacíos correspondientes de la memoria intermedia de vector de movimiento para la dirección faltante se rellenan calculando predictores de vector de movimiento para la dirección faltante para la misma polaridad y la opuesta, seleccionando entre los predictores de vector de movimiento de polaridad diferente, y almacenando los predictores de vector de movimiento en los intervalos vacíos.

Haciendo referencia de nuevo a las Figuras 34A y 34B, para predicción de vector de movimiento, un codificador/decodificador deriva un predictor de vector de movimiento de campo desde otro predictor de vector de movimiento de campo usando la operación de cambio de escala ilustrada en el pseudocódigo 3400. Se muestran dos posibles conjuntos de valores en la tabla 3500 en la Figura 35 para el caso donde el campo actual es el primer campo en el fotograma de vídeo entrelazado, y en la tabla 3600 en la Figura 36 para el caso donde el campo actual es el segundo campo en el fotograma de vídeo entrelazado. En las tablas 3500 y 3600, los valores de SCALEOPP, SCALESAME1, SCALESAME2, SCALEZONE1_X, SCALEZONE1_Y, ZONE1OFFSET_X y ZONE1OFFSET_Y dependen de la distancia de fotograma de referencia.

En algunas implementaciones, las distancias de fotograma de referencia para referencias hacia delante y hacia atrás en campos B entrelazados se calculan usando codificación de fracción. El elemento de sintaxis BFRACTION (que se señala para macrobloques de modo de predicción hacia delante o hacia atrás de campos B entrelazados, no solamente macrobloques de modo directo de campos B entrelazados) se usa para derivar distancias de imagen de referencia hacia delante y hacia atrás como se muestra en el siguiente pseudocódigo:

```

Distancia de fotograma de referencia hacia delante (FRFD) =
NINT ((BFRACTION numerador / BFRACTION denominador) * Distancia de fotograma de
referencia) - 1
if (FRFD < 0) then FRFD = 0
Distancia de fotograma de referencia hacia atrás (BRFD) = Distancia de fotograma
de referencia - FRFD - 1
(donde NINT es un operador de número entero más cercano).
```

El numerador y denominador de BFRACTION se decodifican desde el elemento de sintaxis BFRACTION. El elemento BFRACTION puede usarse para representar diferentes fracciones que pueden enviarse en un flujo de bits (por ejemplo, a nivel de fotograma para campos B entrelazados). La fracción toma un conjunto limitado de valores discretos entre 0 y 1 e indica la posición temporal relativa de la imagen B en el intervalo formado por sus anclas.

Para predicción hacia delante y para predicción hacia atrás del segundo campo en un fotograma que tiene campos B entrelazados, el codificador/decodificador realiza cambio de escala de vector de movimiento de acuerdo con el pseudocódigo 3400 en las Figuras 34A y 34B. Sin embargo, en algunas implementaciones un codificador/decodificador que realiza predicción de vector de movimiento hacia atrás para el primer campo usa las funciones scaleforopposite_x, scaleforopposite_y, scaleforsame_x y scaleforsame_y como se definen en el pseudocódigo 4800 mostrado en la Figura 48. Los valores de SCALESAME, SCALEOPP1, SCALEOPP2, SCALEZONE1_X, SCALEZONE1_Y, ZONE1OFFSET_X y ZONE1OFFSET_Y para un primer campo B entrelazado en una implementación se muestran en la tabla 4900 en la Figura 49.

En la tabla 4900 la relación entre la variable N y el intervalo del vector de movimiento es la misma que la relación descrita con referencia a las Figuras 35 y 36 y la Tabla 1 anterior.

Como alternativa, las distancias de fotograma de referencia se calculan de otra manera o se realiza cambio de escala de acuerdo con diferentes algoritmos. Por ejemplo, se realiza cambio de escala sin considerar el valor de N (es decir, N se supone que es 1).

X. Fotogramas de “auto-referencia” con campos B entrelazados

Los fotogramas que tienen campos B entrelazados se codifican como dos campos separados (y un tanto codificados de manera independiente). El campo superior consiste en las líneas de trama pares (empezando con la línea 0) del fotograma, y el campo inferior consiste en líneas de trama impares del fotograma. Puesto que los campos en “imágenes de campo” son decodificables de manera independiente, no necesitan enviarse en ningún orden pre-

establecido. Por ejemplo, un codificador puede enviar el campo inferior en primer lugar seguido por el campo superior, o viceversa. En algunas implementaciones, el orden de los dos campos se indica mediante un elemento de sintaxis de “campo superior en primer lugar”, que es cualquiera de verdadero o falso dependiendo del orden temporal correcto para decodificar los dos campos de un fotograma.

Los codificadores y decodificadores anteriores han usado los fotogramas de ancla anterior y siguiente (por ejemplo, fotogramas I o P) o campos en los fotogramas de ancla anterior y siguiente como las imágenes de “referencia” para realizar compensación de movimiento para una imagen B actual. Los codificadores y decodificadores anteriores tienen también imágenes B restringidas o cualquier parte de las mismas que se use como referencias de compensación de movimiento para cualquier imagen. En algunas implementaciones de las técnicas y herramientas descritas, sin embargo, una o más de estas “reglas” están relajadas.

Por ejemplo, en algunas implementaciones, un primer campo B entrelazado hace referencia al primer y segundo campo desde las imágenes de ancla anterior y siguiente. El segundo campo B entrelazado hace referencia al primer campo B entrelazado desde la imagen actual como el campo de “polaridad opuesta” y el campo de misma polaridad del fotograma de ancla anterior como el campo de la “misma polaridad”, además del primer y segundo campos de la siguiente imagen de ancla.

La Figura 50B es un diagrama que muestra campos de referencia para cada uno de dos campos B entrelazados en un fotograma de vídeo entrelazado B2. En el ejemplo mostrado en la Figura 50B, el primer campo B a codificarse (en este punto, el campo superior) está permitido a hacer referencia a dos campos de referencia en el ancla hacia delante (pasado temporal) P1 y dos campos de referencia desde el ancla hacia atrás (futuro temporal) P3, para un total de cuatro campos de referencia. El segundo campo B entrelazado a codificarse para B2 está permitido a hacer referencia al primer campo desde el mismo fotograma de vídeo entrelazado (rompiendo por lo tanto con la convención de no permitir que partes de imágenes B actúen como referencias) y un campo de referencia desde el ancla anterior P1, así como ambos campos desde el ancla futura P3. Por motivos de comparación, la Figura 50A muestra una convención seguida por campos P entrelazados de fotogramas de vídeo entrelazados.

Las técnicas y herramientas que implementan estas reglas de referencia de campo B entrelazado pueden proporcionar mejor compresión. La codificación de campo de vídeo entrelazado es más eficaz para codificar movimiento elevado, es decir, cuando hay movimiento considerable entre campos superior e inferior. En este escenario, por ejemplo, el campo superior (y primero codificado) de un fotograma será un predictor mucho mejor para los píxeles en el campo inferior del mismo fotograma que el campo superior tomado desde el fotograma anterior, que proviene desde una distancia lejos temporal mayor. Debido a su distancia temporal mayor, estos predictores temporalmente adicionales ofrecen predicción mucho más débil cuando el movimiento es elevado. Además, la probabilidad de oclusiones se amplía para predictores que están temporalmente más remotos, que da como resultado más macrobloques intra-codificados que son costosos de codificador. En particular, la experimentación verifica que permitir el segundo campo B entrelazado temporal de un fotograma para hacer referencia al primer campo B entrelazado temporal del mismo fotograma puede producir ganancias de compresión significativas.

XI. Codificación de plano de bits de modo hacia delante en campos B entrelazados

Como se ha mencionado en la sección X anterior, en algunas implementaciones un segundo campo B entrelazado codificado de un fotograma actual puede hacer referencia al primer campo B entrelazado codificado del fotograma actual. Esta técnica de “auto-referencia” es eficaz en campos B entrelazados en fotogramas que tienen elevado movimiento puesto que el campo B temporalmente más cercano en el fotograma actual es a menudo un mejor predictor que campos de ancla temporalmente más hacia delante. Cuando los fotogramas con campos B entrelazados tienen elevado movimiento y el segundo campo B entrelazado temporal prefiere el primer campo B entrelazado temporal que una referencia predictiva, entonces el modo de predicción más eficaz para los macrobloques en el segundo campo B entrelazado a menudo será “hacia delante”.

Puesto que la predicción de modo hacia delante en campos B entrelazados es una herramienta eficaz para reducir la tasa de bits, es ventajoso reducir la tara de señalización para reducir el coste global de la predicción de señalización de modo hacia delante, especialmente en situaciones de baja tasa de bits. Por consiguiente, en algunas realizaciones, un codificador codifica información de predicción de modo hacia delante usando una técnica de codificación de plano de bits unificada. Por ejemplo, un codificador codifica información de predicción de modo hacia delante en un plano de bits comprimido, donde cada bit en el plano de bits está asociado con un macrobloque, y el valor de cada bit señala si el macrobloque se codificó en modo hacia delante o en un modo de predicción no hacia delante.

El plano de bits comprimido puede enviarse al nivel de fotograma, nivel de campo o a algún otro nivel. La técnica de codificación de plano de bits se aprovecha de la desviación de modo hacia delante en comparación con otros modos de predicción para campos B entrelazados. Por ejemplo, si la mayoría de los macrobloques en un campo B entrelazado usan predicción hacia delante, un codificador puede reducir la tara de señalización a menos de un bit por macrobloque codificando en plano de bits la decisión hacia delante/no hacia delante.

La Figura 51 muestra una técnica 5100 para codificar información de decisión de modo de predicción hacia delante/no hacia delante para macrobloques de un campo B entrelazado en un codificador de vídeo que tiene uno o más modos de codificación de plano de bits. La Figura 52 muestra una técnica 5200 correspondiente para decodificar información de decisión de modo de predicción hacia delante/no hacia delante codificada mediante un codificador de vídeo que tiene uno o más modos de codificación de plano de bits.

Con referencia a la figura 51, el codificador selecciona un modo de codificación de plano de bits para codificar información 5110 de decisión de modo de predicción hacia delante/no hacia delante. Después de que se selecciona el modo de codificación, el codificador codifica la información 5120 de decisión de modo de predicción hacia delante/no hacia delante en el modo seleccionado. El codificador selecciona modos de codificación de plano de bits en una base de campo a campo. Como alternativa, el codificador selecciona modos de codificación de plano de bits en alguna otra base (por ejemplo, a nivel de secuencia). O, si únicamente se usa un modo de codificación de plano de bits, no se realiza selección de modos de codificación de plano de bits. Cuando el codificador está haciendo la codificación de información 5130 de decisión de modo de predicción hacia delante/no hacia delante, la codificación de la información de decisión de modo de predicción hacia delante/no hacia delante finaliza.

Con referencia a la Figura 52, el decodificador determina el modo de codificación de plano de bits usado mediante (y señalado mediante) el codificador para codificar la información 5210 de decisión de modo de predicción hacia delante/no hacia delante. El decodificador a continuación decodifica la información 5220 de decisión de modo de predicción hacia delante/no hacia delante en el modo seleccionado. El decodificador determina modos de codificación de plano de bits en una base campo a campo. Como alternativa, el decodificador determina modos de codificación de plano de bits en alguna otra base (por ejemplo, a nivel de secuencia). O, si únicamente está disponible un modo de codificación de plano de bits, no se realiza selección de modos de codificación de plano de bits. Cuando el decodificador está haciendo la codificación de la información 5230 de decisión de modo de predicción hacia delante/no hacia delante, la decodificación de la información de decisión de modo de predicción hacia delante/no hacia delante finaliza.

Para detalles adicionales sobre señalización y decodificación para los diversos modos de codificación de plano de bits de acuerdo con varias implementaciones combinadas, véase la sección XIV a continuación. Para más información sobre codificación de plano de bits en general, véase la Solicitud de Patente de Estados Unidos con N.º de Serie 10/321.415, titulada, "Skip Macroblock Coding", presentada el 16 de diciembre de 2002, la divulgación de la cual se incorpora por la presente por referencia. Como alternativa, los bits que representan la información de modo hacia delante/no hacia delante pueden enviarse sin comprimir y/o a algún otro nivel (por ejemplo, a nivel de macrobloque).

Si se indica predicción no hacia delante, el codificador especifica el modo de predicción no hacia delante (por ejemplo, modo hacia atrás, modo directo, modo interpolado o intra modo) para el macrobloque. En algunas realizaciones, un codificador codifica modos de predicción no hacia delante a nivel de macrobloque con referencia a una tabla de VLC, como se muestra en la Tabla 2 a continuación.

Tabla 2: Tabla de VLC de modo de predicción de movimiento

VLC BMVTYPE	Modo de predicción de movimiento
0	Hacia atrás
10	Directo
11	Interpolado

En el ejemplo mostrado en la Tabla 2, el modo hacia atrás es el modo de predicción no hacia delante preferido. El codificador representa modo hacia atrás con una señal de un bit y representa modos directo e interpolado con señales de dos bits. Como alternativa, el codificador usa diferentes códigos para representar diferentes modos de predicción y/o prefiere un modo de predicción no hacia delante diferente.

En algunas realizaciones, el modo intra se señala mediante un valor de vector de movimiento diferencial especial, que indica mediante la manera en la que se codifica que el modo de predicción es el modo intra. El valor de vector de movimiento diferencial se usa por lo tanto para inferir que el macrobloque está intra codificado, pero como una convención el codificador establece el tipo de predicción a hacia atrás, para no tener ningún tipo de predicción no definido.

XII. Selección de vectores de movimiento co-localizados para modo directo en campos B entrelazados

En algunas implementaciones, los vectores de movimiento de modo directo para macrobloques en imágenes B codificadas por campo se seleccionan usando lógica especial. Para un macrobloque actual en un campo B entrelazado, si el macrobloque co-localizado en el campo correspondiente de la siguiente imagen de ancla se codificó usando cuatro vectores de movimiento, la lógica favorece la polaridad más dominante (por ejemplo, la misma u opuesta) entre hasta los cuatro vectores de movimiento para el macrobloque co-localizado. Una vez que se selecciona un vector de movimiento para usar para el macrobloque actual, el codificador/decodificador puede aplicar operaciones de cambio de escala para proporcionar los vectores de movimiento de modo directo.

En algunas implementaciones, para un macrobloque de 1 MV de modo directo de un campo B entrelazado, un codificador/decodificador calcula el vector de movimiento para usar para el cambio de escala de modo directo basándose en uno o más vectores de movimiento para el macrobloque co-localizado en un campo de referencia (por ejemplo, el campo P temporalmente siguiente) con la misma polaridad. Si el macrobloque co-localizado en el campo de referencia es un macrobloque de 1 MV, el codificador/decodificador usa el vector de movimiento único para derivar vectores de movimiento de modo directo para el macrobloque en el campo B entrelazado. Si, por otra parte, el macrobloque co-localizado en el campo de referencia es un macrobloque de 4MV, el codificador/decodificador considera las polaridades de los cuatro vectores de movimiento, que favorece la polaridad dominante, al seleccionar el vector de movimiento usado para derivar los vectores de movimiento de modo directo para el macrobloque en el campo B entrelazado. El codificador/decodificador puede aplicar la lógica de selección a macrobloques de 4MV en el campo de referencia cuando sea necesario durante la decodificación del campo B entrelazado. O, el codificador/decodificador puede aplicar la lógica de selección después de la decodificación del campo de referencia, almacenando en memoria intermedia a continuación únicamente el valor a usarse en la decodificación de campo B entrelazado posterior.

Por ejemplo, para el macrobloque de 4MV co-localizado en el campo de referencia, si el número de vectores de movimiento (de cuatro) de campo de la misma polaridad superan en número a aquellos del campo de la polaridad opuesta, un codificador/decodificador puede usar mediana de 4, mediana de 3, media aritmética de 2 o los valores del vector de movimiento de campo de la misma polaridad para calcular un vector de movimiento para uso en decodificación de campo B entrelazado de modo directo, si el número de vectores de movimiento de la misma polaridad es cuatro, tres, dos o uno, respectivamente. De otra manera, si los vectores de movimiento desde el campo de polaridad opuesta superan en número a aquellos del campo de la misma polaridad, un codificador/decodificador puede usar operaciones similares para obtener un vector de movimiento representativo desde los vectores de movimiento de campo de polaridad opuesta para uso en decodificación de campo B entrelazado de modo directo. Si más de dos del conjunto original de cuatro vectores de movimiento para el macrobloque co-localizado (independientemente de la polaridad) son intra, entonces el codificador/decodificador puede simplemente tratar el vector de movimiento representativo co-localizado como intra (es decir, (0, 0)). Sin embargo, en algunas implementaciones, todos los MB intra en campos B entrelazados se codifican como 1MV, por lo que el caso cuando más de dos de unos cuatro vectores de movimiento originales que son intra conduce a que el vector de movimiento representativo co-localizado se trate como intra nunca surge en la práctica.

El pseudocódigo 5300 en la Figura 53 muestra un procedimiento de selección para un vector de movimiento para usar como una base para vectores de movimiento de modo directo en campos B entrelazados. En algunas implementaciones, este procedimiento de selección es un pre-cursor a operaciones de cambio de escala que producen vectores de movimiento de modo directo que apuntan hacia delante y hacia atrás.

XIII. Campos B intra-codificados en fotogramas de vídeo entrelazados

Un campo BI entrelazado (o "intra campo B") es un campo que está codificado de manera independiente de sus imágenes de referencia. Los campos BI entrelazados son diferentes de otros campos intra (por ejemplo, campos I entrelazados) en el sentido en el que no pueden usarse como anclas para predecir otras imágenes. No hay dependencia inter-imagen en el campo BI entrelazado, y su presencia en un flujo de bits no representa el inicio de un segmento o grupo de imágenes independientemente decodificables. Sin embargo, un primer campo en un fotograma de vídeo entrelazado, si se codifica como un campo BI, podría usarse para predecir el segundo campo en el fotograma, que puede codificarse como un campo B entrelazado. Esta innovación también mejora la compresión global usando únicamente intra codificación para la mitad del fotograma (el primer campo codificado) en muchos casos, en lugar de codificar un fotograma completo como un intra fotograma o codificar ambos campos como intra campos. En algunas implementaciones, un fotograma puede comprender dos campos B, dos campos BI o un campo B y un campo BI.

Hay razones para usar campos BI entrelazados en lugar de campos I entrelazados. Una es evitar sacrificar la escalabilidad temporal. Por ejemplo, cuando un decodificador está representando vídeo digital y necesita descartar inmediatamente algunas imágenes para mantener las demandas de procesamiento, puede mirar una secuencia de campos que pueden potencialmente descartar. Si los campos intra en la secuencia resultan ser campos clave, el decodificador se verá forzado a decodificarlos para usarlos como referencias para otros campos, y no puede descartarlos. Sin embargo, si los campos intra en la secuencia se codifican como campos BI, el decodificador tendrá aún la opción de descartarlos sin comprometer compensación de movimiento posterior.

Los campos BI entrelazados son diferentes a los campo B entrelazados con intra macrobloques en el sentido en el que los campos BI entrelazados señalizan más eficazmente los elementos de sintaxis usados para intra codificación y decodificación, puesto que los elementos relacionados con compensación de movimiento dentro de un campo BI (o los elementos que señalizan la ausencia de los mismos) pueden evitarse. En otras palabras, una razón para usar campos BI entrelazados (en lugar de campos B normales) surge cuando se codifica un campo B entrelazado en un punto en una secuencia de vídeo donde la inter-predicción de campo se descompone (por ejemplo, debido a cambios de escena o movimiento complejo). A menudo, la mayoría de los macrobloques en un campo de este tipo necesitarán codificarse como macrobloques intra. En este escenario, a menudo es más económico en términos de tasa de bits codificar todo el campo B como un campo BI que enviar la información de modo de predicción para cada macrobloque en el campo. Cuando no es posible buena predicción o compensación de movimiento para un campo B entrelazado, puede codificarse como un campo BI.

En algunas implementaciones, un codificador puede señalar la aparición de un campo BI en el flujo de bits como uno de los posibles valores del tipo de imagen. Como alternativa, la presencia de un campo BI puede indicarse de alguna otra manera.

XIV. Implementaciones combinadas

Una implementación combinada detallada para una sintaxis de flujo de bits, semántica y el decodificador se describen ahora, además de una implementación combinada alternativa con diferencias menores de la implementación combinada principal.

A. Sintaxis de flujo de bits

En diversas implementaciones combinadas, los datos para imágenes B entrelazadas se presentan en forma de un flujo de bits que tiene varias capas (por ejemplo, capas de secuencia, fotograma, campo, macrobloque, bloque y/o sub-bloque).

Para fotogramas de vídeo entrelazados con campos B entrelazados y/o campos BI, se muestran elementos de flujo de bits de nivel de fotograma en la Figura 54. Los datos para cada fotograma consisten en un encabezamiento de fotograma seguido por datos para las capas de campo (mostradas como el elemento por campo repetido "FieldPicLayer"). Los elementos de flujo de bits que componen los encabezamientos de campo para campos B entrelazados y campos BI se muestran en las Figuras 55 y 56, respectivamente. Los elementos de flujo de bits que componen la capa de macrobloque para campos B entrelazados (ya sea para macrobloques intra, 1MV o 4MV) y campos BI se muestran en las Figuras 57 y 58, respectivamente.

Para fotogramas B entrelazados, los elementos de flujo de bits de nivel de fotograma se muestran en la Figura 59. Los datos para cada fotograma consisten en un encabezamiento de fotograma seguido por datos para la capa de macrobloque. Los elementos de flujo de bits que componen la capa de macrobloque para fotogramas B entrelazados (ya sea para macrobloques de tipo intra o diversos inter) se muestran en la Figura 60.

Las siguientes secciones describen elementos de flujo de bits seleccionados en las capas de fotograma, campo y macrobloque que están relacionadas con señalización relacionada con imágenes entrelazadas previstas bidireccionalmente. Aunque los elementos de flujo de bits seleccionados se describen en el contexto de una capa particular, algunos elementos de flujo de bits pueden usarse en más de una capa.

1. Elementos de capa de fotograma seleccionados

La Figura 54 es un diagrama que muestra una sintaxis de flujo de bits de nivel de fotograma para fotogramas que contienen campos B entrelazados y/o campos BI (o potencialmente otras clases de campos entrelazados). La Figura 59 es un diagrama que muestra una sintaxis de flujo de bits de nivel de fotograma para fotogramas B entrelazados. Se describen a continuación elementos de flujo de bits específicos.

Modo de codificación de fotograma (FCM) (Tamaño variable)

FCM es una palabra de código de longitud variable ["VLC"] usada para indicar el tipo de codificación de imagen. FCM toma valores para modos de codificación de fotograma como se muestra en la Tabla 3 a continuación:

Tabla 3: VLC de modo de codificación de fotograma

valor FCM	Modo de codificación de fotograma
0	Progresivo
10	Fotograma entrelazado
11	Campo entrelazado

Tipo de imagen de campo (FPTYPE) (3 bits)

FPTYPE es un elemento de sintaxis de tres bits presente en el encabezamiento de fotograma para un fotograma que incluye campos P entrelazados, campos I entrelazados, campos B entrelazados y/o campos BI entrelazados. FPTYPE toma valores para diferentes combinaciones de tipos de campo en el fotograma de vídeo entrelazado, de acuerdo con la Tabla 4 a continuación.

5

Tabla 4: FLC de tipo de imagen de campo

FLC FPTYPE	Primer tipo de campo	Segundo tipo de campo
000	I	I
001	I	P
010	P	I
011	P	P
100	B	B
101	B	BI
110	BI	B
111	BI	BI

Distancia de referencia (REFDIST) (Tamaño variable)

REFDIST es un elemento de sintaxis con tamaño variable. Este elemento indica el número de fotogramas entre el fotograma actual y el fotograma de referencia. La Tabla 5 muestra el VLC usada para codificar los valores de REFDIST.

10

Tabla 5: tabla de VLC de REFDIST

Distancia de fotograma de referencia	Palabra de código de VLC (Binario)	Tamaño de VLC
0	00	2
1	01	2
2	10	2
N	11[(N-3) números 1]0	N

La última fila en la tabla 5 indica las palabras de código usadas para representar distancias de fotograma de referencia mayores de 2. Estas se codifican como (binario) 11 seguido por N-3 números 1, donde N es la distancia de fotograma de referencia. El último bit en la palabra de código es 0. Por ejemplo: N = 3,

15

N = 3, palabra de código de VLC = 110, tamaño de VLC = 3
 N = 4, palabra de código de VLC = 1110, tamaño de VLC = 4
 N = 5, palabra de código de VLC = 11110, tamaño de VLC = 5

20 *Tipo de imagen (PTYPE) (Tamaño variable)*

PTYPE es un elemento de sintaxis de tamaño variable presente en el encabezamiento de fotograma para un fotograma B entrelazado (u otra clase de fotograma entrelazado tal como fotograma I entrelazado o fotograma P entrelazado). PTYPE toma valores para diferentes tipos de fotograma de acuerdo con la Tabla 6 a continuación.

Tabla 6: VLC de tipo de imagen

VLC PTYPE	Tipo de imagen
110	I
0	P
10	B
1110	BI
1111	Saltada

25

Si PTYPE indica que el fotograma está saltado entonces el fotograma se trata como un fotograma P que es idéntico

a su fotograma de referencia. La reconstrucción del fotograma saltado es equivalente conceptualmente a copiar el fotograma de referencia. Un fotograma de saltado significa que no se transmiten datos adicionales para este fotograma.

Elemento de sintaxis de bit de MB de modo directo de fotograma B (DIRECTMB) (Tamaño variable)

- 5 El elemento de sintaxis DIRECTMB usa codificación de plano de bits para indicar los macrobloques en la imagen B (en este punto, un fotograma B entrelazado) que se codifica en modo directo. El elemento de sintaxis DIRECTMB puede señalar también que el modo directo está señalado en modo sin procesar, caso en el que el modo directo se señala en el nivel de macrobloque para los macrobloques del fotograma B entrelazado.

Bandera de intervalo de MV extendido (MVRANGE) (Tamaño variable)

- 10 MVRANGE es un elemento de sintaxis con tamaño variable presente cuando el bit de capa de secuencia EXTENDED_MV se establece a 1. El VLC MVRANGE representa un intervalo de vector de movimiento.

Bandera de intervalo de MV diferencial extendido (DMVRANGE) (Tamaño variable)

- 15 DMVRANGE es un elemento de sintaxis con tamaño variable presente si el elemento de sintaxis de nivel de secuencia EXTENDED_DMV = 1. El VLC de DMVRANGE representa un intervalo de vector de movimiento diferencial.

Tabla de modo de macrobloque (MBMODETAB) (2 o 3 bits)

El elemento de sintaxis MBMODETAB es un campo de longitud fija. Para campos P entrelazados, MBMODETAB es un valor de 3 bits que indica cuál de las ocho tablas de Huffman se usa para decodificar el elemento de sintaxis de modo de macrobloque (MBMODE) en la capa de macrobloque.

- 20 *Tabla de vector de movimiento (MVTAB) (2 o 3 bits)*

El elemento de sintaxis MVTAB es un valor de 2 o 3 bits. Para campos P entrelazados en los que NUMREF = 1, MVTAB es un elemento de sintaxis de 3 bits que indica cuál de las ocho tablas de Huffman entrelazadas se usa para decodificar los datos de vector de movimiento.

Tabla de patrón de bloque de 2MV (2MVBPTAB) (2 bits)

- 25 El elemento de sintaxis 2MVBPTAB es un valor de 2 bits que señala cuál de las cuatro tablas de Huffman se usa para decodificar el elemento de sintaxis de patrón de bloque de 2MV (2MVBPTAB) en macrobloques de campo de 2MV.

Tabla de patrón de bloque de 4MV (4MVBPTAB) (2 bits)

El elemento de sintaxis 4MVBPTAB es un valor de 2 bits que señala cuál de las cuatro tablas de Huffman se usa para decodificar el elemento de sintaxis de patrón de bloque de 4MV (4MVBPTAB) en macrobloques de 4MV.

- 30 En una implementación combinada alternativa, la información de tipo de imagen se señala al comienzo del nivel de campo para un campo B entrelazado, en lugar de al nivel de fotograma para el fotograma de vídeo entrelazado que incluye el campo B entrelazado, y se omite la distancia de referencia.

2. Elementos de capa de campo seleccionados

- 35 La Figura 55 es un diagrama que muestra una sintaxis de flujo de bits de nivel de campo para campos B entrelazados en la implementación combinada. Se describen a continuación elementos de flujo de bits específicos.

Modo de vector de movimiento (MVMODE) (Tamaño variable o 1 bit)

- 40 El elemento de sintaxis MVMODE señala uno de cuatro modos de codificación de vector de movimiento o un modo de compensación de intensidad (menos posibilidades para algunos tipos de imágenes). Varios elementos posteriores proporcionan información de modo de vector de movimiento y/o de compensación de intensidad adicional.

Elemento de sintaxis de bit de MB de modo hacia delante de campo B (FORWARDMB) (Tamaño variable)

- 45 El elemento de sintaxis FORWARDMB usa codificación de plano de bits para indicar los macrobloques en el campo B que están codificados en modo hacia delante. El elemento de sintaxis FORWARDMB puede señalar también que el modo hacia delante se señala en modo sin procesar, caso en el que la decisión de modo hacia delante/no hacia delante se señala en el nivel de macrobloque.

La Figura 56 es un diagrama que muestra una sintaxis de flujo de bits de nivel de campo para campos BI entrelazados en la implementación combinada. En la implementación combinada, la sintaxis de flujo de bits de nivel de campo para campos BI entrelazados usa los mismos elementos de sintaxis que los campos I entrelazados.

3. Elementos de capa de macrobloque seleccionados

La Figura 57 es un diagrama que muestra una sintaxis de flujo de bits de nivel de macrobloque para macrobloques de campos B entrelazados en la implementación combinada. La Figura 60 es un diagrama que muestra una sintaxis de flujo de bits de nivel de macrobloque para fotogramas B entrelazados de macrobloques en la implementación combinada. Se describen a continuación elementos específicos de flujo de bits. Los datos para un macrobloque consisten en un encabezamiento de macrobloque seguido por datos de capa de bloque.

Modo de macrobloque (MBMODE) (Tamaño variable)

El elemento de sintaxis MBBMODE indica el tipo de macrobloque (por ejemplo, 1 MV, 4MV o intra para campos B entrelazados) y también la presencia de la bandera CBP y los datos de vector de movimiento.

Modo de codificación de campo B hacia delante (FORWARDBIT) (1 bit)

FORWARDBIT es un elemento de sintaxis de 1 bit presente en macrobloques de campos B entrelazados si el elemento de sintaxis de nivel de campo FORWARDMB indica que se usa el modo sin procesar. Si FORWARDBIT = 1, entonces el macrobloque se codifica usando el modo hacia delante.

Tipo de predicción de movimiento de macrobloque B (BMVTYPE) (Tamaño variable)

BMVTYPE es un elemento de sintaxis con tamaño variable presente en macrobloques de fotogramas B entrelazados y macrobloques de campos B entrelazados que indica si el macrobloque usa predicción hacia delante, hacia atrás o interpolada. Como muestra la Tabla 7, para los macrobloques de fotogramas B entrelazados, el valor de BFRACTION junto con BMVTYPE determina qué tipo se usa.

Tabla 7: VLC BMVTYPE

BMVTYPE	Tipo de predicción de movimiento	
	BFRACTION $\leq$ 1/2	BFRACTION $>$ 1/2
0	Hacia detrás	Hacia delante
10	Hacia delante	Hacia detrás
11	Interpolado	Interpolado

En campos B entrelazados, BMVTYPE se envía si el modo de macrobloque no es hacia delante (como se indica mediante elemento de sintaxis FORWARDMB o FORWARDBIT) y no se está usando 4MV. En este caso, se usa BMVTYPE para señalar si el macrobloque B es hacia atrás, directo o interpolado. Esto es un VLC sencillo, donde hacia atrás = 0, directo = 10, interpolado = 11. En el caso donde el modo de macrobloque no sea hacia delante y 4MV esté en uso, BMVTYPE es hacia atrás puesto que únicamente están permitidos modos hacia delante y hacia atrás con 4MV.

MV presente interpolado (INTERPMVP) (1 bit)

INTERPMVP es un elemento de sintaxis de 1 bit presente en macrobloques de campo B si el elemento de sintaxis BMVTYPE de nivel de campo indica que el tipo de macrobloque es interpolado. Si INTERPMVP = 1, entonces el MV interpolado está presente, de otra manera no está presente.

Vector de movimiento de macrobloque B 1 (BMV1) (Tamaño variable)

BMV1 es un elemento de sintaxis con tamaño variable que codifica diferencialmente el primer vector de movimiento para el macrobloque.

Vector de movimiento de macrobloque B 2 (BMV2) (Tamaño variable)

BMV2 es un elemento de sintaxis de tamaño variable presente en macrobloques de fotograma B entrelazado y macrobloques de campo B entrelazado si se usa el modo interpolado. Este elemento de sintaxis codifica diferencialmente el segundo vector de movimiento para el macrobloque.

Patrón de bloque de 4MV (4MVBP) (4 bits)

El elemento de sintaxis 4MVBP indica cuál de los 4 bloques de luminancia contiene vectores de movimiento diferenciales distintos de cero, el uso del cuál se describe en detalle a continuación.

Datos de vector de movimiento de nivel de bloque (BLKMVDA TA) (Tamaño variable)

BLKMVDATA es un elemento de sintaxis con tamaño variable que contiene información de movimiento para el

bloque, y está presente en macrobloques de 4MV.

Bandera de transformación de campo (FIELDTX) (1 bit)

5 FIELDTX es una sintaxis de 1 bit presente en macrobloques de fotograma B entrelazados intra-codificados. Este elemento de sintaxis indica si un macrobloque está codificado por fotograma o por campo (básicamente, la organización interna del macrobloque). FIELDTX = 1 indica que el macrobloque está codificado por campo. De otra manera, el macrobloque está codificado por fotograma. En macrobloques inter-codificados, este elemento de sintaxis puede inferirse a partir de MBMODE.

Modo de codificación de fotograma B directo (DIRECTBBIT) (1 bit)

10 DIRECTBBIT es un elemento de sintaxis de 1 bit presente en macrobloques de fotograma B entrelazado si la sintaxis de nivel de fotograma DIRECTMB indica que se usa el modo sin procesar. Si DIRECTBBIT = 1, entonces el macrobloque se codifica usando el modo directo.

Conmutación de MV de fotograma B (MVSW) (1 bit)

15 MVSW es un elemento de sintaxis de 1 bit presente en macrobloques de fotograma B entrelazado si el MB es en modo de campo y si el BMVTYPE es hacia delante o hacia atrás. Si MVSW = 1, entonces el tipo de MV y tipo de predicción cambian de hacia delante a hacia atrás (o de hacia atrás a hacia delante) al pasar desde el campo superior al inferior.

Patrón de bloque de dos vectores de movimiento (2MVBP) (Tamaño variable)

20 2MVBP es un elemento de sintaxis con tamaño variable presente en macrobloques de fotograma B entrelazado. Este elemento de sintaxis está presente si el elemento de sintaxis MBMODE indica que el macrobloque contiene un vector de movimiento, y si el macrobloque es un macrobloque interpolado. En este caso, 2MVBP indica cuál de los dos vectores de movimiento (vectores de movimiento hacia delante y hacia atrás) están presentes.

Datos de vector de movimiento (MVDATA) (Tamaño variable)

MVDATA es un elemento de sintaxis con tamaño variable que codifica diferenciales para el vector o vectores de movimiento para el macrobloque, la decodificación de los cuales se describe en detalle a continuación.

25 La Figura 58 es un diagrama que muestra una sintaxis de flujo de bits de nivel de macrobloque para campos BI entrelazados en la implementación combinada. En la implementación combinada, la sintaxis de flujo de bits de nivel de macrobloque para campos BI entrelazados usa los mismos elementos de sintaxis que los campos I entrelazados.

B. Decodificar campos B entrelazados

30 Las siguientes secciones describen un procedimiento para decodificar campos B entrelazados en la implementación combinada.

1. Decodificación de capa de fotograma/campo

Los campos B entrelazados pueden ser uno de dos tipos: de 1 MV o MV mixto.

35 En campos B entrelazados de 1 MV, se usa cualquiera de cero, uno o dos vectores de movimiento para indicar el desplazamiento de los bloques previstos, dependiendo del tipo de predicción (BMVTYPE) de ese macrobloque. Cuando BMVTYPE es igual a DIRECTO, los vectores de movimiento hacia delante y hacia atrás se infieren y no se señalizan explícitamente vectores de movimiento adicionales. Cuando BMVTYPE es INTERPOLADO, se decodifican dos vectores de movimiento: hacia delante y hacia atrás. En los casos hacia delante y hacia atrás, únicamente se decodifica un vector de movimiento. El modo de 1 MV se señala mediante los elementos de sintaxis de capa de imagen de MVMODE.

40 En campos B entrelazados de MV mixto, cada macrobloque puede codificarse como un macrobloque de 1MV o de 4MV. En macrobloques de 4MV, cada uno de los 4 bloques de luminancia tiene un vector de movimiento asociado con él. Adicionalmente, pueden asociarse únicamente macrobloques de 4MV con tipos de predicción hacia delante o hacia atrás (BMVTYPE) en campos B entrelazados. El modo de 1 MV o 4MV para cada macrobloque se indica mediante el elemento de sintaxis MBMODE en cada macrobloque. El modo de MV mixto se señala mediante los elementos de sintaxis de capa de imagen de MVMODE.

2. Decodificación de capa de macrobloque

50 Los macrobloques en campos B entrelazados pueden ser uno de tres posibles tipos: 1 MV, 4MV e intra. Adicionalmente, los macrobloques pueden ser uno de cuatro tipos de predicción (BMVTYPE): hacia delante, hacia atrás, directa o interpolada. El tipo de macrobloque se señala mediante el elemento de sintaxis MBMODE en la capa de macrobloque. El tipo de predicción se señala mediante una combinación del plano de bits de nivel de

fotograma FORWARDMB que señala hacia delante/no hacia delante para cada macrobloque, y el elemento de sintaxis BMVTYPE de nivel de macrobloque en el caso en el que el tipo de predicción sea no hacia delante.

Las siguientes secciones describen los tipos de 1 MV y 4MV y cómo se señalizan.

Macrobloques de 1 MV en campos B entrelazados

5 Los macrobloques de 1MV pueden tener lugar en campos B entrelazados de 1MV y de MV mixto. En un macrobloque de 1 MV, un único vector de movimiento representa el desplazamiento entre las imágenes actual y de referencia para todos los seis bloques en el macrobloque. Para un macrobloque de 1 MV, el elemento de sintaxis MBMODE en la capa de macrobloque indica tres cosas:

- 1) Que el tipo de macrobloque es 1 MV
- 10 2) Si el elemento de sintaxis CBPCY está presente
- 3) Si el elemento de sintaxis BMV1 está presente

Si el elemento de sintaxis MBMODE indica que el elemento de sintaxis BMV1 está presente, entonces el elemento de sintaxis BMV1 está presente en la capa de macrobloque en la posición correspondiente. El elemento de sintaxis BMV1 codifica el vector de movimiento diferencial. El vector de movimiento diferencial se combina con el predictor de vector de movimiento para reconstruir el vector de movimiento. Si el elemento de sintaxis MBMODE indica que el elemento de sintaxis BMV1 no está presente, entonces el vector de movimiento diferencial se supone que es cero y por lo tanto el vector de movimiento es igual al predictor de vector de movimiento.

Si el elemento de sintaxis MBMODE indica que el elemento de sintaxis CBPCY está presente, entonces el elemento de sintaxis CBPCY está presente en la capa de macrobloque en la posición correspondiente. CBPCY indica cuál de los seis bloques están codificados en la capa de bloque. Si el elemento de sintaxis MBMODE indica que CBPCY no está presente, entonces se supone que CBPCY es igual a 0 y ningún dato de bloque está presente para ninguno de los seis bloques en el macrobloque.

Adicionalmente, si el tipo de macrobloque es 1 MV y el tipo de predicción del macrobloque es interpolado, el codificador usa el elemento de sintaxis INTERPMVP para señalar si está presente o no el segundo vector de movimiento diferencial, BMV2. Si está presente, el decodificador decodifica BMV2 inmediatamente después de BMV1. De otra manera, el vector de movimiento diferencial para BMV2 se supone que es cero, y el segundo vector de movimiento es igual al predictor de vector de movimiento.

Cuando el tipo de predicción es interpolado, BMV1 corresponde al vector de movimiento hacia delante y BMV2 corresponde al vector de movimiento hacia atrás.

30 *Macrobloques de 4MV en campos B entrelazados*

Los macrobloques de 4MV pueden tener lugar únicamente en imágenes de campo B de MV mixto y se restringen a tipos de predicción hacia delante y hacia atrás. En un macrobloque de 4MV, cada uno de los 4 bloques de luminancia tiene un vector de movimiento asociado. El desplazamiento para los bloques de crominancia se deriva desde los 4 vectores de movimiento de luminancia. En campos B entrelazados de MV mixto, los macrobloques de 4MV pueden únicamente asociarse con tipos de predicción hacia delante y hacia atrás.

Para un macrobloque de 4MV, el elemento de sintaxis MBMODE en la capa de macrobloque indica tres cosas:

- 1) Que el tipo de macrobloque es 4MV
- 2) Si el elemento de sintaxis CBPCY está presente
- 3) Si el elemento de sintaxis 4MVBP está presente

40 El elemento de sintaxis 4MVBP indica cuál de los 4 bloques de luminancia contienen vectores de movimiento diferenciales distintos de cero. El elemento de sintaxis 4MVBP decodifica a un valor entre 0 y 15. Para cada una de las posiciones de 4 bits en el 4MVBP, un valor de 0 indica que ningún vector de movimiento diferencial (BLKMVDATA) está presente para ese el bloque y que el vector de movimiento diferencial se supone que es 0. Un valor de 1 indica que un vector de movimiento diferencial (BLKMVDATA) está presente para ese bloque en la posición correspondiente. Por ejemplo, si 4MVBP se decodifica a un valor de 1100 (binario), entonces el flujo de bits contiene BLKMVDATA para los bloques 0 y 1 y BLKMVDATA no está presente para los bloques 2 y 3.

Si el elemento de sintaxis MBMODE indica que el elemento de sintaxis 4MVBP no está presente, entonces se supone que los datos de vector de movimiento diferencial (BLKMVDATA) están presentes para todos los 4 bloques de luminancia.

50 Dependiendo de si el elemento de sintaxis MVMODE indica MV mixto o todo 1M el MBMODE señala la información como sigue. La Tabla 8 a continuación muestra cómo el elemento MBMODE señala información acerca de un macrobloque en una imagen todo 1MV.

Tabla 8: modo de macrobloque en imágenes todo 1 MV

Índice	Tipo de macrobloque	CBP presente	MV presente
0	Intra	No	NA
1	Intra	Sí	NA
2	1MV	No	No
3	1MV	No	Sí
4	1MV	Sí	No
5	1MV	Sí	Sí

La Tabla 9 a continuación muestra cómo el elemento MBMODE señala información acerca de un macrobloque en una imagen de MV mixto.

5

Tabla 9: modo de macrobloque en imágenes 1MV mixta

Índice	Tipo de macrobloque	CBP Presente	MV Presente
0	Intra	No	NA
1	Intra	Sí	NA
2	1MV	No	No
3	1MV	No	Sí
4	1MV	Sí	No
5	1MV	Sí	Sí
6	4MV	No	NA
7	4MV	Sí	NA

Una de las 8 tablas de codificación se usa para señalar el MBMODE. La tabla específica usada se señala mediante el elemento de sintaxis MBMODETAB.

Las siguientes secciones describen decodificación de tipo de predicción y decodificación de vectores de movimiento de modo directo.

10

Decodificación de tipo de predicción (BMVTYPE) en campos B entrelazados

El tipo de predicción se decodifica de acuerdo con las siguientes reglas. Si el plano de bits de nivel de imagen FORWARDMB indica que un macrobloque es de tipo hacia delante, entonces el tipo de predicción para ese macrobloque se establece a hacia delante. Si el elemento FORWARDMB se codifica sin procesar, entonces un codificador/decodificador usa un bit adicional en el nivel de macrobloque, FORWARDBIT, para decidir si el tipo de predicción es o no hacia delante.

15

Si el tipo de predicción es no hacia delante, y si el macrobloque usa 4MV, según se señala mediante el elemento de sintaxis MBMODE (únicamente posible en una imagen B de MV mixto), entonces un decodificador puede inferir directamente que el tipo de predicción es hacia atrás, puesto que únicamente pueden asociarse tipos hacia delante y hacia atrás con modo de 4MV. De otra manera, el decodificador decodifica explícitamente el elemento de sintaxis BMVTYPE.

20

Decodificar vectores de movimiento de modo directo en campos B entrelazados

Para decodificar vectores de movimiento de modo directo en campos B entrelazados, un decodificador en primer lugar almacena en memoria intermedia vectores de movimiento desde la imagen de ancla (I o P) previamente decodificada (es decir, temporalmente futura). De estos, el decodificador usa los vectores de movimiento almacenados en memoria intermedia que corresponden al campo superior como predictores en orden para calcular los vectores de movimiento de modo directo en el campo B superior, y usa aquellos correspondientes al campo inferior para calcular los vectores de movimiento del campo B inferior. Por ejemplo, el macrobloque (x, y) en el campo z (z = superior/inferior) hará referencia a los vectores de movimiento almacenados en memoria intermedia desde el macrobloque (x, y) del campo z I o P previamente decodificado (es decir, un macrobloque co-localizado en un campo de ancla de la misma polaridad que el campo actual).

25

30

Si el vector de movimiento almacenado en memoria intermedia desde el imagen de ancla es intra (tal como cuando el campo z previamente decodificado es un campo I) o si la imagen de ancla es un campo P pero el macrobloque (x, y) está intra-codificado, el decodificador trata el vector de movimiento almacenado en memoria intermedia como (0, 0). Si el macrobloque co-localizado es 1 MV, el decodificador usa ese vector de movimiento. Si el macrobloque co-localizado es de 4MV, el decodificador usa la lógica descrita en el pseudocódigo 5300 en la Figura 53 para calcular el predictor de vector de movimiento.

En el pseudocódigo 5300, SelectDirectModeMVFromColocatedMB deriva el predictor de vector de movimiento para usar en cálculos de modo directo. El decodificador puede almacenar en memoria intermedia todos los vectores de movimiento desde la imagen de ancla previamente decodificada y a continuación aplicar las reglas de modo directo anteriores durante la decodificación de campos B, o el decodificador puede aplicar las reglas de modo directo anteriores mientras decodifica los campos de ancla y almacena en memoria intermedia los vectores de movimiento resultantes para los campos B.

Con el vector de movimiento obtenido anteriormente, el decodificador aplica lógica de cambio de escala (Scale_Direct_MV en la Figura 19). Scale_Direct_MV obtiene los vectores de movimiento que apuntan hacia delante y hacia atrás. Scale_Direct_MV puede dar como resultado vectores de movimiento hacia delante y hacia atrás que apuntan a ambos campos superior e inferior. Esto es eficaz puesto que los vectores de movimiento directo se evalúan mediante el codificador y se seleccionan únicamente cuando proporcionan buena predicción, y también puesto que los campos B entrelazados usan dos campos de referencia en ambas direcciones hacia delante y hacia atrás.

En una implementación alternativa, podría usarse cualquier otro procedimiento para generar los vectores de movimiento para cambiar de escala en modo directo, incluyendo procedimientos que no impliquen ningún almacenamiento en memoria intermedia, que puede ser útiles en dispositivos restringidos en memoria (por ejemplo, usando un generador de números aleatorio para simular una distribución de Laplace desviada a cero). Un procedimiento de este tipo funcionaría aún, puesto que un buen codificador descartaría malos intentos para vectores de movimiento de modo directo, dejando los más precisos en el flujo de bits.

3. Procedimiento de decodificación de vector de movimiento

Las siguientes secciones describen el procedimiento de decodificación de vector de movimiento para los bloques y macrobloques de campos B entrelazados en la implementación combinada.

Rellenar los contextos de predicción hacia delante y hacia atrás

Los vectores de movimiento hacia delante y hacia atrás se almacenan en memoria intermedia por separado y se usan para predecir vectores de movimiento hacia delante y hacia atrás, respectivamente. El uso de memorias intermedias separadas para contextos hacia delante y hacia atrás se describe, por ejemplo, en la sección X anterior. Las técnicas para selección de predictores de vector de movimiento se describen en los antecedentes en la sección III, la descripción detallada en la sección III, y en otras partes de la memoria descriptiva.

Hay dos detalles adicionales que añadir cuando se rellena la memoria intermedia hacia atrás (la memoria intermedia de “dirección faltante”) con el vector de movimiento previsto cuando se decodifica un vector de movimiento hacia delante (o rellenar la memoria intermedia hacia delante cuando se decodifica un vector de movimiento hacia atrás). Típicamente, un codificador/decodificador puede usar tanto la información de tipo de vector de movimiento (por ejemplo, 1 MV, etc.) así como la polaridad del vector de movimiento previamente decodificado para formar la predicción. Sin embargo, en el caso de “relleno de huecos”, un codificador/decodificador no tiene información de tipo de vector de movimiento o información de polaridad (por ejemplo, misma polaridad o polaridad opuesta) puesto que el codificador/decodificador no decodificó realmente un vector de movimiento del tipo de dirección faltante. En esta implementación combinada, el codificador/decodificador establece el tipo de vector de movimiento a 1 MV y elige el vector de movimiento de campo dominante como el predictor. El pseudocódigo 4700 en la Figura 47 describe el procedimiento de selección de polaridad en esta implementación combinada.

Para macrobloques intra codificados, el “vector de movimiento intra” se usa para relleno tanto en los planos de predicción de movimiento hacia delante como hacia atrás. Cualquier representación coherente de “vector de movimiento intra” puede elegirse mediante la implementación del decodificador. Por ejemplo, si los vectores de movimiento se están almacenando en una matriz corta de 2 bytes, entonces el “vector de movimiento intra” podría representarse como una constante larga única que se rellena en la matriz de vector de movimiento para indicar que el macrobloque se codificó como intra.

Predicción de vector de movimiento hacia delante en campos B

La distancia de fotograma de referencia hacia delante se calcula desde el elemento de sintaxis BFRACTION y desde el elemento de sintaxis REFDIST. La predicción de vector de movimiento hacia delante continúa como se ha descrito anteriormente en la sección X.

Predicción de vector de movimiento hacia atrás en campos B

La distancia de fotograma de referencia hacia atrás se calcula desde el elemento de sintaxis BFRAC_{ION} y desde el elemento de sintaxis REFDIST. La predicción de vector de movimiento hacia delante continúa como se ha descrito anteriormente en la sección X.

5 *Decodificar vectores de movimiento diferenciales*

Los elementos de sintaxis BMV1, BMV2 o BLKMV_{DATA} codifican información de movimiento para macrobloques o los bloques en el macrobloque. Los macrobloques de 1 MV tienen un elemento de sintaxis BMV1 y un BMV2, y los macrobloques de 4MV pueden tener entre cero y cuatro elementos BLKMV_{DATA}.

10 Cuando el tipo de predicción (BMV_{TYPE}) es interpolado, BMV1 corresponde al vector de movimiento residual hacia delante y BMV2 corresponde al de hacia atrás.

Las siguientes secciones describen cómo calcular el vector de movimiento diferencial para el caso de dos referencias que se aplica a imágenes B.

Vectores de movimiento diferenciales en imágenes de campo de dos referencias

15 Las imágenes de campo de dos referencias tienen lugar en la codificación de fotogramas entrelazados que usan imágenes de campo. Cada fotograma de la secuencia se separa en dos campos, y cada campo se codifica usando lo que es esencialmente la trayectoria de codificación progresiva.

En imágenes de campo que tienen dos campos de referencia (tal como imágenes con campos B entrelazados), cada elemento de sintaxis M_{DATA} o BLKMV_{DATA} en la capa de macrobloque codifica conjuntamente tres cosas: 1) el componente de vector de movimiento diferencial horizontal, 2) el componente de vector de movimiento diferencial vertical y 3) si se usa el predictor dominante o no dominante, es decir, cuál de los dos campos se hace referencia por el vector de movimiento.

25 El elemento de sintaxis M_{DATA} o BLKMV_{DATA} es una palabra de código de Huffman de longitud variable seguida por una palabra de código de longitud fija. El valor de la palabra de código de Huffman determina el tamaño de la palabra de código de longitud fija. El elemento de sintaxis MVTAB en la capa de imagen especifica la tabla de Huffman usada para decodificar la palabra de código con longitud variable. El pseudocódigo 6100 en la Figura 61A ilustra cómo se decodifica el vector de movimiento diferencial y la información de predictor dominante/no dominante.

Los valores de *predictor_flag*, *dmv_x* y *dmv_y* se calculan en el pseudocódigo 6100 en la Figura 61A. Los valores en el pseudocódigo 6100 se definen como sigue:

30 *dmv_x*: componente de vector de movimiento horizontal diferencial,
dmv_y: componente de vector de movimiento vertical diferencial,
k_x, *k_y*: longitud fija para vectores de movimiento largos,
k_x y *k_y* dependen del intervalo del vector de movimiento como se define por el símbolo MVRANGE.

Tabla 10: *k_x* y *k_y* especificados por MVRANGE

MVRANGE	<i>k_x</i>	<i>k_y</i>	range_x	Range_y
0 (por defecto)	9	8	256	128
10	10	9	512	256
110	12	10	2048	512
111	13	11	4096	1024
<i>extend_x</i> : intervalo extendido para vector de movimiento horizontal diferencial, <i>extend_y</i> : intervalo extendido para vector de movimiento vertical diferencial, <i>extend_x</i> y <i>extend_y</i> se derivan a partir del elemento de sintaxis del campo de imagen DMVRANGE. Si DMVRANGE indica que se usa el intervalo extendido para el componente horizontal, entonces <i>extend_x</i> = 1. De otra manera <i>extend_x</i> = 0. De manera similar, si DMVRANGE indica que se usa el intervalo extendido para el componente vertical, entonces <i>extend_y</i> = 1 de otra manera <i>extend_y</i> = 0.				

35 La variable *predictor_flag* es una bandera binaria que indica si se usa el predictor de vector de movimiento dominante o no dominante (0 = predictor dominante usado, 1 = predictor no dominante usado). La matriz *offset_table* y la matriz *size_table* se definen como se muestra en la Figura 61A.

El pseudocódigo 6110 en la Figura 61B ilustra cómo se decodifica el vector de movimiento diferencial para un campo de dos referencias en una implementación combinada alternativa. El pseudocódigo 6110 decodifica vectores de

movimiento diferenciales de una manera diferente. Por ejemplo, el pseudocódigo 6110 omite el manejo de los intervalos de vector de movimiento diferencial extendidos.

Predictores de vector de movimiento

5 Los vectores de movimiento se calculan añadiendo el vector de movimiento diferencial calculado en la sección anterior a un predictor de vector de movimiento. Las siguientes secciones describen cómo se calculan los predictores de vector de movimiento para macrobloques en campos B entrelazados de 1 MV y MV mixto en esta implementación combinada.

Predictores de vector de movimiento en campos B entrelazados de 1 MV

10 Las Figuras 5A y 5B son diagramas que muestran las localizaciones de macrobloques considerados para predictores de vector de movimiento candidato para un macrobloque de 1 MV. Los predictores candidatos se toman desde el macrobloque izquierdo, superior y superior derecho, excepto en el caso donde el macrobloque es el último macrobloque en la fila. En este caso, el Predictor B se toma desde el macrobloque superior izquierdo en lugar del superior-derecho. Para el caso especial donde el fotograma es de un macrobloque de ancho, el predictor es siempre el Predictor A (el predictor superior). Los casos especiales para el macrobloque actual que está en la fila superior (con predictores no A y B, o con ningún predictor en absoluto) se han tratado anteriormente con referencia a las Figuras 33A-33F y a continuación con referencia a las Figuras 62A-62F.

Predictores de vector de movimiento en campos B entrelazados de MV mixto

20 Las Figuras 6A-10 muestran las localizaciones de los bloques o macrobloques considerados para los vectores de movimiento candidatos para un vector de movimiento para un macrobloque de 1 MV o 4MV en campos B entrelazados de MV mixto.

Predictores de MV dominante y no dominante en campos B entrelazados

25 Para cada macrobloque inter-codificado, se derivan dos predictores de vector de movimiento. Uno proviene del campo dominante y el otro proviene del campo no dominante. El campo dominante se considera que es el campo que contiene la mayoría de los candidatos de predictor de vector de movimiento de valor real en las cercanías. En el caso de un empate, el predictor de vector de movimiento para el campo opuesto se considera que es el predictor dominante (puesto que es temporalmente más cercano). Los macrobloques intra-codificados no se consideran en el cálculo del predictor dominante/no dominante. Si todos los macrobloques de predictor candidatos están intra-codificados, entonces los predictores dominante y no dominante de vector de movimiento se establecen a cero y el predictor dominante se toma para que sea desde el campo opuesto.

Calcular el predictor de vector de movimiento en campos B entrelazados

30 Se calculan dos predictores de vector de movimiento para cada vector de movimiento de un bloque o macrobloque, uno para cada referencia. El pseudocódigo 6200 en las Figuras 62A-62F describe cómo se calculan los predictores de vector de movimiento para el caso de dos referencias en la implementación combinada. (El pseudocódigo 3300 en las Figuras 33A-33F describe cómo se calculan los predictores de vector de movimiento para el caso de dos referencias en otra implementación). En imágenes de dos referencias el campo actual puede hacer referencia a los dos campos más recientes. Un predictor es para el campo de referencia de la misma polaridad y el otro es para el campo de referencia con la polaridad opuesta.

Reconstruir vectores de movimiento en campos B entrelazados

40 Las siguientes secciones describen cómo reconstruir los vectores de movimiento de luminancia y crominancia para macrobloques de 1 MV y de 4MV. Después de que se reconstruye un vector de movimiento, puede usarse posteriormente como un vector de movimiento vecino para predecir el vector de movimiento para un macrobloque cercano. El vector de movimiento tendrá una polaridad asociada de "misma" u "opuesta", y puede usarse para derivar un predictor de vector de movimiento para la polaridad del otro campo para predicción de vector de movimiento.

Reconstrucción de Vector de movimiento de luminancia en campos B entrelazados

En todos los casos (macrobloques de 1MV y de 4MV) el vector de movimiento de luminancia se reconstruye añadiendo el diferencial al predictor como sigue:

$$mv_x = (dmv_x + predictor_x) \text{ smod range_x}$$

$$mv_y = (dmv_y + predictor_y) \text{ smod range_y}$$

50 La operación módulo "smod" es un módulo con signo, que se define como sigue:

$$A \text{ smod } b = ((A + b) \% (2*b)) - b$$

Esto asegura que los vectores reconstruidos son válidos. (A smod b) radica dentro de -b y b-1. range_x y range_y dependen de MVRANGE.

Puesto que la imagen de campo B entrelazado usa dos imágenes de referencia, *predictor_flag* derivada después de decodificar el vector de movimiento diferencial se combina con el valor de *dominantpredictor* derivado de la predicción de vector de movimiento para determinar qué campo se usa como referencia. El pseudocódigo 6300 en la Figura 63 describe cómo se determina el campo de referencia.

En macrobloques de 1 MV habrá un único vector de movimiento para los cuatro bloques que componen el componente de luminancia del macrobloque. Si el elemento de sintaxis MBMODE indica que no están presentes datos de MV en la capa de macrobloque, entonces $dmv_x = 0$ y $dmv_y = 0$ ($mv_x = predictor_x$ y $mv_y = predictor_y$).

En macrobloques de 4MV, cada uno de los bloques de luminancia inter-codificados en un macrobloque tendrá su propio vector de movimiento. Por lo tanto habrá entre cero y cuatro vectores de movimiento de luminancia en cada macrobloque de 4MV. Si el elemento de sintaxis 4MVBP indica que no está presente información de vector de movimiento para un bloque, entonces $dmv_x = 0$ y $dmv_y = 0$ para ese el bloque ($mv_x = predictor_x$ y $mv_y = predictor_y$).

Reconstrucción de vector de movimiento de crominancia

Los vectores de movimiento de crominancia se derivan desde los vectores de movimiento de luminancia. También, para macrobloques de 4MV, la decisión sobre si codificar los bloques de crominancia como Inter o Intra se realiza basándose en el estado de los bloques o campos de luminancia.

C. Decodificar fotogramas P entrelazados

Antes de describir el procedimiento para decodificar fotogramas B entrelazados en la implementación combinada, se describe un procedimiento para decodificar fotogramas P entrelazados. La sección que describe el procedimiento para decodificar fotogramas B entrelazados continuará con referencia a conceptos analizados en esta sección.

1. Decodificación de capa de macrobloque de fotogramas P entrelazados

En un fotograma P entrelazado, cada macrobloque puede compensarse en movimiento en modo de fotograma usando uno o cuatro vectores de movimiento o en modo de campo usando dos o cuatro vectores de movimiento. Un macrobloque que está inter-codificado no contiene ningún intra bloque. Además, el residuo después de la compensación de movimiento puede codificarse en modo de transformación de fotograma o modo de transformación de campo. Más específicamente, los componentes de luminancia del residuo se reorganizan de acuerdo con los campos si se codifican en modo de transformación de campo y permanecen sin cambiar en modo de transformación de fotograma mientras que el componente de crominancia permanece igual. Un macrobloque puede codificarse también como intra.

La compensación de movimiento puede restringirse para no incluir cuatro (tanto de campo/fotograma) vectores de movimiento, y esto se señala a través de 4MVSWITCH. El tipo de compensación de movimiento y codificación residual se indica conjuntamente para cada macrobloque a través de MBMODE y SKIPMB. MBMODE emplea un conjunto de tablas diferente de acuerdo con 4MVSWITCH.

Los macrobloques en fotogramas P entrelazados se clasifican en cinco tipos: 1 MV, MV de 2 campos, MV de 4 fotogramas, MV de 4 campos e intra. Los primeros cuatro tipos de macrobloques están inter-codificados mientras que el último tipo indica que el macrobloque está intra-codificado. El tipo de macrobloque se señala mediante el elemento de sintaxis MBMODE en la capa de macrobloque junto con el bit de salto. MBMODE codifica conjuntamente tipos de macrobloque junto con diversas piezas de información con respecto al macrobloque para diferentes tipos de macrobloque.

Señalización de macrobloque saltado

El campo SKIPMB indica la condición de salto para un macrobloque. Si el campo SKIPMB es 1, entonces el macrobloque actual se dice que se salta y no se envía otra información después del campo SKIPMB. La condición de salto implica que el macrobloque actual es de 1 MV con vector de movimiento diferencial cero (es decir el macrobloque está compensado en movimiento usando su predictor de movimiento de 1 MV) y no hay bloques codificados (CBP = 0).

Por otra parte, si el campo SKIPMB no es 1, el campo MBMODE se decodifica para indicar el tipo de macrobloque y otra información con respecto al macrobloque actual, tal como información descrita en la siguiente sección.

Señalización de modo de macrobloque

Hay quince posibles eventos que se indican por MBMODE; MBMODE especifica conjuntamente el tipo de macrobloque (1MV, MV de 4 fotogramas, MV de 2 campos, MV de 4 campos o intra), tipos de transformación para

macrobloque inter-codificado (es decir bloques de campo o fotograma o no codificados), y si hay un vector de movimiento diferencial para el macrobloque de 1 MV.

Indicando <MVP> un evento binario que señala si hay o no vector de movimiento diferencial de 1 MV distinto de cero. Indicando <transformación de campo/fotograma> un evento ternario que señala si el residuo del macrobloque está codificado por transformación de fotograma, codificado por transformación de campo o cero bloques codificados (es decir CBP = 0). MBMODE señala el siguiente conjunto de eventos conjuntamente:

```
MBMODE = { <1MV, MVP, transformación de campo/fotograma>, <MV de 2 campos,
transformación de campo/fotograma>, <MV de 4 fotogramas, transformación de
campo/fotograma>, <MV de 4 campos, transformación de campo/fotograma>, <INTRA>};
excluyendo el caso donde <1MV, MVP=0, CBP=0>, que se señala mediante la condición
de salto.
```

Para macrobloques inter-codificados, el elemento de sintaxis CBPCY no se decodifica cuando el evento de transformación de campo/fotograma en MBMODE indica no bloques codificados. Por otra parte, si el evento de transformación de campo/fotograma en MBMODE indica transformación de campo o fotograma, entonces se decodifica CBPCY. El evento codificado <transformación de campo/fotograma> se usa para establecer la bandera FIELDTX. Si este evento indica que el macrobloque está codificado por transformación de campo, FIELDTX se establece a uno. Si el evento indica que el macrobloque está codificado por transformación de fotograma, FIELDTX se establece a cero. Si el evento indica un bloque codificado a cero, FIELDTX se establece al mismo tipo que el vector de movimiento, es decir, FIELDTX se establece a 1 si es un FIELDMV, y se establece a 0 si es un MV de FOTOGRAMA.

Para unos macrobloques inter-codificados no de 1MV, se envía un campo adicional para indicar el evento de vectores de movimiento diferencial de cero. En el caso de macrobloques de MV de 2 campos, el campo 2MVBP se envía para indicar cuál de los dos vectores de movimiento contiene vectores de movimiento diferencial distintos de cero. De manera similar, el campo 4MVBP se envía para indicar cuál de los cuatro vectores de movimiento contiene vectores de movimiento diferencial distintos de cero.

Para macrobloques intra-codificados, los bloques de transformación de campo/fotograma y codificados a cero se codifican en campos separados.

2. Decodificación de vector de movimiento para fotogramas P entrelazados

Predictores de vector de movimiento para fotogramas P entrelazados

El procedimiento de calcular el predictor o predictores de vector de movimiento para el macrobloque actual consiste en dos etapas. En primer lugar, se recopilan tres vectores de movimiento candidatos para el macrobloque actual desde sus macrobloques vecinos. En segundo lugar, el predictor o predictores de vector de movimiento para el macrobloque actual se calculan desde el conjunto de vectores de movimiento candidatos. Las Figuras 40A-40B muestran macrobloques vecinos desde los cuales se recuperan los vectores de movimiento candidatos. El orden de la recopilación de vectores de movimiento candidatos es importante. En esta implementación combinada, el orden de la recopilación siempre se inicia en A, continúa a B y finaliza en C. Obsérvese que un candidato de predictor se considera que es no existente si el correspondiente bloque está fuera del límite de fotograma o si el correspondiente bloque es parte de un corte diferente. Por lo tanto, la predicción de vector de movimiento no se realiza a través de límites de corte.

Las siguientes secciones describen cómo se recopilan los vectores de movimiento candidatos para diferentes tipos de macrobloques y cómo se calculan los predictores de vector de movimiento.

Vectores de movimiento candidatos de 1MV

En esta implementación combinada, el pseudocódigo 6400 en la Figura 64 se usa para recopilar los hasta tres vectores de movimiento candidatos para el vector de movimiento:

Vectores de movimiento candidatos de MV de 4 fotogramas

Para macrobloques de MV de 4 tramas, para cada uno de los cuatro vectores de movimiento de bloque de fotograma en el macrobloque actual, se recopilan los vectores de movimiento candidatos desde los bloques vecinos. En esta implementación combinada, el pseudocódigo 6500 en la Figura 65 se usa para recopilar los hasta tres vectores de movimiento candidatos para el vector de movimiento de bloque de fotograma superior izquierdo. El pseudocódigo 6600 en la Figura 66 se usa para recopilar los hasta tres vectores de movimiento candidatos para el vector de movimiento de bloque de fotograma superior derecho. El pseudocódigo 6700 en la Figura 67 se usa para recopilar los hasta tres vectores de movimiento candidatos para el vector de movimiento de bloque de fotograma inferior izquierdo. El pseudocódigo 6800 en la Figura 68 se usa para recopilar los hasta tres vectores de movimiento candidatos para el vector de movimiento de bloque de fotograma inferior derecho.

Derivación de vectores de movimiento candidatos de MV de 2 campos

Para macrobloques de MV de 2 campos, para cada uno de los dos vectores de movimiento de campo en el macrobloque actual, se recopilan los vectores de movimiento candidatos desde los bloques vecinos. El pseudocódigo 6900 en la Figura 69 se usa para recopilar los hasta tres vectores de movimiento candidatos para el vector de movimiento de campo superior. El pseudocódigo 7000 en la Figura 70 se usa para recopilar los hasta tres vectores de movimiento candidatos para el vector de movimiento de campo inferior.

Derivación de vectores de movimiento candidatos de MV de 4 campos

Para macrobloques de MV de 4 campos, para cada uno de los cuatro bloques de campo en el macrobloque actual, se recopilan los vectores de movimiento candidatos desde los bloques vecinos. El pseudocódigo 7100 en la Figura 71 se usa para recopilar los hasta tres vectores de movimiento candidatos para el vector de movimiento de bloque de campo superior izquierdo. El pseudocódigo 7200 en la Figura 72 se usa para recopilar los hasta tres vectores de movimiento candidatos para el bloque vector de movimiento de campo superior derecho. El pseudocódigo 7300 en la Figura 73 se usa para recopilar los hasta tres vectores de movimiento candidatos para el vector de movimiento de bloque de campo inferior izquierdo. El pseudocódigo 7400 en la Figura 74 se usa para recopilar los hasta tres vectores de movimiento candidatos para el vector de movimiento de bloque de campo inferior derecho.

Vectores de movimiento de campo promedio

Dados dos vectores de movimiento de campo (MVX_1, MVY_1) y (MVX_2, MVY_2), la operación de promedio usada para formar un vector de movimiento de candidato (MVX_A, MVY_A) es:

$$MVX_A = (MVX_1 + MVX_2 + 1) \gg 1;$$

$$MVY_A = (MVY_1 + MVY_2 + 1) \gg 1;$$

Calcular predictores de MV de fotograma desde vectores de movimiento candidatos

Esta sección describe cómo se calculan predictores de vector de movimiento para vectores de movimiento de fotograma dado un conjunto de vectores de movimiento candidatos. En esta implementación combinada, la operación es la misma para calcular el predictor para 1 MV o para cada uno de los cuatro vectores de movimiento de bloque de fotograma en macrobloques de MV de 4 fotogramas.

El pseudocódigo 7500 en la Figura 75 describe cómo el predictor de vector de movimiento (PMV_x, PMV_y) se calcula para vectores de movimiento de fotograma. En el pseudocódigo 7500, TotalValidMV indica el número total de vectores de movimiento en el conjunto de vectores de movimiento candidatos (TotalValidMV = 0, 1, 2 o 3), la matriz ValidMV indica el vector de movimiento en el conjunto de vectores de movimiento candidatos.

Calcular predictores de MV de campo desde vectores de movimiento candidatos

Esta sección describe cómo se calculan predictores de vector de movimiento para vectores de movimiento de campo dado el conjunto de vectores de movimiento candidatos. La operación es la misma para calcular el predictor para cada uno de los dos vectores de movimiento de campo en macrobloques de MV de 2 campos o para cada uno de los cuatro vectores de movimiento de bloque de campo en macrobloques de MV de 4 campos.

En primer lugar, los vectores de movimiento candidatos se separan en dos conjuntos, donde un conjunto contiene únicamente vectores de movimiento candidatos que apuntan al mismo campo que el campo actual y el otro conjunto contiene vectores de movimiento candidatos que apuntan al campo opuesto. Suponiendo que los vectores de movimiento candidatos se representan en unidades de cuarto de píxel, el codificador o decodificador puede comprobar si un vector de movimiento de candidato apunta al mismo campo mediante la siguiente comprobación en su componente y:

```
if (ValidMVy & 4) {
    ValidMV apunta al campo opuesto.
}else{
    ValidMV apunta al mismo campo.
}
```

El pseudocódigo 7600 en la Figura 76 describe cómo el predictor de vector de movimiento (PMV_x, PMV_y) se calcula para vectores de movimiento de campo. En el pseudocódigo 7600, SameFieldMV y OppFieldMV indican dos conjuntos de vectores de movimiento candidatos y NumSameFieldMV y NumOppFieldMV indican el número de vectores de movimiento candidatos que pertenecen a cada conjunto. El orden de vectores de movimiento candidatos en cada conjunto se inicia con el candidato A si existe, seguido por el candidato B si existe, y a continuación el candidato C si existe. Por ejemplo, si el conjunto de vectores de movimiento candidatos de SameFieldMV contiene únicamente el candidato B y el candidato C, entonces SameFieldMV[0] es el candidato B.

Decodificar vectores de movimiento diferenciales

Los elementos de sintaxis de MVDATA contienen información de vector de movimiento diferencial para el macrobloque. Dependiendo del tipo de compensación de movimiento y patrón de bloque de vector de movimiento señalado en cada macrobloque, puede haber hasta cuatro elementos de sintaxis MVDATA por macrobloque. Más específicamente,

- 5 • Para macrobloques de 1 MV, puede haber cualquiera del elemento de sintaxis MVDATA 0 o 1 presente dependiendo del campo MVP en MBMODE.
- Para macrobloques de MV de 2 campos, puede haber cualquiera del elemento o elementos de sintaxis MVDATA 0, 1, o 2 presentes dependiendo de 2MVBP.
- 10 • Para macrobloques de MV de 4 fotogramas / campos, puede haber cualquiera del elementos o elementos de sintaxis MVDATA 0, 1, 2, 3 o 4 presentes dependiendo de 4MVBP.

En esta implementación combinada, el vector de movimiento diferencial se decodifica de la misma manera que un vector de movimiento diferencial campo de referencia para campos P entrelazados. (El pseudocódigo 7700 en la Figura 77A ilustra cómo se decodifica el vector de movimiento diferencial para un campo de una referencia. El pseudocódigo 7710 en la Figura 77B ilustra cómo se decodifica el vector de movimiento diferencial para un campo de una referencia en una implementación combinada alternativa. El pseudocódigo 7710 decodifica vectores de movimiento diferenciales de una manera diferente. Por ejemplo, el pseudocódigo 7710 omite el manejo de intervalos de vector de movimiento diferenciales extendidos).

Reconstruir vectores de movimiento

Dado el vector de movimiento diferencial dmv, el vector de movimiento de luminancia se reconstruye añadiendo el diferencial al predictor como se describe en la sección XV.B.3 anterior. Dado un vector de movimiento de fotograma o campo de luminancia, se deriva un correspondiente vector de movimiento de fotograma o campo de crominancia para compensar una porción de o el bloque de Cb/Cr completo. El pseudocódigo 7800 en la Figura 78 describe cómo se deriva un vector de movimiento de crominancia CMV desde un vector de movimiento de luminancia LMV en fotogramas P entrelazados.

D. Decodificar fotogramas B entrelazados

Esta sección describe el procedimiento para decodificar fotogramas B entrelazados en la implementación combinada, con referencia a conceptos analizados en la sección anterior.

1. Decodificación de capa de macrobloque de fotogramas B entrelazados

En el nivel de macrobloque, la sintaxis de fotograma B entrelazado es similar a la sintaxis de fotograma P entrelazado anteriormente descrita. Los macrobloques en fotogramas B entrelazados se clasifican en 3 tipos. 1 MV, MV de 2 campos e intra. Los modos de MV de 4 fotogramas y MV de 4 campos no están permitidos para fotogramas B entrelazados en esta implementación combinada. Los tres modos se codifican conjuntamente con el elemento de sintaxis MBMODE, como en fotogramas P entrelazados. Cada macrobloque se predice también como hacia delante, hacia atrás, directo o interpolado (usando los elementos de sintaxis DIRECTMB y BMVTYPE). Si un macrobloque de 1 MV es cualquiera de hacia delante o hacia atrás, usa un único vector de movimiento. Si es 1 MV pero directo o interpolado, usa dos vectores de movimiento. Si es de tipo MV de 2 campos y cualquiera de previsto hacia delante o hacia atrás, entonces usa dos vectores de movimiento. Si es de tipo MV de 2 campos y directo o interpolado, entonces usa cuatro vectores de movimiento.

Las siguientes secciones describen características de diferentes tipos de macrobloque inter-codificado en fotogramas B entrelazados.

Macrobloques de 1MV en fotogramas B entrelazados

En un macrobloque de 1 MV en un fotograma B entrelazado, el desplazamiento de los bloques de luminancia se representa mediante un único vector de movimiento cuando el tipo de predicción es hacia delante o hacia atrás, y mediante dos vectores de movimiento cuando el tipo es directo o interpolado. Se derivan los correspondientes vectores de movimiento de crominancia en cada caso. En el caso de predicción interpolada y directa, los píxeles de movimiento compensado desde imágenes de referencia hacia delante y hacia atrás se promedian para formar la predicción final.

Macrobloques de MV de 2 campos en fotogramas B entrelazados

En macrobloques de MV de 2 campos en fotogramas B entrelazados, el desplazamiento de cada campo de los bloques de luminancia se describe mediante un vector de movimiento diferente, como se muestra en la Figura 37. Adicionalmente, el tipo de predicción se permite conmutar desde hacia delante a hacia atrás o viceversa al pasar desde el campo superior al inferior, permitiendo por lo tanto que el campo superior se compense en movimiento desde una imagen de referencia y que el campo inferior se compense en movimiento desde la otra imagen de referencia, como se analiza en la sección VII, anterior.

Interpretación de 2MVBP, 4MVBP y orden de vectores de movimiento en fotogramas B entrelazados

En un macrobloque de 1MV, un codificador usa el elemento de sintaxis 2MVBP en modo interpolado para indicar cuál de los dos vectores de movimiento está presente. El bit 1 corresponde al vector de movimiento hacia delante y el bit 0 corresponde al vector de movimiento hacia atrás.

- 5 En un macrobloque de MV de 2 campos, un codificador usa el elemento de sintaxis 2MVBP en modo hacia delante y hacia atrás para indicar cuál de los dos vectores de movimiento de campo está presente. El bit 1 corresponde al vector de movimiento de campo superior y el bit 0 corresponde al vector de movimiento de campo inferior. El codificador usa la misma señalización superior/inferior cuando se usa el elemento de sintaxis MVSW para conmutar desde predicción hacia delante para el campo superior a predicción hacia atrás para el campo inferior, o viceversa.
- 10 El codificador usa el elemento de sintaxis 4MVBP en modo interpolado para indicar cuál de los cuatro vectores de movimiento está presente. El bit 3 corresponde al vector de movimiento hacia delante de campo superior, el bit 2 corresponde al vector de movimiento hacia atrás de campo superior, el bit 1 corresponde al vector de movimiento hacia delante de campo inferior, y el bit 0 corresponde al vector de movimiento hacia atrás de campo inferior.

- 15 Los bits de 2MVBP y 4MVBP establecidos a '1' significan que el vector de movimiento diferencial correspondiente está presente, mientras que los bits establecidos a '0' significan que el vector de movimiento correspondiente es igual al vector de movimiento previsto, es decir que no hay vector de movimiento diferencial correspondiente presente. Los vectores de movimiento decodificados reales se envían en el mismo orden que los bits en 2MVBP o 4MVBP. Por ejemplo, en un macrobloque de MV de 2 campos que usa modo interpolado, el primer vector de movimiento a recibirse mediante el decodificador es el vector de movimiento hacia delante de campo superior, y el
- 20 último (es decir, cuarto) vector de movimiento a recibirse es el vector de movimiento hacia atrás de campo inferior.

Señalización de macrobloque saltado

- Los macrobloques saltados se señalizan de la misma manera que con fotogramas P. Sin embargo, los macrobloques saltados en fotogramas B entrelazados se restringen a tipo de fotograma de 1 MV, es decir no están permitidos tipos de campo. El vector de movimiento se codifica con un vector de movimiento de diferencial cero (es
- 25 decir el macrobloque está compensado en movimiento usando su predictor de movimiento de 1 MV) y no hay bloques codificados (CBP = 0). Si un macrobloque se salta, el codificador envía únicamente la información de BMVTYPE para ese macrobloque, de modo que los vectores de movimiento pueden predecirse correctamente como hacia delante, hacia atrás, directo o interpolado.

Señalización de modo de macrobloque

- 30 La señalización de modo de macrobloque se realiza de la misma manera que los fotogramas P entrelazados, como se ha descrito en la sección XV.C. anterior.

Decodificación de tipo de predicción (BMVTYPE y MVSW)

- El tipo de predicción se decodifica para fotogramas B entrelazados de acuerdo con las siguientes reglas. Si el plano de bits de nivel de imagen DIRECTMB indica que un macrobloque es de tipo directo, entonces el tipo de predicción
- 35 para ese macrobloque se establece a directo. Si la decisión directo/no directo se codifica en modo sin procesar, el codificador usa un bit adicional en el nivel de macrobloque, DIRECTBBIT, para indicar si el tipo de predicción es directo o no.

- Si el tipo de predicción es no directo, entonces el decodificador decodifica el elemento de sintaxis BMVTYPE. Si el modo de macrobloque es "codificado con campo de 2 MV" y si el BMVTYPE es cualquiera de hacia delante o hacia
- 40 atrás, el decodificador también decodifica el bit de MVSW para decidir si cambiará o no el tipo de predicción (es decir, dar la vuelta desde hacia delante a hacia atrás o viceversa) al pasar desde el campo superior al inferior para ese macrobloque.

Decodificar vectores de movimiento de modo directo

- Para decodificar vectores de movimiento de modo directo, un decodificador almacena en primer lugar en memoria intermedia vectores de movimiento desde el fotograma de ancla previamente decodificado. Específicamente, para
- 45 un fotograma P futuro previamente decodificado, el decodificador almacena en memoria intermedia la mitad del número máximo posible de vectores de movimiento de luminancia decodificados desde el fotograma P futuro (es decir, (2 * Número de MB) vectores de movimiento). El procedimiento de selección de estos vectores de movimiento para almacenar en memoria intermedia desde el fotograma de ancla se ha descrito en la sección XIII anterior.

- 50 Con los vectores de movimiento anteriormente obtenidos, el decodificador aplica la lógica de cambio de escala en Scale_Direct_MV, como se muestra en el pseudocódigo 1900 en la Figura 19, para obtener vectores de movimiento que apuntan hacia delante y hacia atrás, sin retirar el vector de movimiento.

En esta implementación combinada, no hay cálculo de vectores de movimiento de modo directo para macrobloques que no están usando predicción de modo directo, tal como macrobloques previstos hacia delante y hacia atrás. En

su lugar, los vectores de movimiento para macrobloques no directos se predicen basándose en las memorias intermedias de vector de movimiento hacia delante o hacia atrás.

2. Decodificar vector de movimiento para fotogramas B entrelazados

Predictores de vector de movimiento para fotogramas B entrelazados

- 5 Como con fotogramas P entrelazados, el procedimiento de calcular los predictores de vector de movimiento para un macrobloque actual en un fotograma B entrelazado incluye recoger vectores de movimiento candidatos para el macrobloque actual desde sus macrobloques vecinos, y calcular los predictores de vector de movimiento para el macrobloque actual desde el conjunto de vectores de movimiento candidatos. Las Figuras 40A-40B muestran macrobloques vecinos desde los que se recogen los vectores de movimiento candidatos. En esta implementación combinada, los predictores de vector de movimiento para fotogramas B entrelazados se seleccionan desde un conjunto de candidatos de acuerdo con las reglas descritas en la sección XV.C. anterior para fotogramas P entrelazados.

- 15 Se usan contextos de predicción separados para vectores de movimiento de modo hacia delante y hacia atrás. El decodificador usa el contexto de predicción hacia delante para predecir vectores de movimiento hacia delante y el contexto de predicción hacia atrás para predecir vectores de movimiento hacia atrás.

Rellenar los contextos de predicción hacia delante y hacia atrás en fotogramas B entrelazados

- 20 El decodificador almacena en memoria intermedia los vectores de movimiento hacia delante y hacia atrás de manera separada y los usa para predecir vectores de movimiento hacia delante y hacia atrás, respectivamente. Para macrobloques interpolados, el decodificador usa la memoria intermedia de predicción hacia delante para predecir el vector de movimiento hacia delante (el primer elemento MVDATA decodificado) y la memoria intermedia hacia atrás para predecir el vector de movimiento hacia atrás (el segundo elemento MVDATA decodificado). Cuando el macrobloque es directo o interpolado, el decodificador almacena en memoria intermedia el componente de MV hacia delante en la memoria intermedia hacia delante y el componente de MV hacia atrás en la memoria intermedia hacia atrás. La lógica de predicción real en cada caso (por ejemplo, para macrobloques de 1 MV, macrobloques de MV de 2 campos, etc.) para seleccionar un predictor de vector de movimiento de entre un conjunto de candidatos se ha descrito anteriormente en la sección XV.C.

- 25 El esquema para rellenar las memorias intermedias de vector de movimiento hacia delante y hacia atrás y predecir vectores de movimiento desde los vectores de movimiento en estas memorias intermedias se ha descrito anteriormente en la sección X.C.

Decodificar vectores de movimiento diferenciales en fotogramas B entrelazados

Los vectores de movimiento diferenciales en fotogramas B entrelazados se decodifican de acuerdo con el pseudocódigo 7700 y 7710 en las Figuras 77A y 77B como se ha analizado anteriormente en la sección XV.C.2.

Reconstruir vectores de movimiento en fotogramas B entrelazados

- 35 Los vectores de movimiento que se reconstruyen en fotogramas B entrelazados que se decodifican de acuerdo con el pseudocódigo 7800 en la Figura 78 y se han analizado anteriormente en las secciones XV.B.3 y XV.C.2.

E. Codificación de plano de bits

- 40 La información binaria específica de macrobloque tal como (1) decisiones hacia delante/no hacia delante para los macrobloques de campos B entrelazados (es decir, la bandera FORWARDMB), y (2) decisiones de directo/no directo para los macrobloques de fotogramas B entrelazados (es decir, la bandera DIRECTMB), puede codificarse en un símbolo binario por macrobloque. Por ejemplo, si un macrobloque de un campo B entrelazado está o no compensado en movimiento en modo hacia delante (frente a otro modo tal como hacia atrás, directo o interpolado) puede señalizarse con un bit. En estos casos, el estado de todos los macrobloques en un campo o fotograma puede codificarse como un plano de bits y transmitirse en el encabezamiento de campo o de fotograma. Una excepción para esta regla es si el modo de codificación de plano de bits se establece a Modo Sin Procesar, caso en el que el estado para cada macrobloque se codifica como un bit por símbolo y se transmite junto con otros elementos de sintaxis de nivel de macrobloque en el nivel de macrobloque.

- 45 La codificación de plano de bits de nivel de campo/fotograma se usa para codificar matrices binarias bidimensionales. El tamaño de cada matriz es $rowMB \times colMB$, donde $rowMB$ y $colMB$ son el número de filas y columnas de macrobloque, respectivamente, en el campo o fotograma en cuestión. En el flujo de bits, cada matriz se codifica como un conjunto de bits consecutivos. Se usa uno de siete modos para codificar cada matriz. Los siete modos son:

1. modo sin procesar - la información se codifica como un bit por símbolo y se transmite como parte de sintaxis de nivel de MB;
2. modo normal-2 - dos símbolos codificados conjuntamente;

3. modo diferencial-2 - codificación diferencial del plano de bits, seguida por codificar dos símbolos residuales conjuntamente;
4. modo normal-6 - seis símbolos codificados conjuntamente;
5. modo diferencial-6 - codificación diferencial del plano de bits, seguida por codificar seis símbolos residuales conjuntamente;
6. modo de salto de fila - salto de un bit para señalar filas con bits no posicionados; y
7. modo de salto de columna - salto de un bit para señalar columnas con bits no posicionados.

Los elementos de sintaxis para un plano de bits en el nivel de campo o de fotograma están en la siguiente secuencia: INVERT, IMODE y DATABITS.

Bandera de inversión (INVERT)

El elemento de sintaxis INVERT es un valor de 1 bit, que si está posicionado indica que el plano de bits tiene más bits posicionados que cero bits. Dependiendo de INVERT y el modo, el decodificador deberá invertir el plano de bits interpretado para recrear el original. Obsérvese que el valor de este bit deberá ignorarse cuando se usa el modo sin procesar. La descripción de cómo se usa el valor de INVERT al decodificar el plano de bits se proporciona a continuación.

Modo de codificación (IMODE)

El elemento de sintaxis IMODE es un valor de longitud variable que indica el modo de codificación usado para codificar el plano de bits. La tabla 11 muestra la tabla de códigos usada para codificar el elemento de sintaxis IMODE. La descripción de cómo se usa el valor IMODE al decodificar el plano de bits se proporciona a continuación.

Tabla 11: tabla de códigos de VLC de IMODE

VLC IMODE	Modo de codificación
10	Norm-2
11	Norm-6
010	Salto de fila
011	Salto de columna
001	Dif-2
0001	Dif-6
0000	Sin procesar

Bits de codificación de plano de bits (DATABITS)

El elemento de sintaxis DATABITS es el elemento de sintaxis con tamaño variable que codifica el flujo de símbolos para el plano de bits. El procedimiento usado para codificar el plano de bits se determina mediante el valor de IMODE. Los siete modos de codificación se describen en las siguientes secciones.

Modo sin procesar

En este modo, el plano de bits se codifica como un bit por símbolo explorado en el orden de exploración de trama de macrobloques, y se envía como parte de la capa de macrobloque. Como alternativa, la información se codifica en modo sin procesar en el nivel de campo o fotograma y DATABITS es rowMB x colMB bits de longitud.

Modo normal-2

Si rowMB x colMB son impares, el primer símbolo se codifica sin procesar. Los símbolos posteriores se codifican por pares, en orden de exploración natural. La tabla de VLC binaria en la tabla 12 se usa para codificar pares de símbolos.

Tabla 12: tabla de códigos Norm-2/Dif-2

Símbolo 2n	Símbolo 2n + 1	Palabra de código
0	0	0
1	0	100
0	1	101
1	1	11

Modo dif-2

El procedimiento normal-2 se usa para producir el plano de bits como se ha descrito anteriormente, y a continuación se aplica la operación dif¹ al plano de bits como se describe a continuación.

Modo normal-6

- 5 En los modos normal-6 y dif-6, el plano de bits se codifica en grupos de seis píxeles. Estos píxeles se agrupan en piezas de 2x3 o 3x2. El plano de bits se apila de manera máxima usando un conjunto de reglas, y los restantes píxeles se codifican usando una variante de modos de salto de fila y salto de columna. Se usan piezas "verticales" de 2x3 si y únicamente si rowMB es un múltiplo de 3 y colMB no. De otra manera, se usan piezas "horizontales" de 3x2. La Figura 79A muestra un ejemplo simplificado de piezas "verticales" de 2x3. Las Figuras 79B y 79C muestran ejemplos simplificados de piezas "horizontales" de 3x2 para las que los rectángulos oscuros alargados son de 1 píxel de ancho y se codifican usando codificación de salto de fila y salto de columna. Para un plano apilado como se muestra en la Figura 79C, con piezas lineales a lo largo de los bordes superior e izquierdo de la imagen, el orden de codificación de las piezas sigue el siguiente patrón. Las piezas de 6 elementos se codifican en primer lugar, seguido por las piezas lineales codificadas por salto de columna y salto de fila. Si el tamaño de la matriz es un múltiplo de 2x3 o de 3x2, las piezas lineales anteriores no existen y el plano de bits está perfectamente apilado.

- 10 Las piezas rectangulares de 6 elementos que usan un código Huffman incompleto, es decir, un código Huffman que no usa todos los nodos de extremo para codificación. Sea N el número de bits posicionados en la pieza, es decir $0 \leq N \leq 6$. Para $N < 3$, se usa un VLC para codificar la pieza. Para $N = 3$, se sigue un escape de longitud fija por un código de longitud de 5 bits fijos, y para $N > 3$, se sigue un escape de longitud fija por el código del complemento de la pieza. La pieza rectangular contiene 6 bits de información. Sea k el código asociado con la pieza, donde $k = b_i 2^i$, b_i es el valor binario del i-ésimo bit en orden de exploración natural dentro de la pieza. Por lo tanto $0 \leq k < 64$. Una combinación de VLC y códigos de escape códigos de longitud fija se usa para señalar k.

Modo dif-6

- 25 El procedimiento normal-6 se usa para producir el plano de bits como se ha descrito anteriormente, y a continuación se aplica la operación dif¹ al plano de bits como se describe a continuación.

Modo salto de fila

- 30 En el modo de codificación de salto de fila, todas las filas cero se saltan con tara de un bit. La sintaxis es como sigue: para cada fila, un único bit de ROWSKIP indica si la fila está saltada; si la fila está saltada, el bit ROWSKIP para la siguiente fila es siguiente; de otra manera (la fila no está saltada), los bits ROWBITS (un bit para cada macrobloque en la fila) son los siguientes. Por lo tanto, si la fila entera es cero, se envía un bit cero como el símbolo ROWSKIP, y se salta ROWBITS. Si hay un bit posicionado en la fila, ROWSKIP se establece a 1, y la fila completa se envía sin procesar (ROWBITS). Las filas se exploran desde la parte superior a la inferior del campo o fotograma.

Modo de salto de columna

- 35 Salto de columna es la traspuesta de salto de fila. Las columnas se exploran desde la izquierda a la derecha del campo o fotograma.

Dif¹: decodificación diferencial inversa

- 40 Si se usa cualquiera del modo diferencial (Dif-2 o Dif-6), un plano de bits de "bits diferenciales" se decodifica en primer lugar usando los correspondientes modos normales (Norm-2 o Norm-6 respectivamente). Los bits diferenciales se usan para regenerar el plano de bits original. El procedimiento de regeneración es un DPCM 2-D en alfabeto binario. Para regenerar el bit en la localización (i, j), el predictor $b_p(i, j)$ se genera como sigue (desde los bits $b(i, j)$ en las posiciones (i, j)):

$$b_p(i, j) = \begin{cases} A & i = j = 0, \text{ o } b(i, j - 1) \neq b(i - 1, j) \\ b(0, j - 1) & i == 0 \\ b(i - 1, j) & \text{de otra manera} \end{cases}$$

- 45 Para el modo de codificación diferencial, no se realiza el procedimiento de inversión a nivel de bits basándose en INVERT. Sin embargo, la bandera INVERT se usa en una capacidad diferente para indicar el valor del símbolo A para la derivación del predictor anteriormente mostrado. Más específicamente, A es igual a 0 si INVERT equivale a 0 y A es igual a 1 si INVERT equivale a 1. El valor real del plano de bits se obtiene realizando la operación xor al predictor con el valor de bit diferencial decodificado. En la ecuación anterior, $b(i, j)$ es el bit en la posición de orden i, j después de la decodificación final (es decir después de hacer Norm-2/Norm-6, seguido por la operación xor diferencial con su predictor).

- 50 Habiendo descrito e ilustrado los principios de nuestra invención con referencia a diversas realizaciones, se

5 reconocerá que las diversas realizaciones pueden modificarse en disposición y detalle sin alejarse de tales principios. Debería entenderse que los programas, procesos o procedimientos descritos en el presente documento no están relacionados o limitados a ningún tipo particular de entorno informático, a menos que se indique de otra manera. Pueden usarse diversos tipos de entornos informáticos de fin general o especializado o realizar operaciones de acuerdo con las enseñanzas descritas en el presente documento. Los elementos de las realizaciones mostradas en software pueden implementarse en hardware y viceversa.

En vista de las muchas posibles realizaciones a las que pueden aplicarse los principios de nuestra invención, reivindicamos nuestra invención todas tales realizaciones como que pueden entrar dentro del alcance de las siguientes realizaciones y equivalentes a las mismas.

10

REIVINDICACIONES

1. Un procedimiento de decodificación de información de vídeo comprimida con un decodificador de vídeo, comprendiendo el procedimiento:

recibir, en el decodificador de vídeo, información de vídeo comprimida en un flujo de bits; y

con el decodificador de vídeo, decodificar una imagen prevista bidireccionalmente que incluye diversos macrobloques usando la información de vídeo comprimida, en el que la decodificación incluye decodificar (5220) un conjunto de bits que representa información de decisión binaria que significa si los diversos macrobloques en la imagen prevista bidireccionalmente están compensados en movimiento usando predicción de modo hacia delante o uno de diversos tipos de predicción de modo no hacia delante, en el que, para cada uno de los diversos macrobloques de la imagen prevista bidireccionalmente, un único bit en el conjunto de bits está asociado con el macrobloque como parte de la información de decisión binaria del conjunto de bits, indicando el único bit si el tipo de predicción de movimiento compensado para el macrobloque es (a) la predicción de modo hacia delante o (b) uno de los diversos tipos de predicción de modo no hacia delante, en el que el conjunto de bits se señala como uno o más elementos de sintaxis de nivel de imagen, en el que una combinación de elementos de sintaxis de nivel de imagen y de nivel de macrobloque indica los tipos de predicción de movimiento compensado para los diversos macrobloques de la imagen prevista bidireccionalmente, y en el que:

para un primer macrobloque de los diversos macrobloques de la imagen prevista bidireccionalmente, el tipo de predicción de movimiento compensado es predicción de modo hacia delante, y el único bit para el macrobloque indica el tipo de predicción de movimiento compensado para el macrobloque; y

para un segundo macrobloque de los diversos macrobloques de la imagen prevista bidireccionalmente, el único bit para el macrobloque y un código de longitud variable de nivel de macrobloque para el macrobloque indican conjuntamente el tipo de predicción de movimiento compensado para el macrobloque.

2. El procedimiento de la reivindicación 1, en el que la imagen prevista bidireccionalmente es un campo B entrelazado.

3. El procedimiento de la reivindicación 1, en el que la decodificación de la imagen prevista bidireccionalmente con el decodificador de vídeo comprende adicionalmente seleccionar un modo de codificación de entre diversos modos de codificación disponibles para procesar el conjunto de bits.

4. El procedimiento de la reivindicación 1, en el que el conjunto de bits se señala en un encabezamiento de campo.

5. El procedimiento de la reivindicación 1, en el que la decodificación de la imagen prevista bidireccionalmente con el decodificador de vídeo comprende adicionalmente, para al menos uno de los diversos macrobloques de la imagen prevista bidireccionalmente:

determinar que la información de decisión binaria indica compensación de movimiento usando uno de los diversos tipos de predicción de modo no hacia delante para el macrobloque; y

decodificar el código de longitud variable de nivel de macrobloque para el macrobloque desde una tabla de códigos de longitud variable, en el que el código de longitud variable de nivel de macrobloque representa uno de los diversos tipos de predicción de modo no hacia delante para el macrobloque.

6. El procedimiento de la reivindicación 5, en el que la tabla de códigos de longitud variable refleja una preferencia de longitud de código para modo hacia atrás.

7. El procedimiento de la reivindicación 1, en el que los diversos tipos de predicción de modo no hacia delante incluyen modo hacia atrás, modo directo y modo interpolado.

8. Un dispositivo informático que implementa un decodificador de vídeo, comprendiendo el dispositivo informático un procesador, memoria y almacenamiento que almacena instrucciones ejecutables por ordenador para provocar que el procesador realice el procedimiento de una cualquiera de las reivindicaciones 1 a 7.

9. Un procedimiento de codificación de información de vídeo con un codificador de vídeo, comprendiendo el procedimiento:

con el codificador de vídeo, seleccionar un modo de codificación desde un grupo de diversos modos de codificación disponibles;

con el codificador de vídeo, codificar (5120) un conjunto de bits para un campo B entrelazado de acuerdo con el modo de codificación seleccionado, en el que el conjunto de bits representa información de decisión binaria que significa si diversos macrobloques en el campo B entrelazado están compensados en movimiento usando predicción de modo hacia delante o uno de los diversos tipos de predicción de modo no hacia delante, en el que, para cada uno de los diversos macrobloques en el campo B entrelazado, un único bit en el conjunto de bits está asociado con el macrobloque como parte de la información de decisión binaria del conjunto de bits, indicando el único bit si el tipo de predicción de movimiento compensado para el macrobloque es (a) la predicción de modo hacia delante o (b) uno de los diversos tipos de predicción de modo no hacia delante, en el que el conjunto de

bits se señala como uno o más elementos de sintaxis de nivel de campo y en el que una combinación de elementos de sintaxis de nivel de campo y de nivel de macrobloque indican los tipos de predicción de movimiento compensado para los diversos macrobloques del campo B entrelazado:

- 5 para un primer macrobloque de los diversos macrobloques del campo B entrelazado, el tipo de predicción de movimiento compensado es predicción de modo hacia delante, y el único bit para el macrobloque indica el tipo de predicción de movimiento compensado para el macrobloque; y
 para un segundo macrobloque de los diversos macrobloques del campo B entrelazado, el único bit para el macrobloque y un código de longitud variable de nivel de macrobloque para el macrobloque indican conjuntamente el tipo de predicción de movimiento compensado para el macrobloque; y con el codificador de vídeo, señalar el conjunto de bits codificados en un flujo de bits.
- 10 10. El procedimiento de la reivindicación 9, en el que al menos dos de los diversos modos de codificación disponibles implican reducción de tasa de bits asociada con la información de decisión binaria.
- 15 11. El procedimiento de la reivindicación 9, en el que los diversos modos de codificación disponibles incluyen un modo de codificación de predicción de salto de fila, un modo de codificación de predicción de salto de columna, uno o más modos de codificación de longitud variable de vector y uno o más modos de codificación diferencial, en el que en el modo de codificación de predicción de salto de fila, un bit se salta para señalar filas con bits no posicionados, y en el modo de codificación de predicción de salto de columna, un bit se salta para señalar columnas con bits no posicionados.
- 20 12. El procedimiento de la reivindicación 9, que comprende adicionalmente, para al menos uno de los diversos macrobloques del campo B entrelazado:

 con el codificador de vídeo, determinar que la información de decisión binaria indica compensación de movimiento usando uno de los diversos tipos de predicción de modo no hacia delante del macrobloque; y
 señalar, en el nivel de macrobloque para el macrobloque, el código de longitud variable de nivel de macrobloque para el macrobloque desde una tabla de códigos de longitud variable, en el que el código de longitud variable de nivel de macrobloque representa uno de los diversos tipos de predicción de modo no hacia delante para el macrobloque.
- 25 13. El procedimiento de la reivindicación 12, en el que la tabla de códigos de longitud variable refleja una preferencia de longitud de código para modo hacia atrás.
- 30 14. El procedimiento de una cualquiera de las reivindicaciones 5 y 12, en el que la tabla de códigos de longitud variable incluye un primer código para modo hacia atrás, un segundo código para modo directo y un tercer código para modo interpolado.
- 35 15. Un medio legible por ordenador que almacena instrucciones ejecutables por ordenador para provocar que un sistema informático programado mediante el mismo realice el procedimiento de cualquiera de las reivindicaciones 1 a 7 y 9 a 14.

Figura 1, técnica anterior

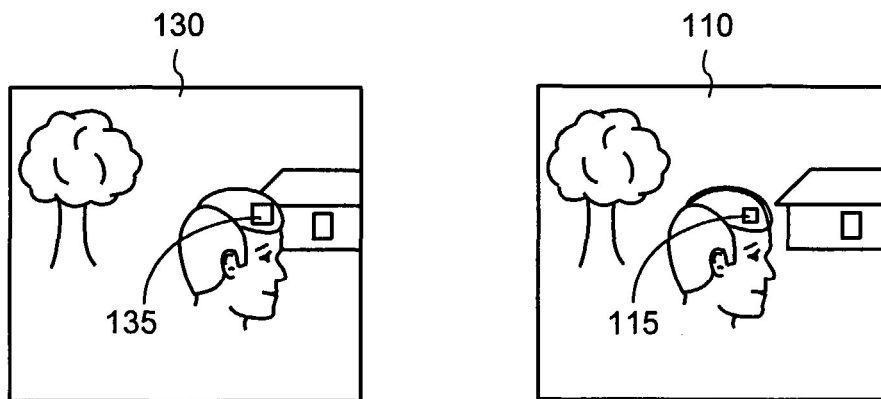


Figura 4, técnica anterior

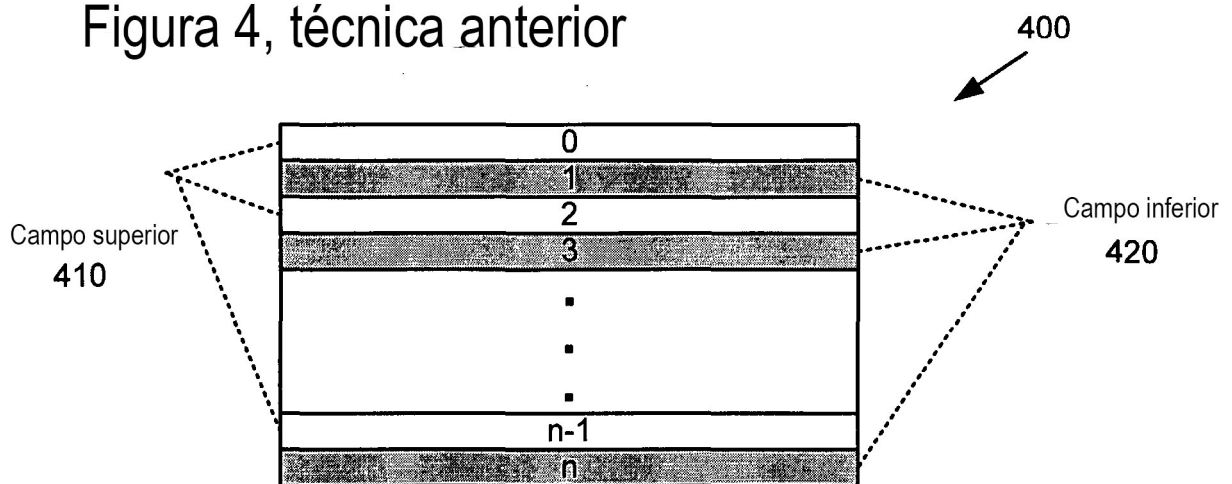


Figura 2, técnica anterior

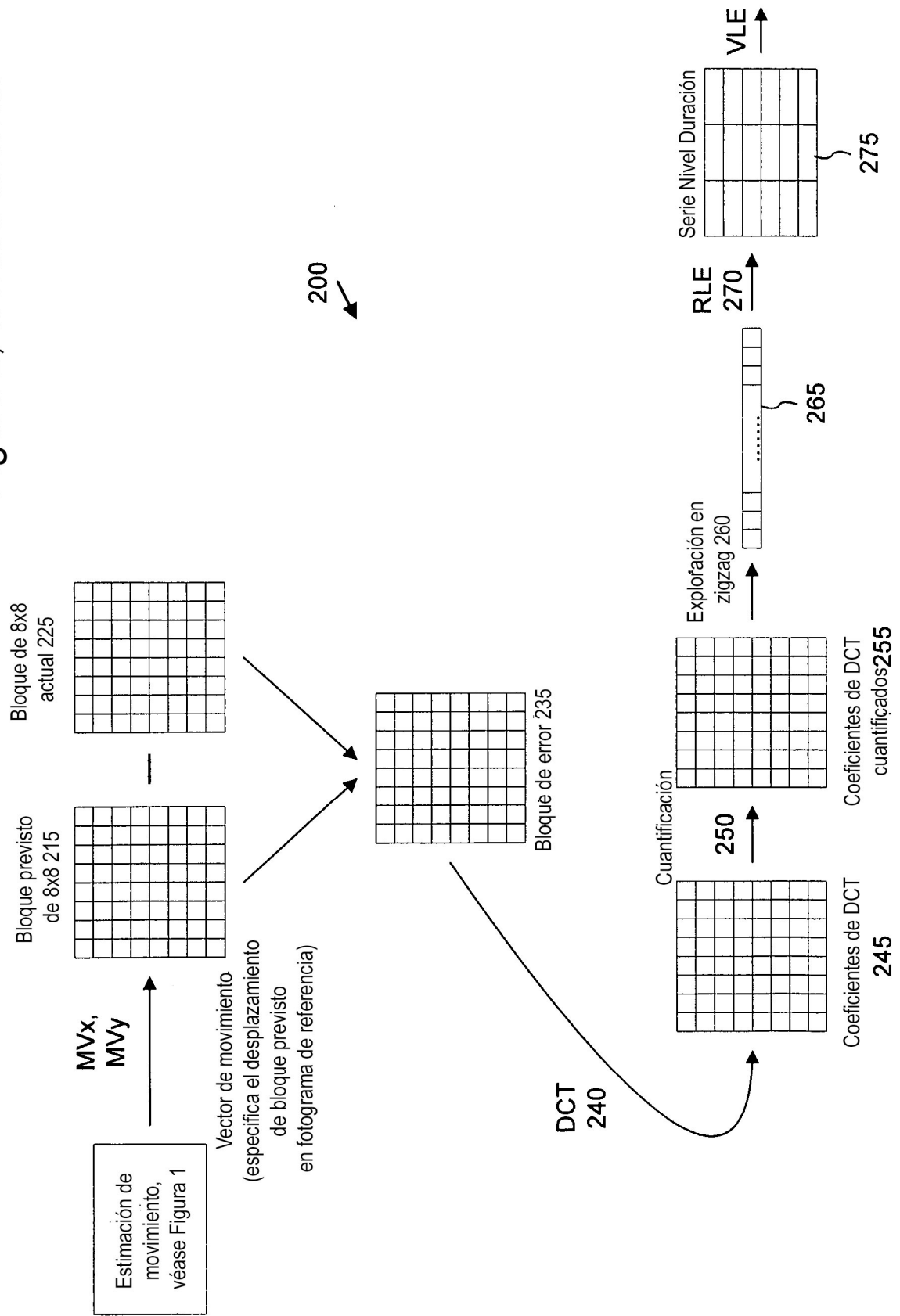


Figura 3, técnica anterior

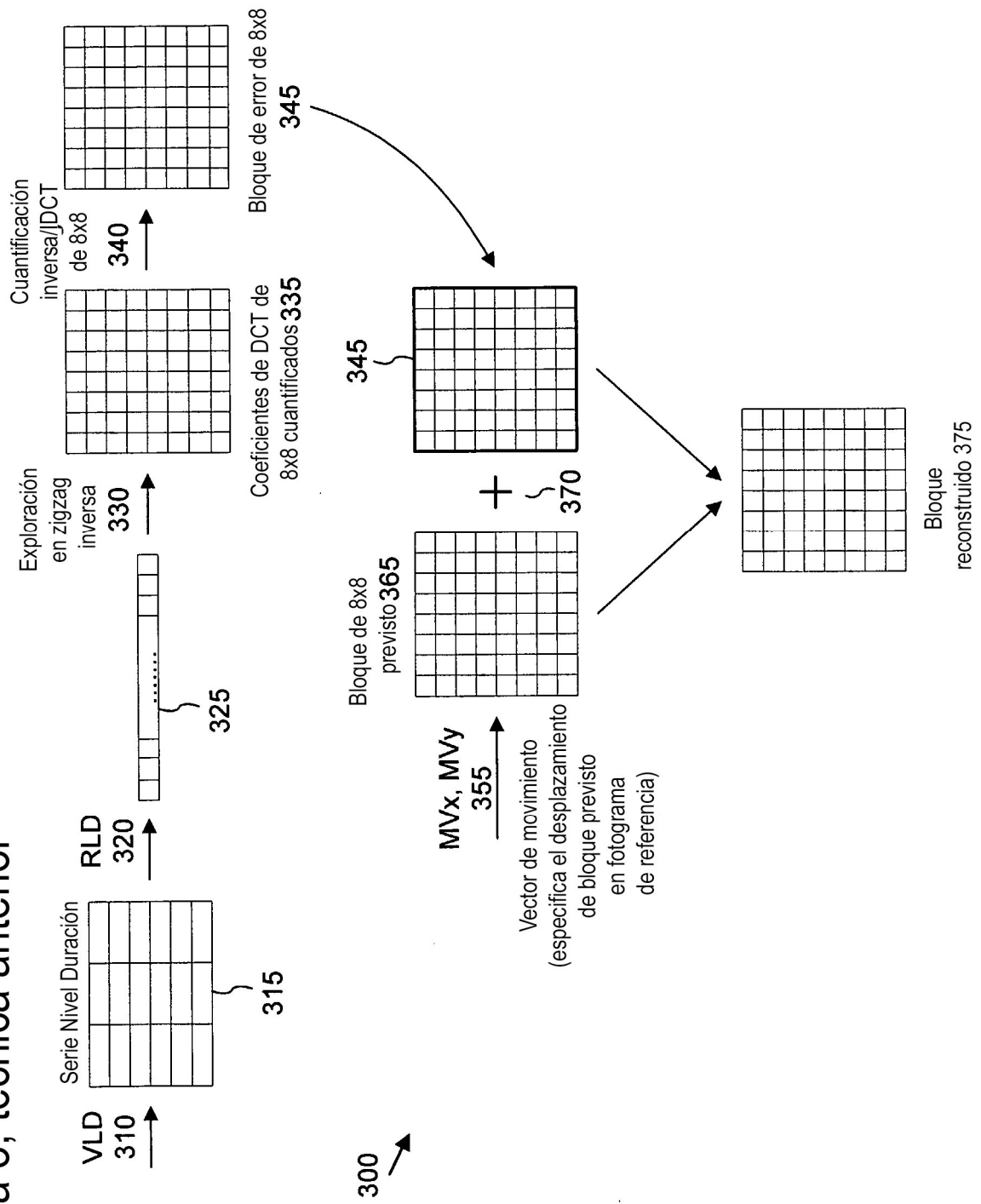


Figura 5A,
técnica anterior

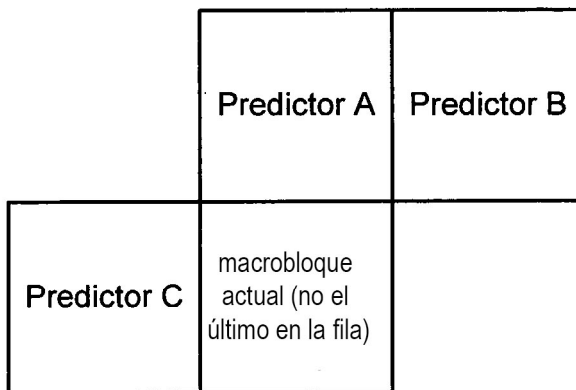


Figura 5B,
técnica anterior

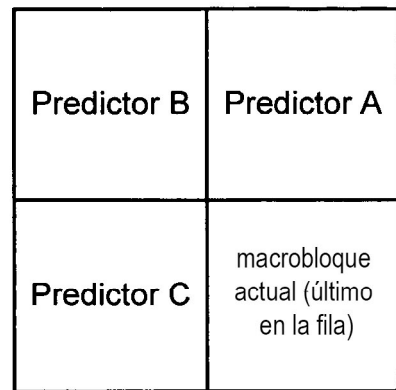


Figura 6A,
técnica anterior

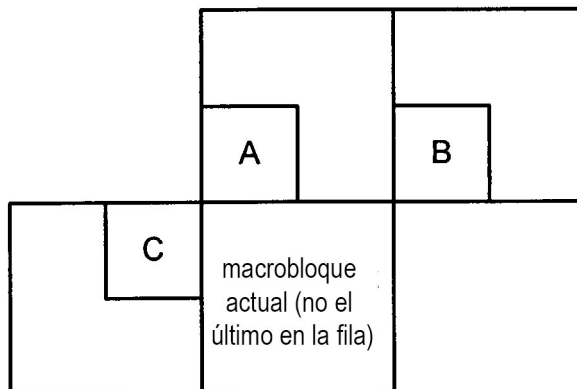


Figura 6B,
técnica anterior

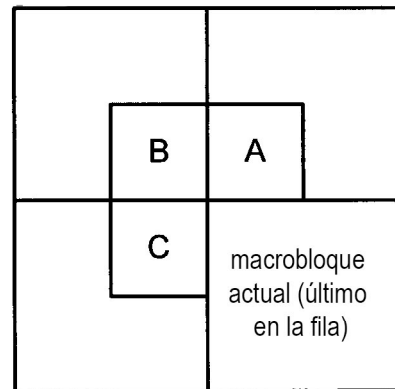


Figura 7A,
técnica anterior

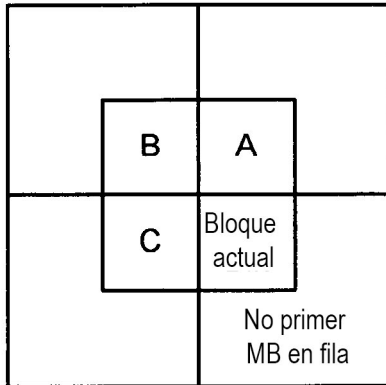


Figura 7B,
técnica anterior

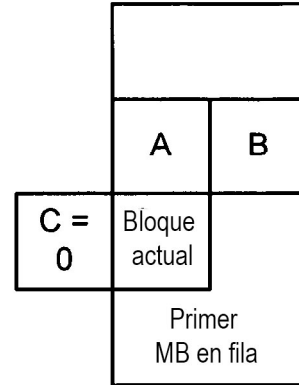


Figura 8A,
técnica anterior

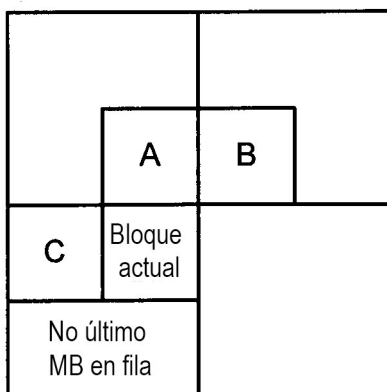


Figura 8B,
técnica anterior

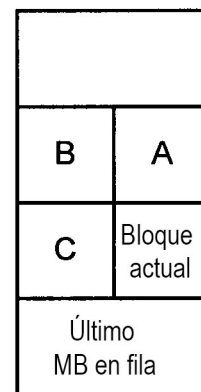


Figura 9,
Técnica anterior

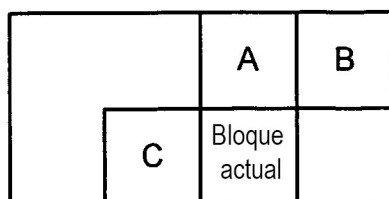


Figura 10,
Técnica anterior

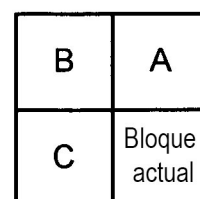


Figura 11, técnica anterior

	Predictor A (calcular promedio si está codificado por campo)	Predictor B (calcular promedio si está codificado por campo)
Predictor C (calcular promedio si está codificado por campo)	macrobloque actual	

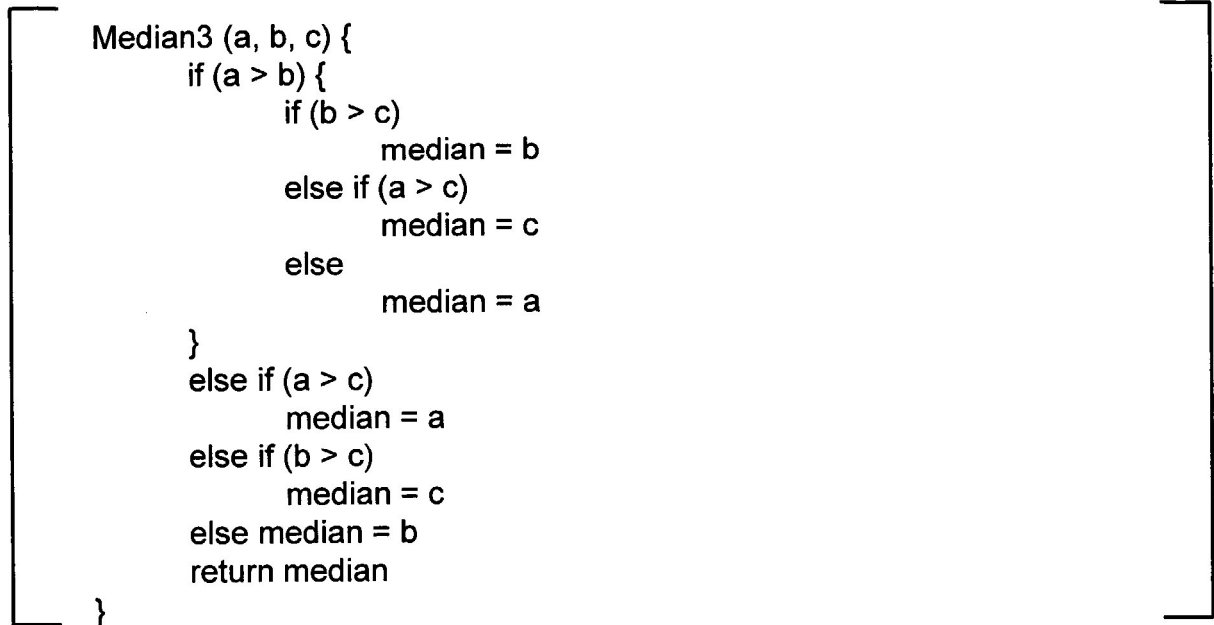
Figura 12A, técnica anterior

	Predictor A	Predictor B
Predictor C	Actual	

Figura 12B, técnica anterior

	Predictor A	Predictor B
Predictor C	Actual	

Figura 13, técnica anterior

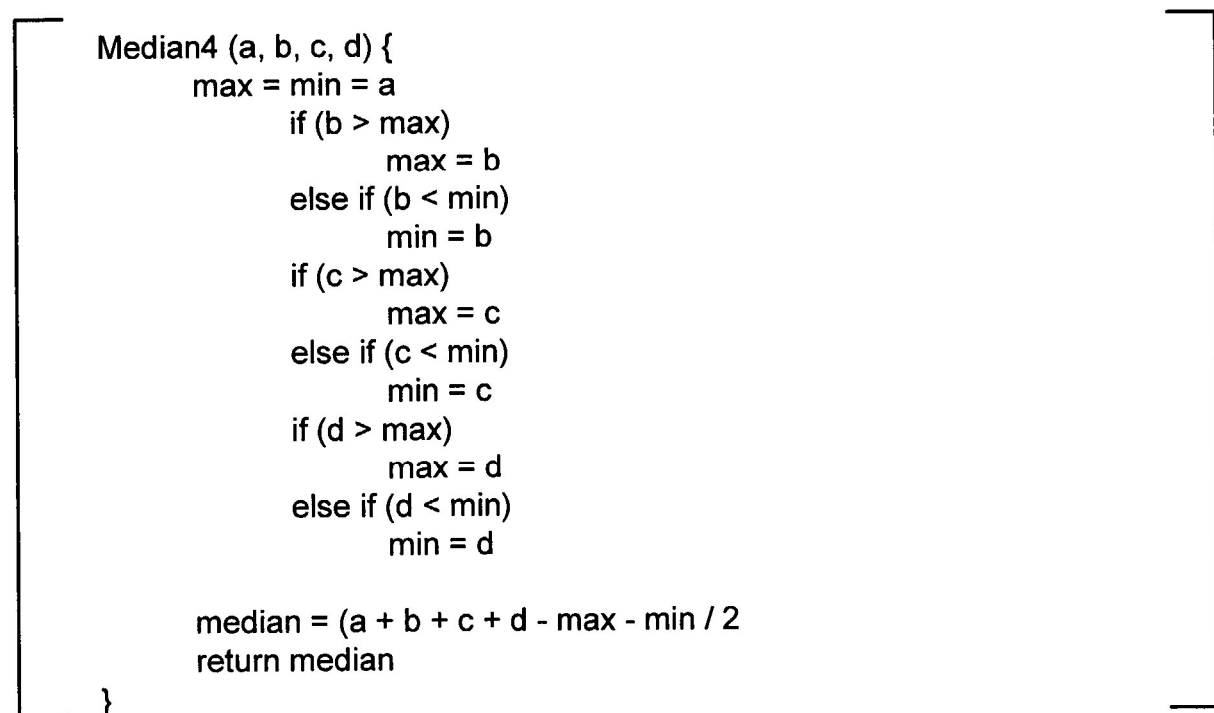


```

Median3 (a, b, c) {
    if (a > b) {
        if (b > c)
            median = b
        else if (a > c)
            median = c
        else
            median = a
    }
    else if (a > c)
        median = a
    else if (b > c)
        median = c
    else median = b
    return median
}

```

Figura 14, técnica anterior



```

Median4 (a, b, c, d) {
    max = min = a
    if (b > max)
        max = b
    else if (b < min)
        min = b
    if (c > max)
        max = c
    else if (c < min)
        min = c
    if (d > max)
        max = d
    else if (d < min)
        min = d

    median = (a + b + c + d - max - min) / 2
    return median
}

```

Figura 15, técnica anterior

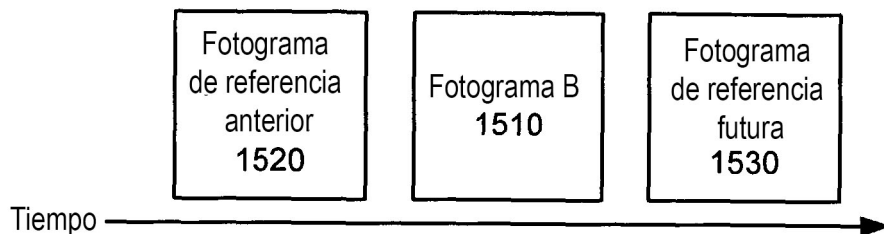


Figura 16, técnica anterior

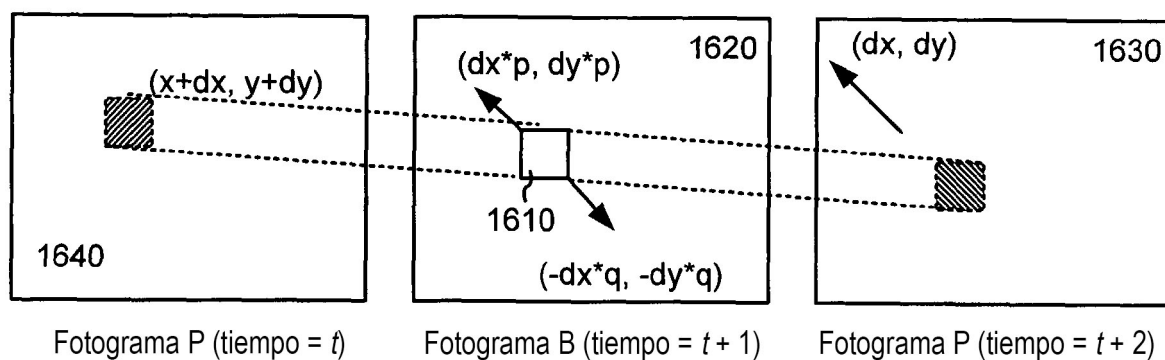


Figura 17, técnica anterior

BFACTION VLC	Fracción	BFACTION VLC	Fracción	BFACTION VLC	Fracción
000	1/2	1110000	3/5	1110111	4/7
001	1/3	1110001	4/5	1111000	5/7
010	2/3	1110010	1/6	1111001	6/7
011	1/4	1110011	5/6	1111010	1/8
100	3/4	1110100	1/7	1111011	3/8
101	1/5	1110101	2/7	1111100	5/8
110	2/5	1110110	3/7	1111101	7/8

1700

Figura 18, técnica anterior

1800

```

Int NumShortVLC[] = {1, 1, 2, 1, 3, 1, 2};
Int DenShortVLC[] = {2, 3, 3, 4, 4, 5, 5};
Int NumLongVLC[] = {3, 4, 1, 5, 1, 2, 3, 4, 5, 6, 1, 3, 5, 7};
Int DenLongVLC[] = {5, 5, 6, 6, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8};
Int Inverse[] = { 256, 128, 85, 64, 51, 43, 37, 32 };

Frame_Initialization(code word)
    if (long code word) {
        Numerator = NumLongVLC[code word - 112];
        Denominator = DenLongVLC[code word - 112];
    }
    else { /* palabra de código corto */
        Numerator = NumShortVLC[code word];
        Denominator = DenShortVLC[code word];
    }
    FrameReciprocal = Inverse[Denominator - 1];
    ScaleFactor = Numerator * FrameReciprocal;
End Frame_Initialization

```

Figura 19, técnica anterior

1900

```

Scale_Direct_MV (IN MV_X, IN MV_Y, OUT MV_X_F, OUT MV_Y_F, OUT
MV_X_B, OUT MV_Y_B)
    if (Half pel units) { // si la instantánea B actual está usando la mitad de MV de pels
        MV_X_F = 2 * ((MV_X * ScaleFactor + 255) >> 9);
        MV_Y_F = 2 * ((MV_Y * ScaleFactor + 255) >> 9);
        MV_X_B = 2 * ((MV_X * (ScaleFactor - 256) + 255) >> 9);
        MV_Y_B = 2 * ((MV_Y * (ScaleFactor - 256) + 255) >> 9);
    }
    else { /* Cuarto de unidades de pels */
        MV_X_F = (MV_X * ScaleFactor + 128) >> 8;
        MV_Y_F = (MV_Y * ScaleFactor + 128) >> 8;
        MV_X_B = (MV_X * (ScaleFactor - 256) + 128) >> 8;
        MV_Y_B = (MV_Y * (ScaleFactor - 256) + 128) >> 8;
    }
End Scale_Direct_MV

```

Figura 20

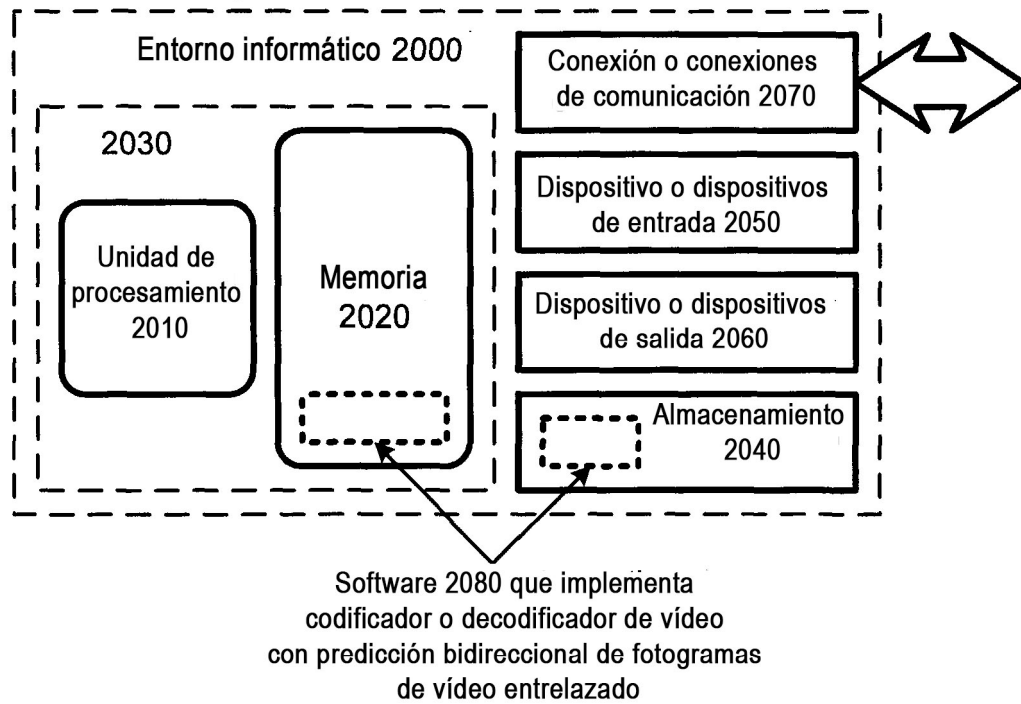


Figura 21

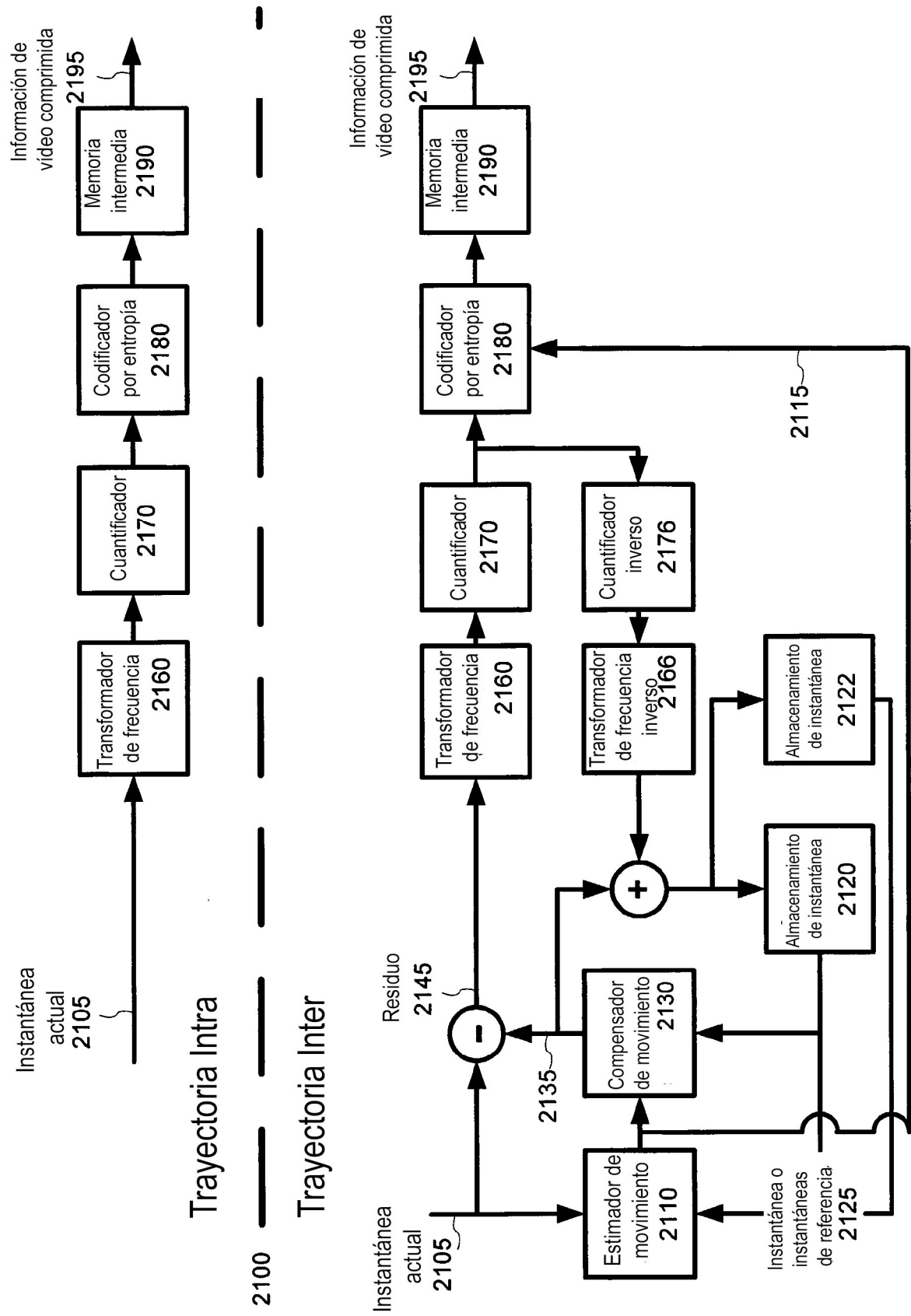


Figura 22

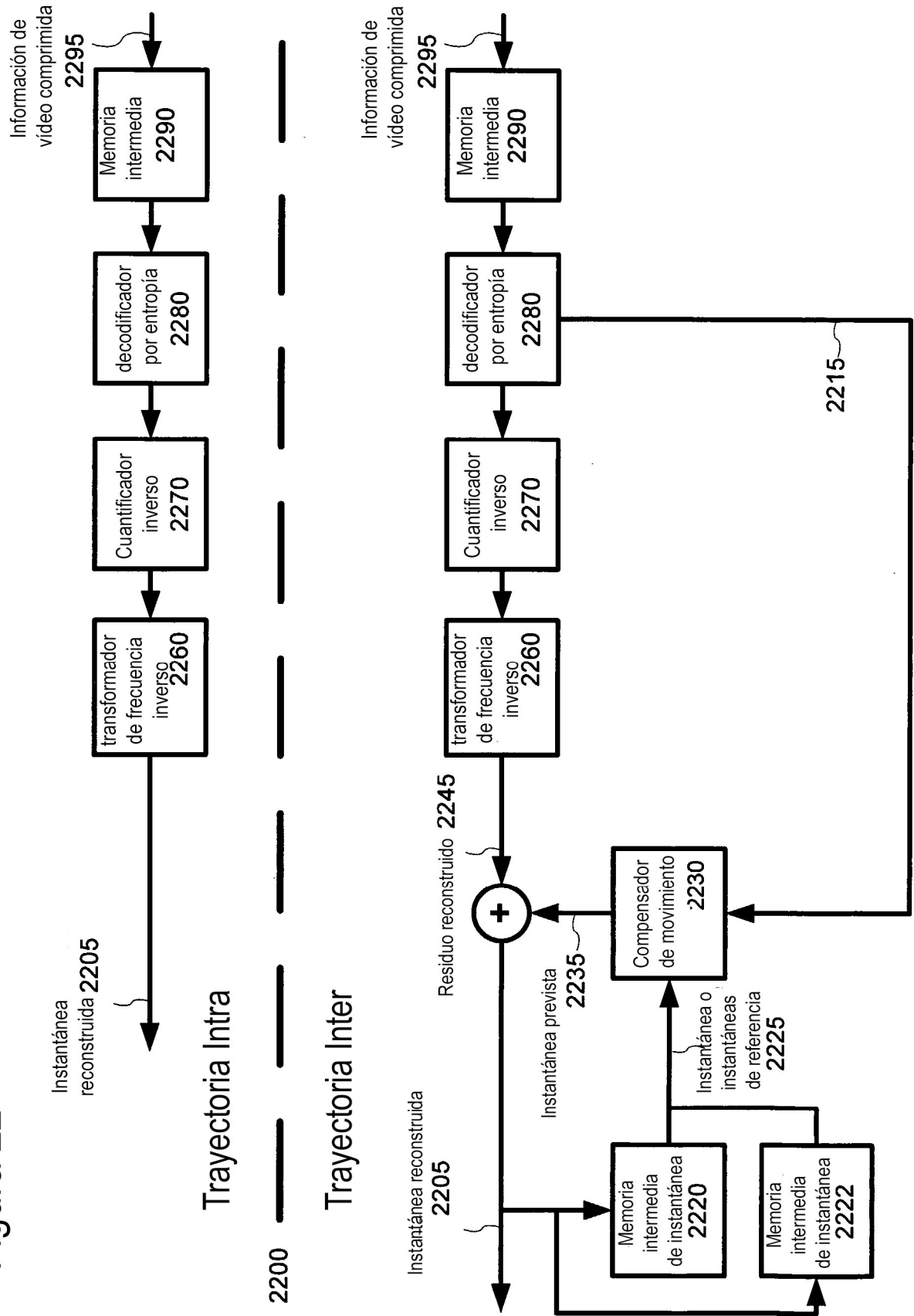


Figura 23

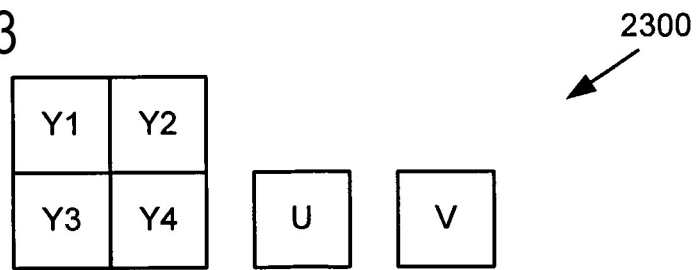


Figura 24A

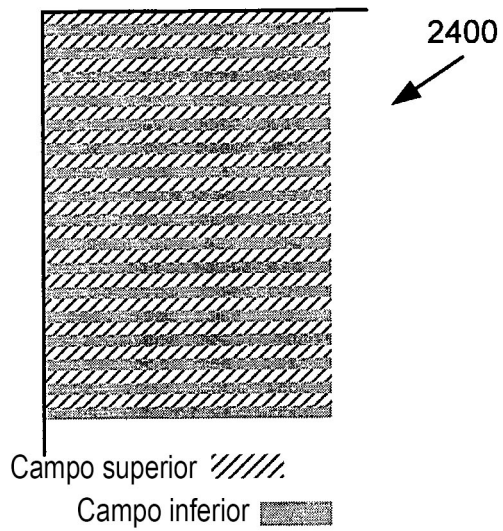


Figura 24B

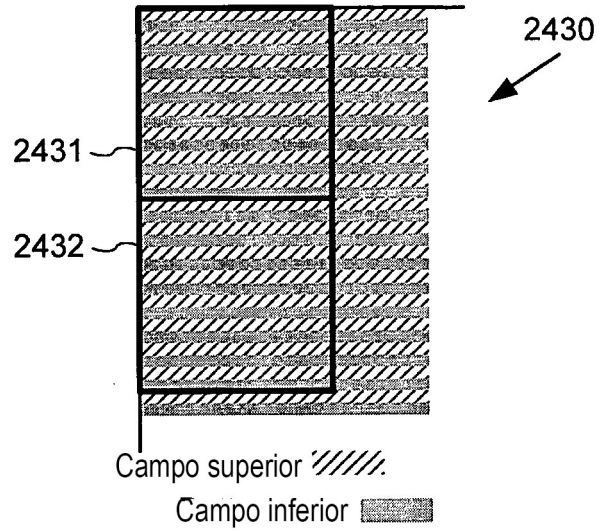


Figura 24C

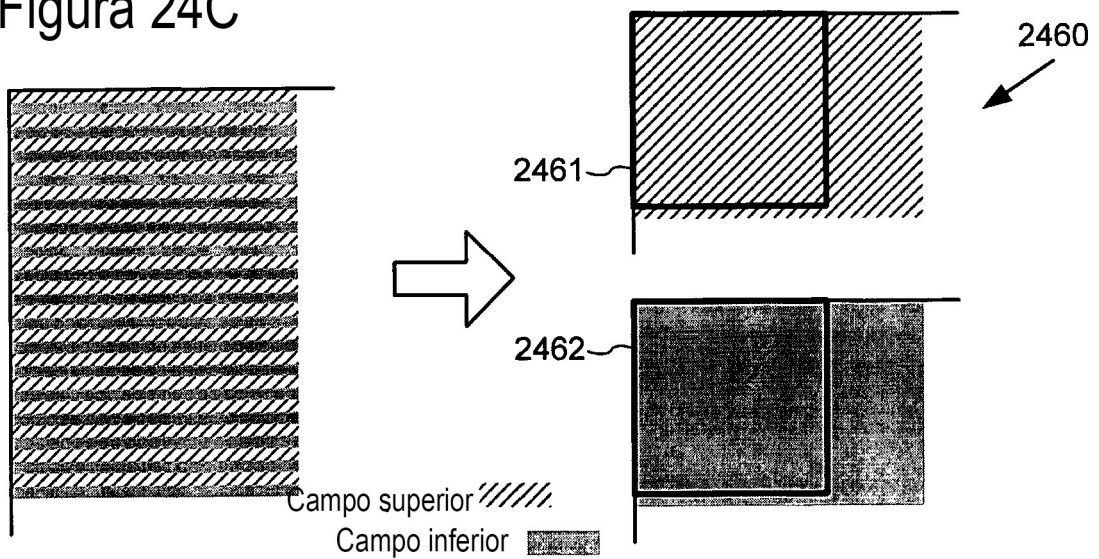


Figura 25

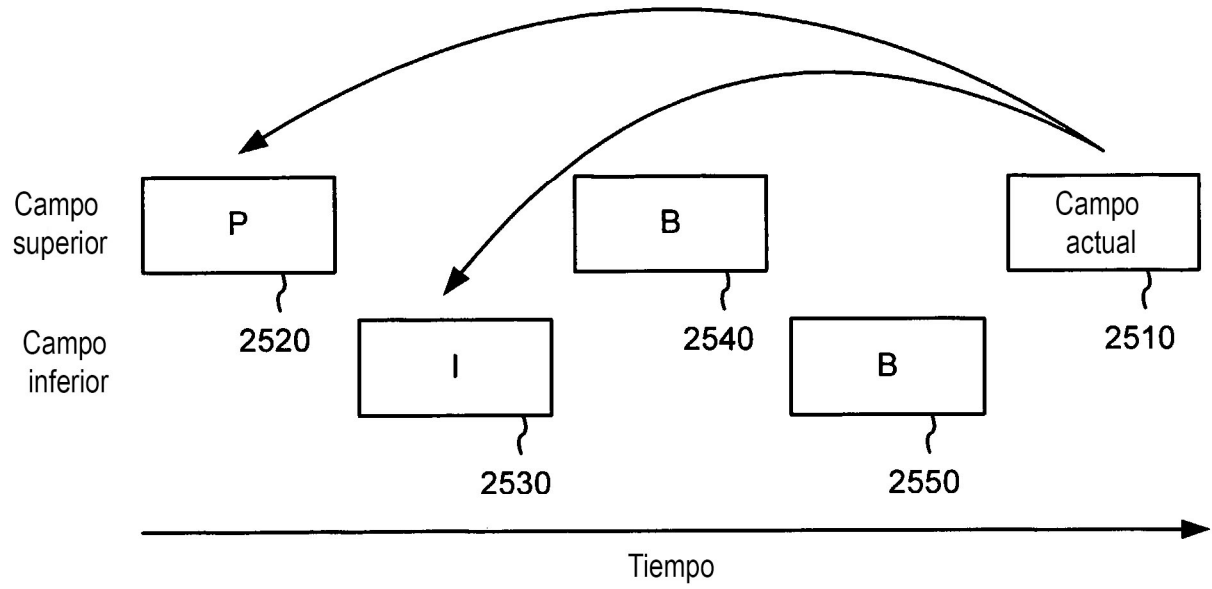


Figura 26

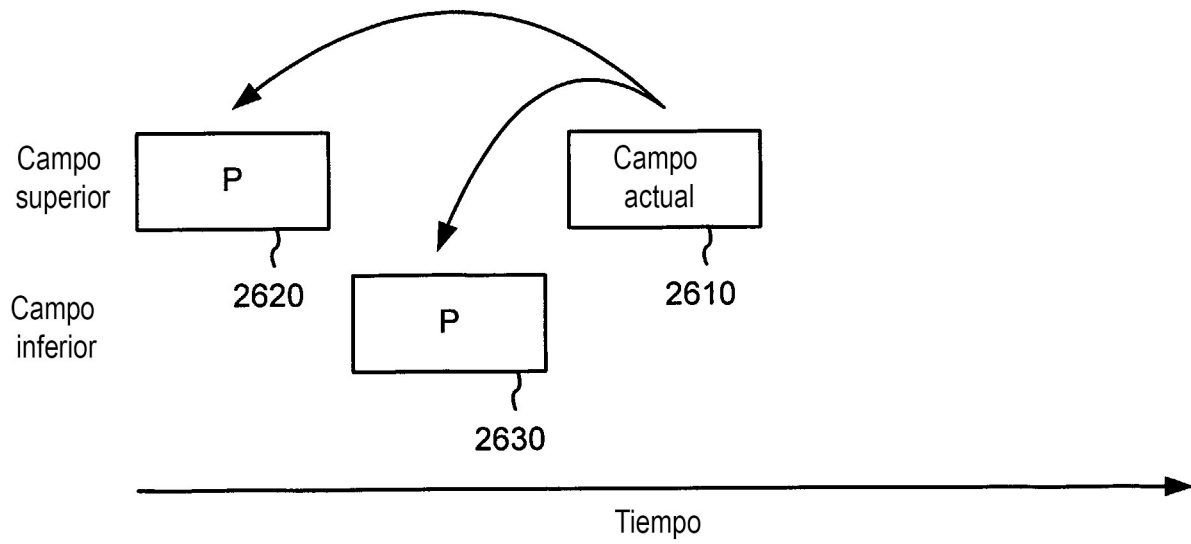


Figura 27

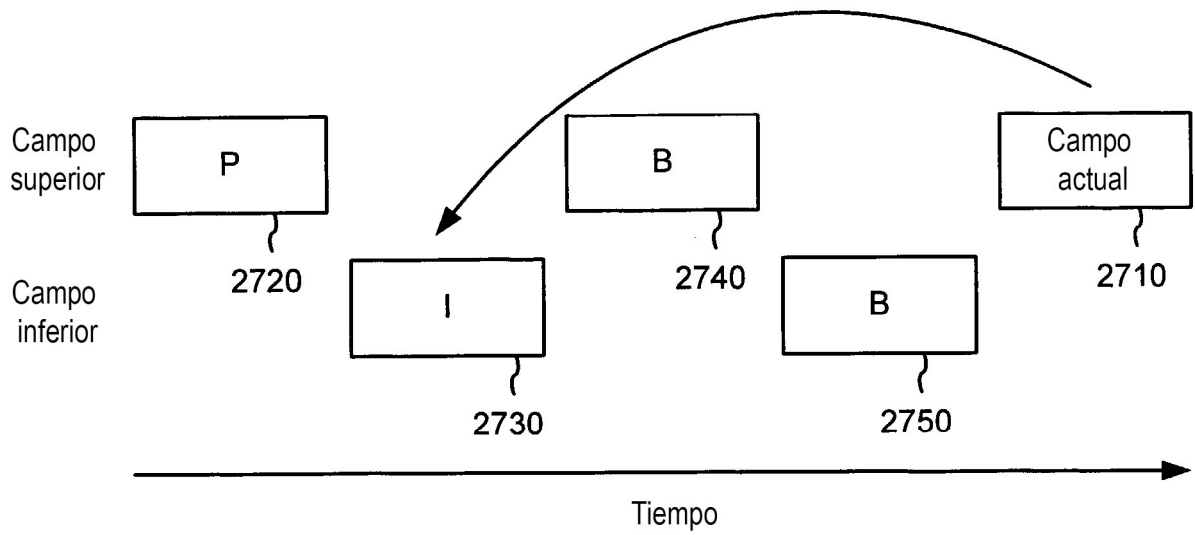


Figura 28

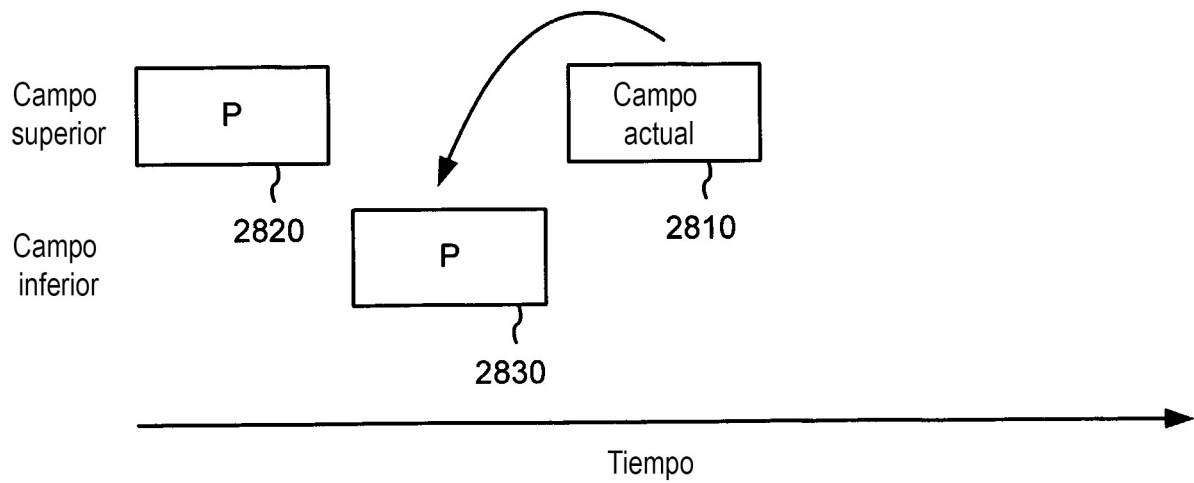


Figura 29

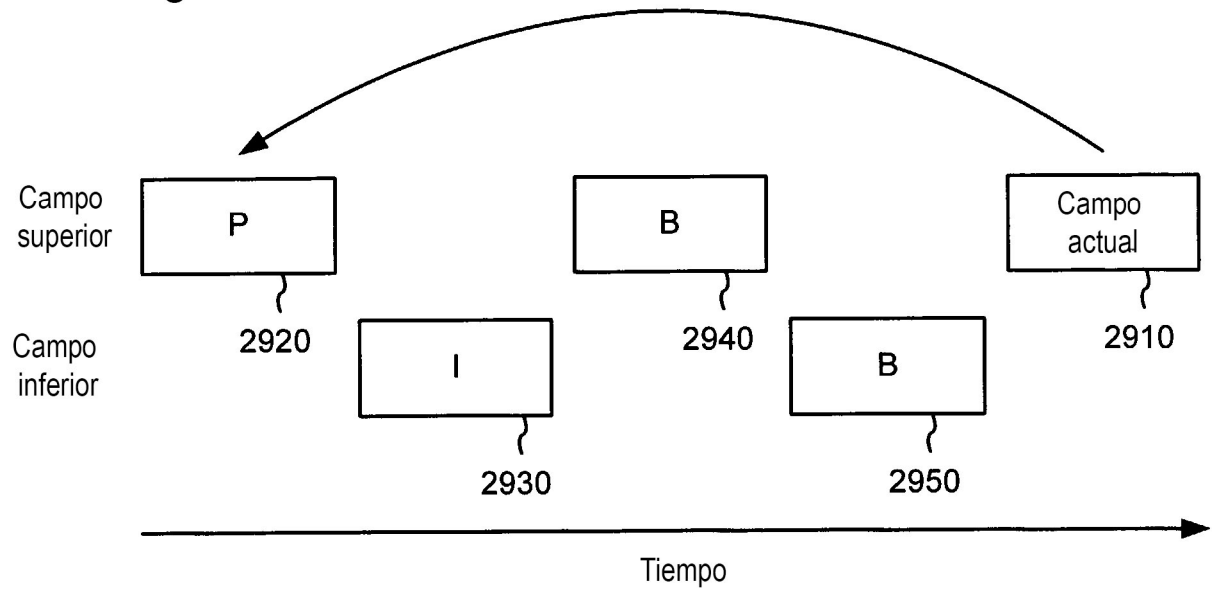


Figura 30

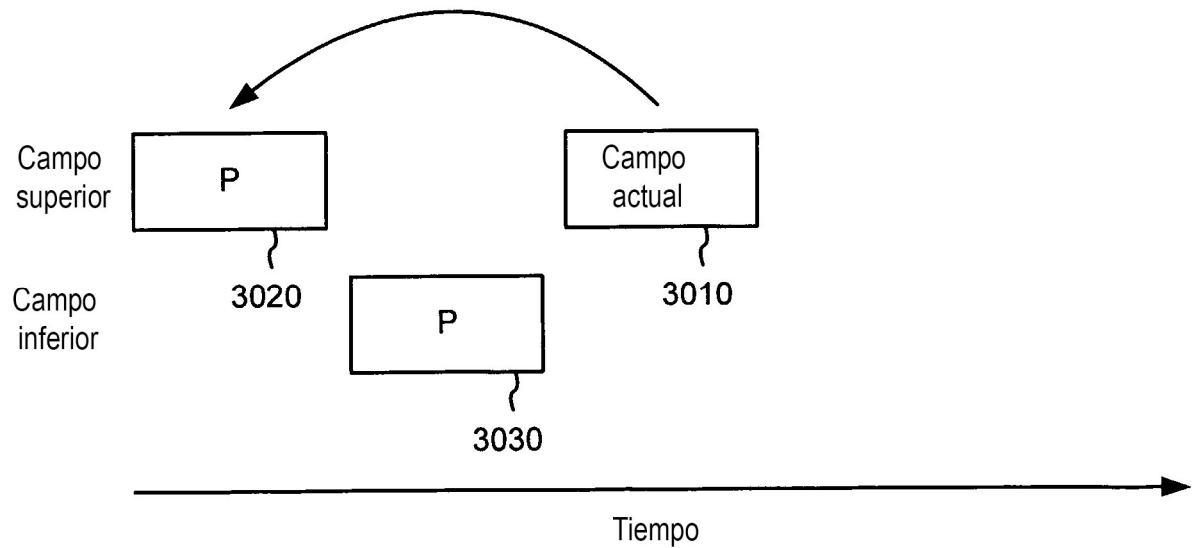


Figura 31

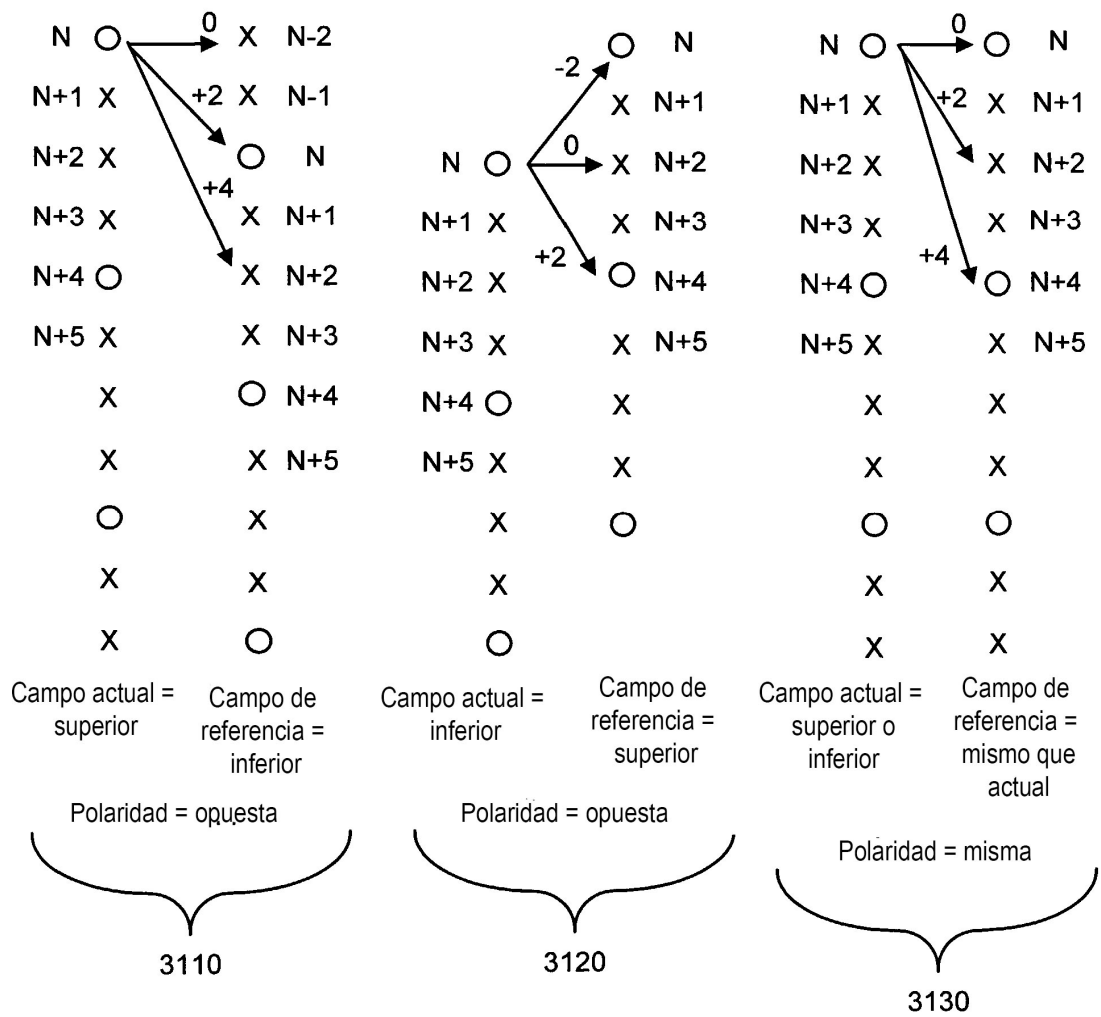


Figura 32

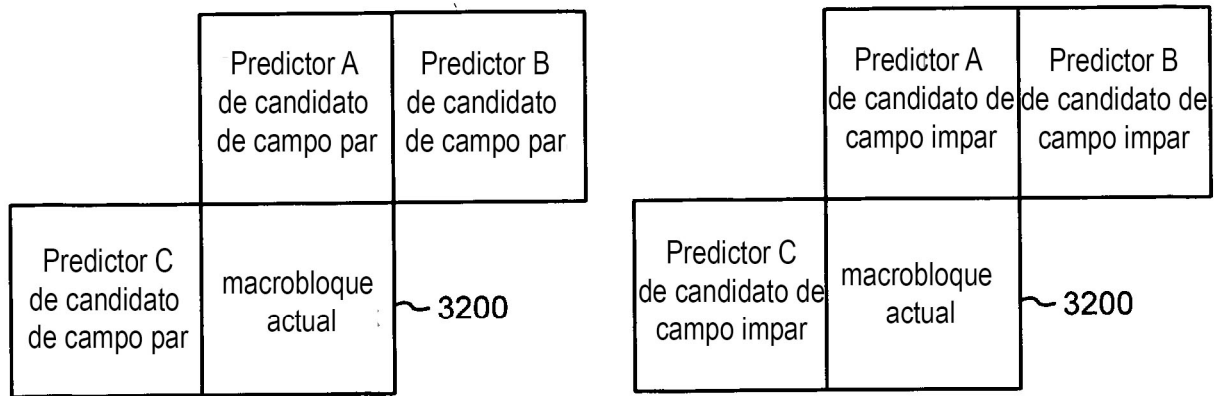


Figura 33A

3300

```

if (predictorA no está fuera de los límites) {
  if (predictorC no está fuera de los límites) {
    if (predictorA es intra) {
      predictorA_x = 0
      predictorA_y = 0
    }
    if (predictorB es intra) {
      predictorB_x = 0
      predictorB_y = 0
    }
    if (predictorC es intra) {
      predictorC_x = 0
      predictorC_y = 0
    }
  }
  if (predictorA es desde el mismo campo) {
    samecount = samecount + 1
    samefieldpredA_x = predictorA_x
    samefieldpredA_y = predictorA_y
    oppositefieldpredA_x = scaleforopposite_x(predictorA_x)
    oppositefieldpredA_y = scaleforopposite_y(predictorA_y)
  }
  else {
    oppositecount = oppositecount + 1
    oppositefieldpredA_x = predictorA_x
    oppositefieldpredA_y = predictorA_y
    samefieldpredA_x = scaleforsame_x(predictorA_x)
    samefieldpredA_y = scaleforsame_y(predictorA_y)
  }
  if (predictorB es desde el mismo campo) {
    samecount = samecount + 1
    samefieldpredB_x = predictorB_x
    samefieldpredB_y = predictorB_y
    oppositefieldpredB_x = scaleforopposite_x(predictorB_x)
    oppositefieldpredB_y = scaleforopposite_y(predictorB_y)
  }
  else {
    oppositecount = oppositecount + 1
    oppositefieldpredB_x = predictorB_x
    oppositefieldpredB_y = predictorB_y
    samefieldpredB_x = scaleforsame_x(predictorB_x)
    samefieldpredB_y = scaleforsame_y(predictorB_y)
  }
}

```

Figura 33B

3300

```

if (predictorC es desde el mismo campo) {
    samecount = samecount + 1
    samefieldpredC_x = predictorC_x
    samefieldpredC_y = predictorC_y
    oppositefieldpredC_x = scaleforopposite_x(predictorC_x)
    oppositefieldpredC_y = scaleforopposite_y(predictorC_y)
}
else {
    oppositecount = oppositecount + 1
    oppositefieldpredC_x = predictorC_x
    oppositefieldpredC_y = predictorC_y
    samefieldpredC_x = scaleforsame_x(predictorC_x)
    samefieldpredC_y = scaleforsame_y(predictorC_y)
}
samefieldpred_x =
    median (samefieldpredA_x, samefieldpredB_x, samefieldpredC_x)
samefieldpred_y =
    median (samefieldpredA_y, samefieldpredB_y, samefieldpredC_y)
oppositefieldpred_x =
    median (oppositefieldpredA_x, oppositefieldpredB_x, oppositefieldpredC_x)
oppositefieldpred_y =
    median (oppositefieldpredA_y, oppositefieldpredB_y, oppositefieldpredC_y)

if (samecount > oppositecount)
    dominantpredictor = samefield
else
    dominantpredictor = oppositefield
}

```

Figura 33C

3300

```

else {
    // PredictorC está fuera de los límites
    if (únicamente 1 macrobloque por fila) {
        if (predictorA esintra) {
            samefieldpred_x = oppositefieldpred_x = 0
            samefieldpred_y = oppositefieldpred_y = 0
            dominantpredictor = oppositefield
        }
        else {
            // Usar predictorA
            if (predictorA es desde el mismo campo) {
                samefieldpred_x = predictorA_x
                samefieldpred_y = predictorA_y
                oppositefieldpred_x = scaleforopposite_x(predictorA_x)
                oppositefieldpred_y = scaleforopposite_y(predictorA_y)
                dominantpredictor = samefield
            }
            else {
                oppositefieldpred_x = predictorA_x
                oppositefieldpred_y = predictorA_y
                samefieldpred_x = scaleforsame_x(predictorA_x)
                samefieldpred_y = scaleforsame_y(predictorA_y)
                dominantpredictor = oppositefield
            }
        }
    }
}

```


Figura 33D

3300

```

else {
    // PredictorC está fuera de los límites, usar PredictorA y Predictor B
    predictorC_x = 0
    predictorC_y = 0
    if (predictorA es intra) {
        predictorA_x = 0
        predictorA_y = 0
    }
    if (predictorB es intra) {
        predictorB_x = 0
        predictorB_y = 0
    }
    if (predictorC es intra) {
        predictorC_x = 0
        predictorC_y = 0
    }
    if (predictorA es desde el mismo campo) {
        samecount = samecount + 1
        samefieldpredA_x = predictorA_x
        samefieldpredA_y = predictorA_y
        oppositefieldpredA_x = scaleforopposite_x(predictorA_x)
        oppositefieldpredA_y = scaleforopposite_y(predictorA_y)
    }
    else {
        oppositecount = oppositecount + 1
        oppositefieldpredA_x = predictorA_x
        oppositefieldpredA_y = predictorA_y
        samefieldpredA_x = scaleforsame_x(predictorA_x)
        samefieldpredA_y = scaleforsame_y(predictorA_y)
    }
}

```

Figura 33E

3300

```

if (predictorB es desde el mismo campo) {
    samecount = samecount + 1
    samefieldpredB_x = predictorB_x
    samefieldpredB_y = predictorB_y
    oppositefieldpredB_x = scaleforopposite_x(predictorB_x)
    oppositefieldpredB_y = scaleforopposite_y(predictorB_y)
}
else {
    oppositecount = oppositecount + 1
    oppositefieldpredB_x = predictorB_x
    oppositefieldpredB_y = predictorB_y
    samefieldpredB_x = scaleforsame_x(predictorB_x)
    samefieldpredB_y = scaleforsame_y(predictorB_y)
}
samefieldpred_x =
    median (samefieldpredA_x, samefieldpredB_x, 0)
samefieldpred_y =
    median (samefieldpredA_y, samefieldpredB_y, 0)
oppositefieldpred_x =
    median (oppositefieldpredA_x, oppositefieldpredB_x, 0)
oppositefieldpred_y =
    median (oppositefieldpredA_y, oppositefieldpredB_y, 0)
if (samecount > oppositecount)
    dominantpredictor = samefield
else
    dominantpredictor = oppositefield
}
}

```

Figura 33F

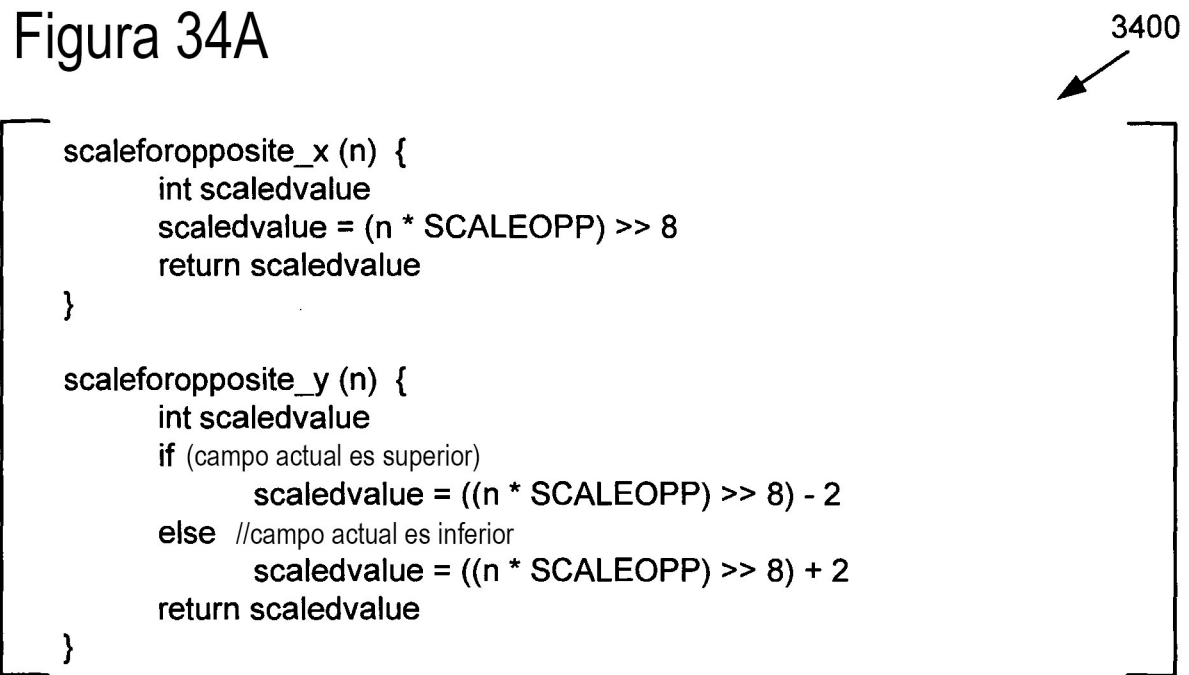
3300

```

else {
    // Predictor A está fuera de los límites
    if (predictorC está fuera de los límites) {
        samefieldpred_x = oppositefieldpred_x = 0
        samefieldpred_y = oppositefieldpred_y = 0
        dominantpredictor = oppositefield
    }
    else {
        // Usar predictorC
        if (predictorC es desde el mismo campo) {
            samefieldpred_x = predictorC_x
            samefieldpred_y = predictorC_y
            oppositefieldpred_x = scaleforopposite_x(predictorC_x)
            oppositefieldpred_y = scaleforopposite_y(predictorC_y)
            dominantpredictor = samefield
        }
        else {
            oppositefieldpred_x = predictorC_x
            oppositefieldpred_y = predictorC_y
            samefieldpred_x = scaleforsame_x(predictorC_x)
            samefieldpred_y = scaleforsame_y(predictorC_y)
            dominantpredictor = oppositefield
        }
    }
}

```

Figura 34A



```
scaleforopposite_x (n) {  
    int scaledvalue  
    scaledvalue = (n * SCALEOPP) >> 8  
    return scaledvalue  
}  
  
scaleforopposite_y (n) {  
    int scaledvalue  
    if (campo actual es superior)  
        scaledvalue = ((n * SCALEOPP) >> 8) - 2  
    else //campo actual es inferior  
        scaledvalue = ((n * SCALEOPP) >> 8) + 2  
    return scaledvalue  
}
```

Figura 34B

3400


```

scaleforsame_x (n) {
  if (abs (n) < SCALEZONE1_X)
    scaledvalue = (n * SCALESAME1) >> 8
  else {
    if (n < 0)
      scaledvalue = ((n * SCALESAME2) >> 8) - ZONE1OFFSET_X
    else
      scaledvalue = ((n * SCALESAME2) >> 8) + ZONE1OFFSET_X
  }
  return scaledvalue
}

scaleforsame_y (n) {
  if (campo actual es superior) {
    if (abs (n) < SCALEZONE1_Y)
      scaledvalue = (n * SCALESAME1) >> 8
    else {
      if (n < 0)
        scaledvalue = ((n * SCALESAME2) >> 8) - ZONE1OFFSET_Y
      else
        scaledvalue = ((n * SCALESAME2) >> 8) + ZONE1OFFSET_Y
    }
  }
  else { //campo actual es inferior
    if (abs (n) < SCALEZONE1_Y)
      scaledvalue = (n * SCALESAME1) >> 8
    else {
      if (n < 0)
        scaledvalue = ((n * SCALESAME2) >> 8) - ZONE1OFFSET_Y
      else
        scaledvalue = ((n * SCALESAME2) >> 8) + ZONE1OFFSET_Y
    }
  }
  return scaledvalue
}

```

Figura 35

3500

	Distancia de fotograma de referencia			
	1	2	3	4 o mayor
SCALEOPP	128	192	213	224
SCALESAME1	512	341	307	293
SCALESAME2	219	236	242	245
SCALEZONE1_X	32 * N	48 * N	53 * N	56 * N
SCALEZONE1_Y	8 * N	12 * N	13 * N	14 * N
ZONE1OFFSET_X	37 * N	20 * N	14 * N	11 * N
ZONE1OFFSET_Y	10 * N	5 * N	4 * N	3 * N

Figura 36

3600

	Distancia de fotograma de referencia			
	1	2	3	4 o mayor
SCALEOPP	128	64	43	32
SCALESAME1	512	1024	1536	2048
SCALESAME2	219	204	200	198
SCALEZONE1_X	32 * N	16 * N	11 * N	8 * N
SCALEZONE1_Y	8 * N	4 * N	3 * N	2 * N
ZONE1OFFSET_X	37 * N	52 * N	56 * N	11 * N
ZONE1OFFSET_Y	10 * N	5 * N	4 * N	3 * N

Figura 37

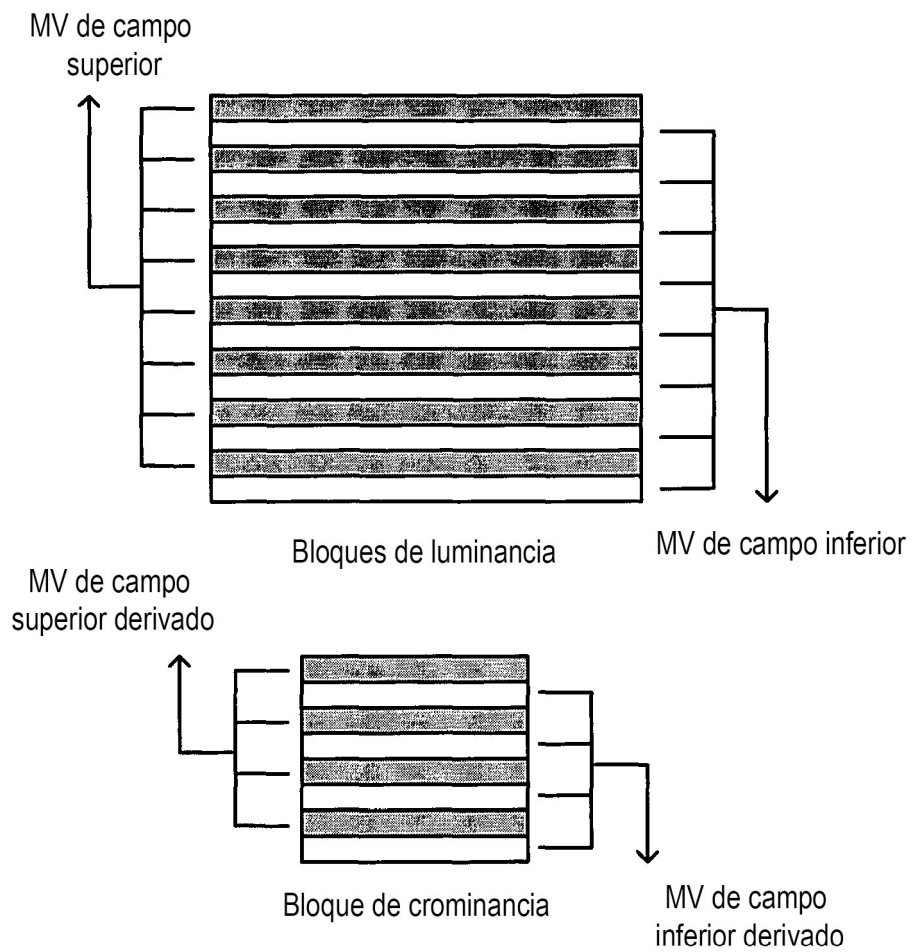


Figura 38

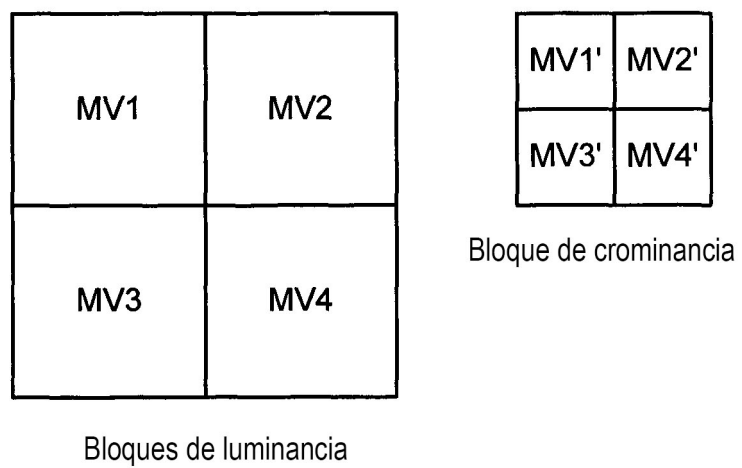


Figura 39

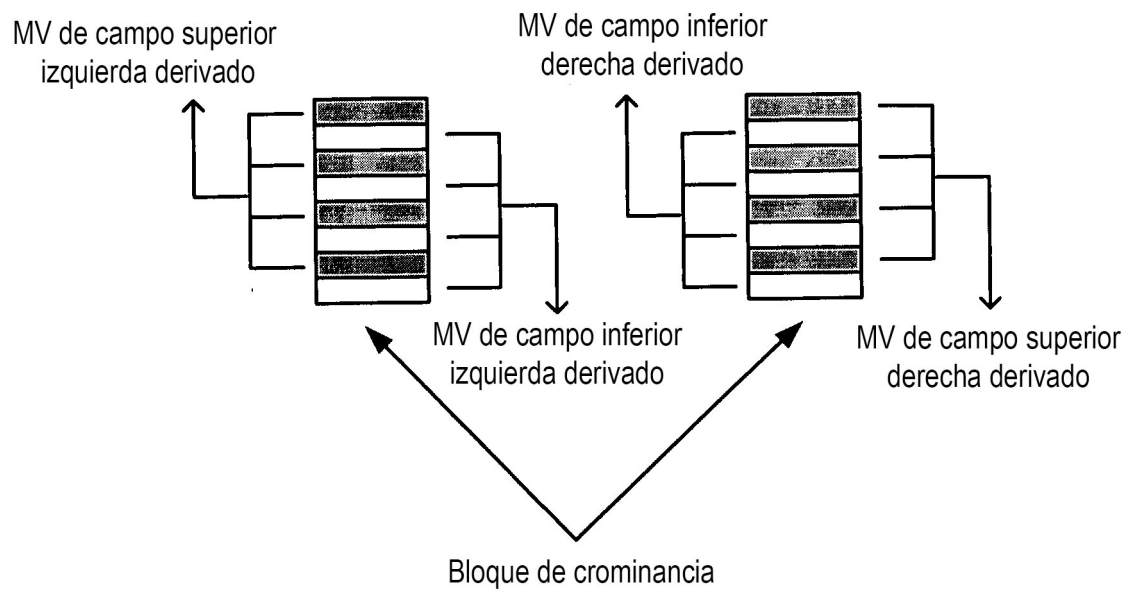
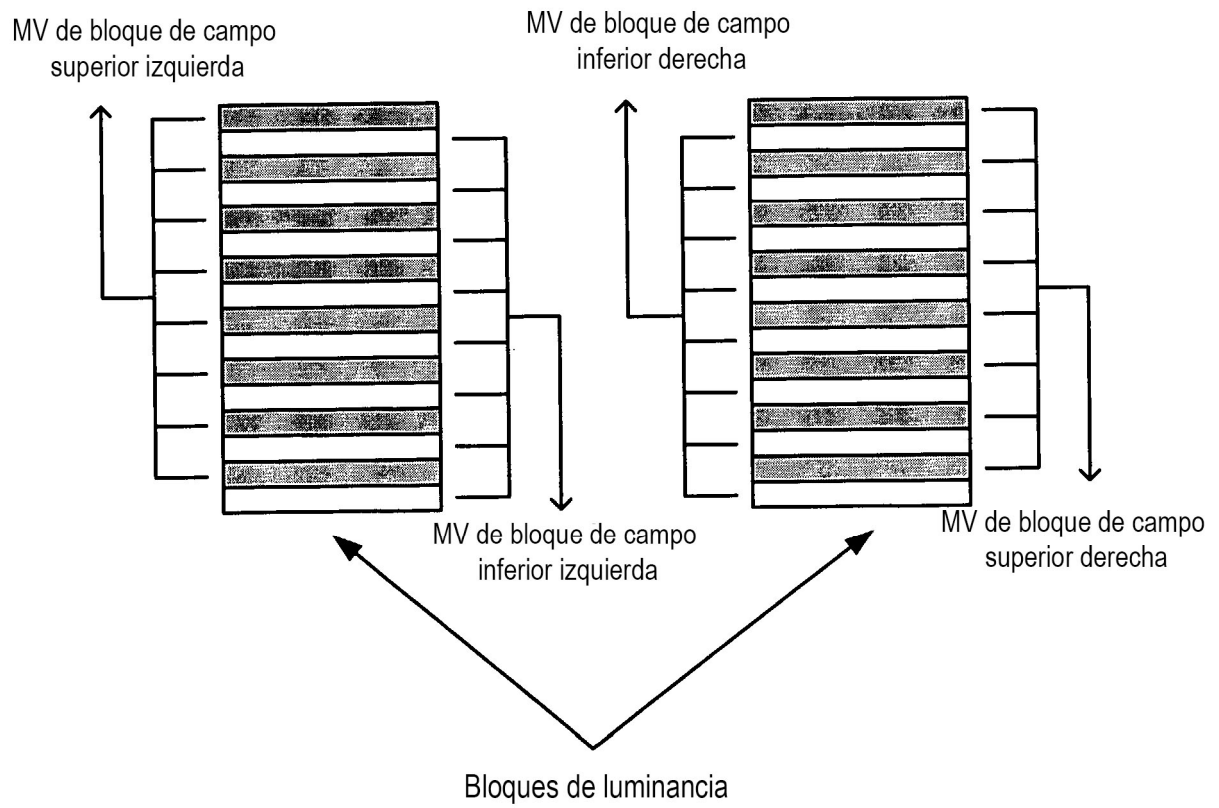


Figura 40A

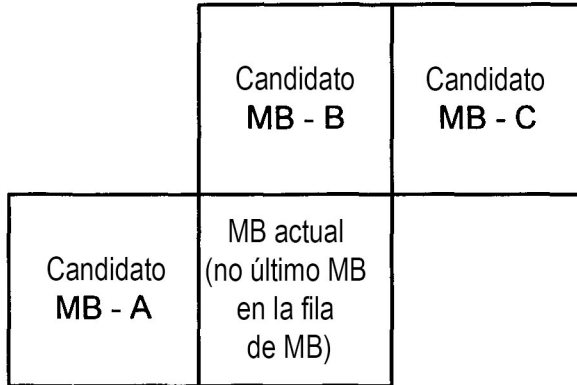


Figura 40B

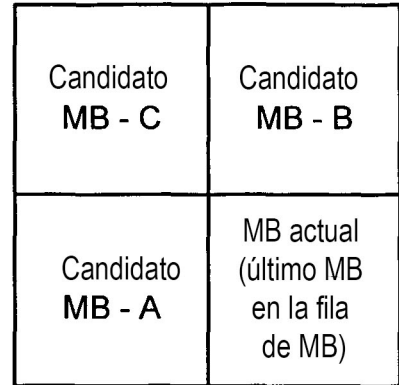


Figura 41

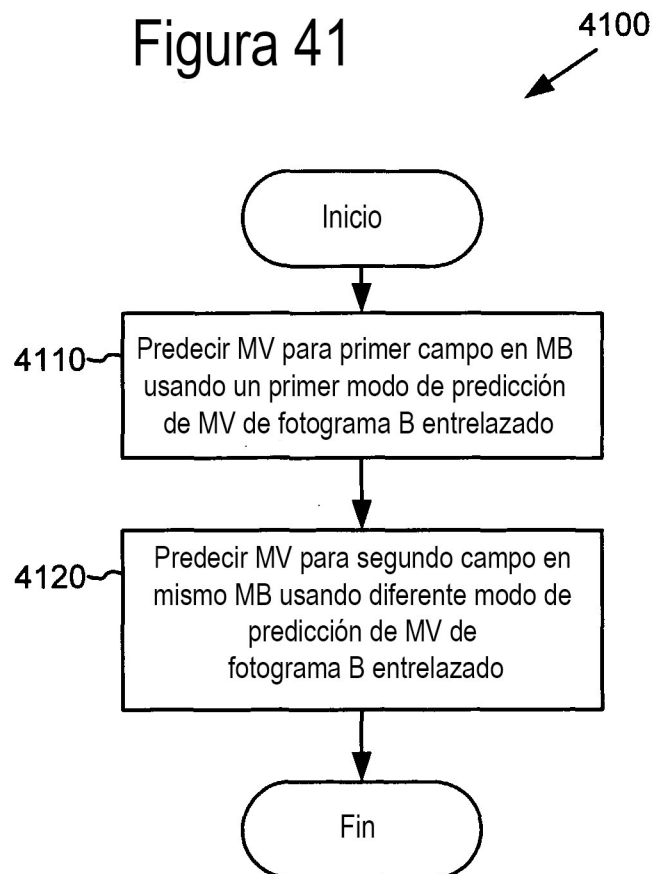


Figura 42

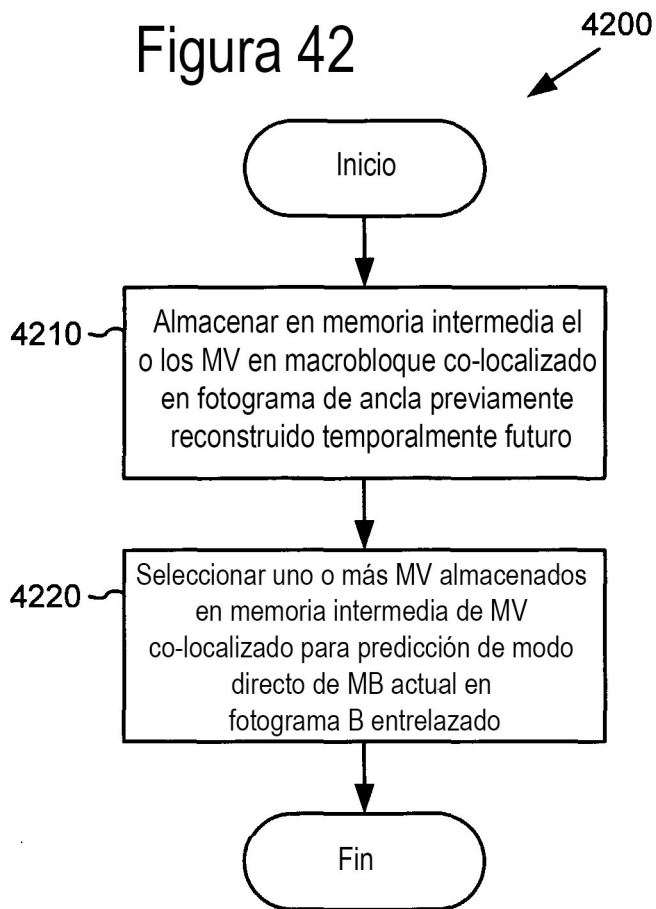


Figura 43

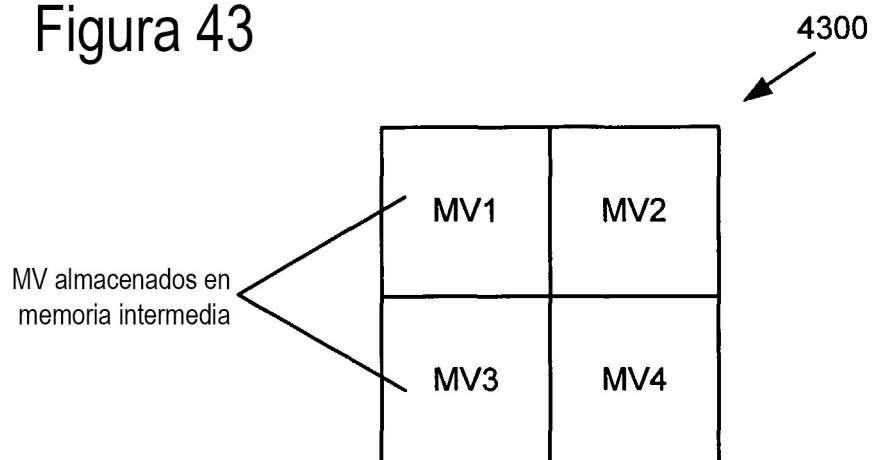


Figura 44

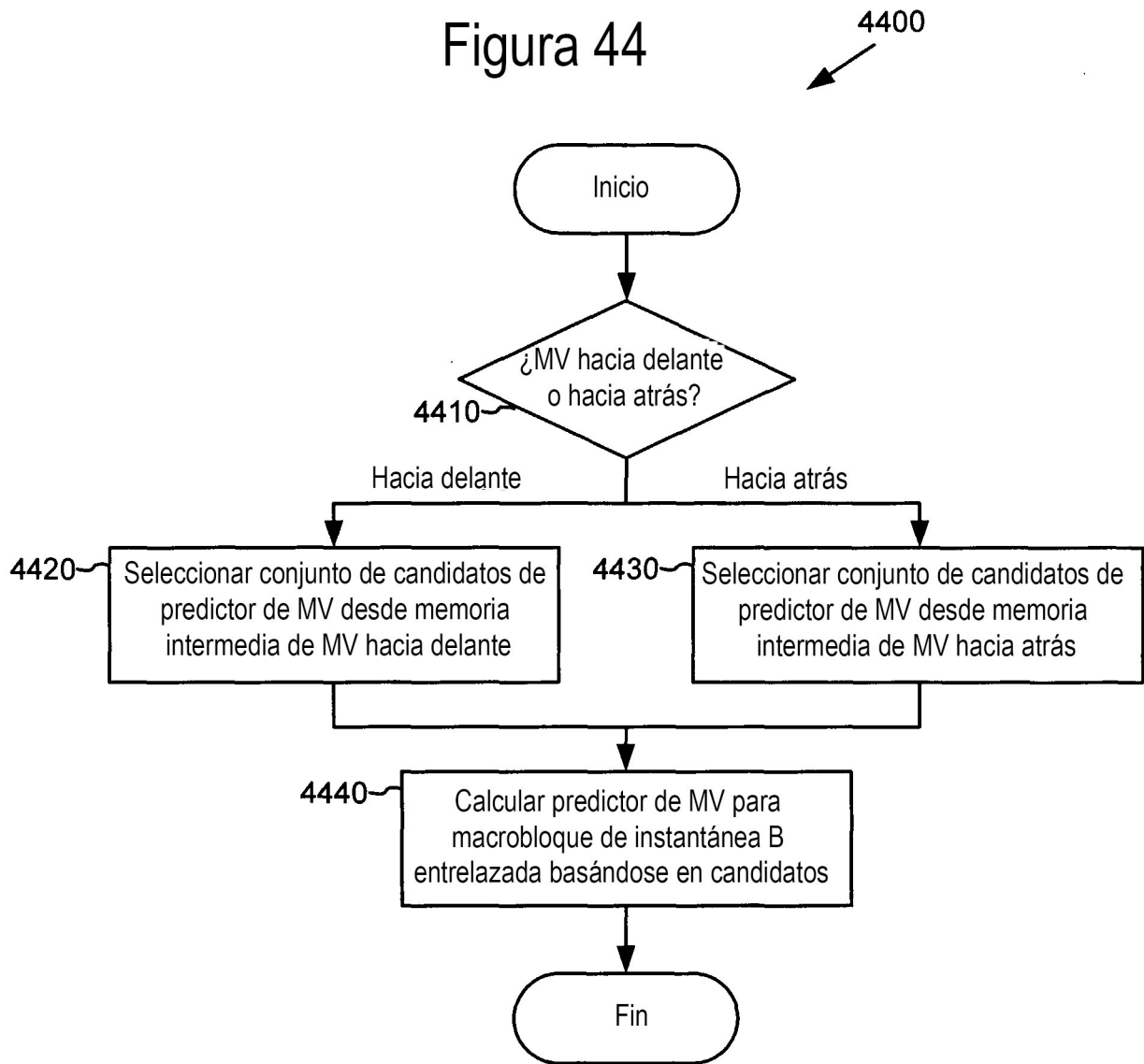


Figura 45

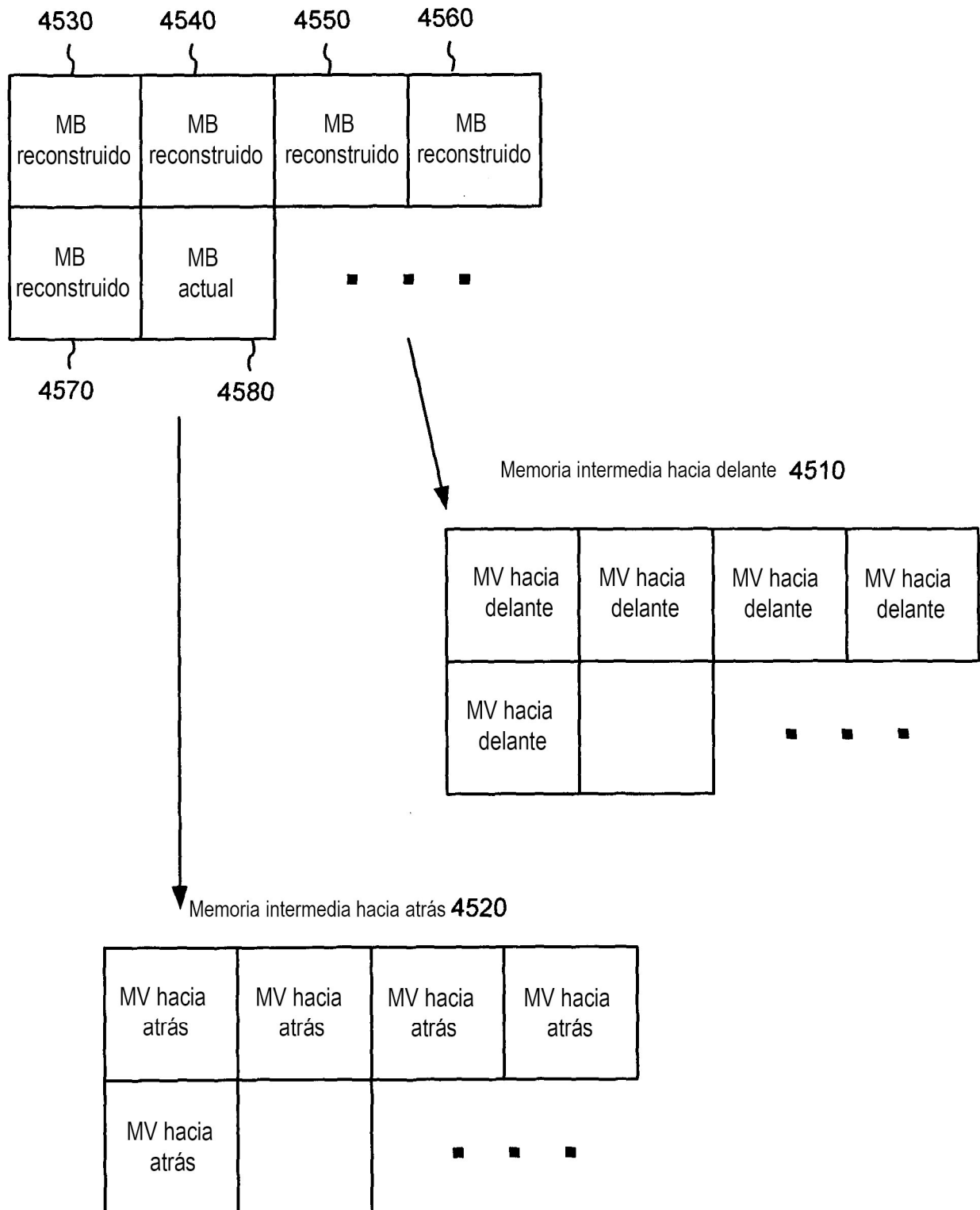


Figura 46

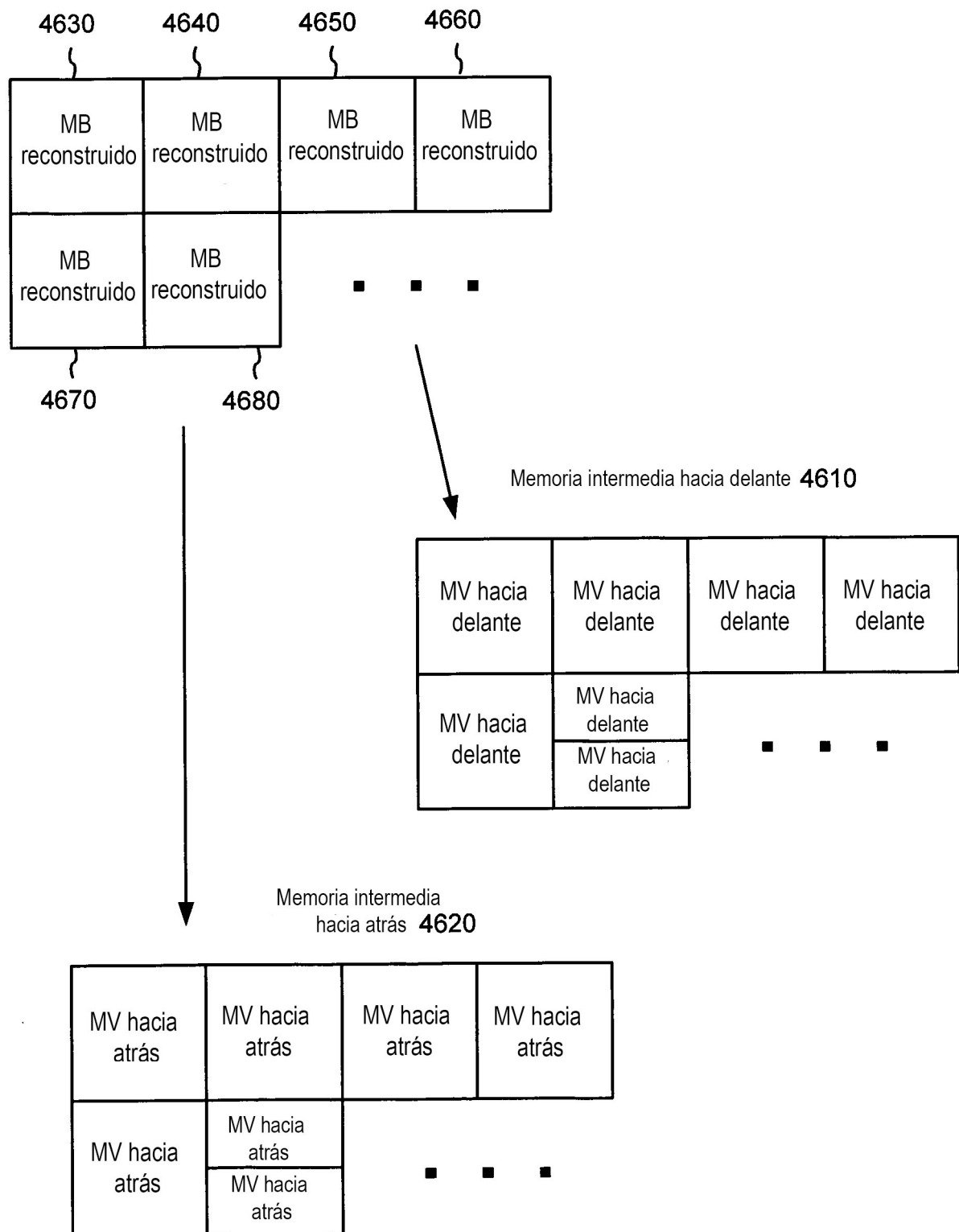


Figura 47

4700



```

PredictedMV PredictMVFieldBPicture (Tipo de pred.) // real o "relleno de huecos"
{
    if ( Tipo de predicción  = Real) {
        //en este caso recibimos un MV real de este tipo
        entonces usar la predicción de MV normal descrita anteriormente con
            la polaridad del MV real recibido
    }
    else {
        // predicción de relleno de huecos - cuando decidimos no recibir un
        // MV real de este tipo
        usar la predicción de MV de campo, y tomar el MV previsto del
            campo más dominante
    }
    return el MV previsto
}

```

Figura 48A

4800


```
scaleforsame_x (n) {  
    int scaledvalue  
    scaledvalue = (n * SCALESAME) >> 8  
    return scaledvalue  
}  
  
scaleforsame_y (n) {  
    int scaledvalue  
    if (campo actual es superior)  
        scaledvalue = ((n * SCALESAME) >> 8) - 2  
    else //campo actual es inferior  
        scaledvalue = ((n * SCALESAME) >> 8) + 2  
    return scaledvalue  
}
```

Figura 48B

4800


```

scaleforopposite_x (n) {
  if (abs (n) < SCALEZONE1_X)
    scaledvalue = (n * SCALEOPP1) >> 8
  else {
    if (n < 0)
      scaledvalue = ((n * SCALEOPP2) >> 8) - ZONE1OFFSET_X
    else
      scaledvalue = ((n * SCALEOPP2) >> 8) + ZONE1OFFSET_X
  }
  return scaledvalue
}

scaleforopposite_y (n) {
  if (campo actual es superior) {
    if (abs (n) < SCALEZONE1_Y)
      scaledvalue = ((n + 2) * SCALEOPP1) >> 8
    else {
      if (n < 0)
        scaledvalue = (((n + 2) * SCALEOPP2) >> 8) - ZONE1OFFSET_Y
      else
        scaledvalue = (((n + 2) * SCALEOPP2) >> 8) + ZONE1OFFSET_Y
    }
  }
  else { //campo actual es inferior
    if (abs (n) < SCALEZONE1_Y)
      scaledvalue = ((n - 2) * SCALEOPP1) >> 8
    else {
      if (n < 0)
        scaledvalue = (((n - 2) * SCALEOPP2) >> 8) - ZONE1OFFSET_Y
      else
        scaledvalue = (((n - 2) * SCALEOPP2) >> 8) + ZONE1OFFSET_Y
    }
  }

  return scaledvalue
}

```


Figura 49

4900
↙

	Distancia de instantánea de referencia			
	1	2	3	4 o mayor
SCALESAME	171	205	219	228
SCALEOPP1	384	320	299	288
SCALEOPP2	230	239	244	246
SCALEZONE1_X	32 * N	48 * N	53 * N	56 * N
SCALEZONE1_Y	8 * N	12 * N	13 * N	14 * N
ZONE1OFFSET_X	37 * N	20 * N	14 * N	11 * N
ZONE1OFFSET_Y	10 * N	5 * N	4 * N	3 * N

Figura 50A

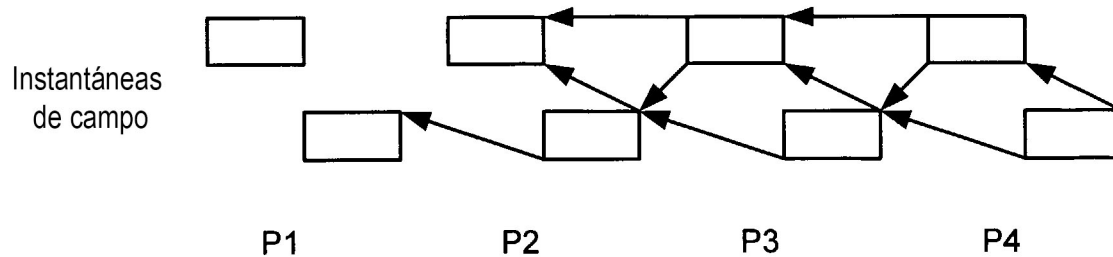


Figura 50B

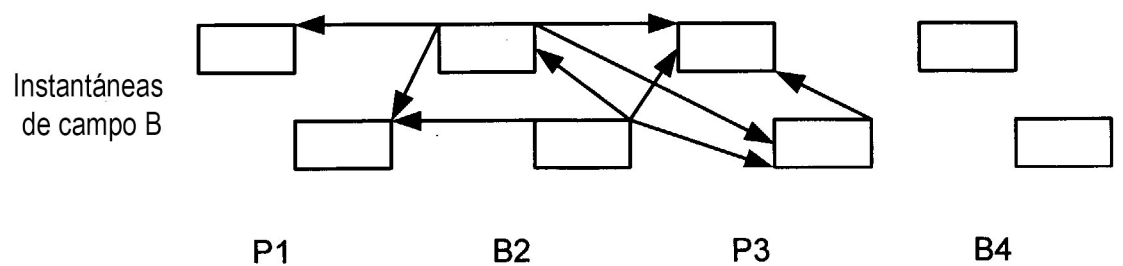


Figura 51

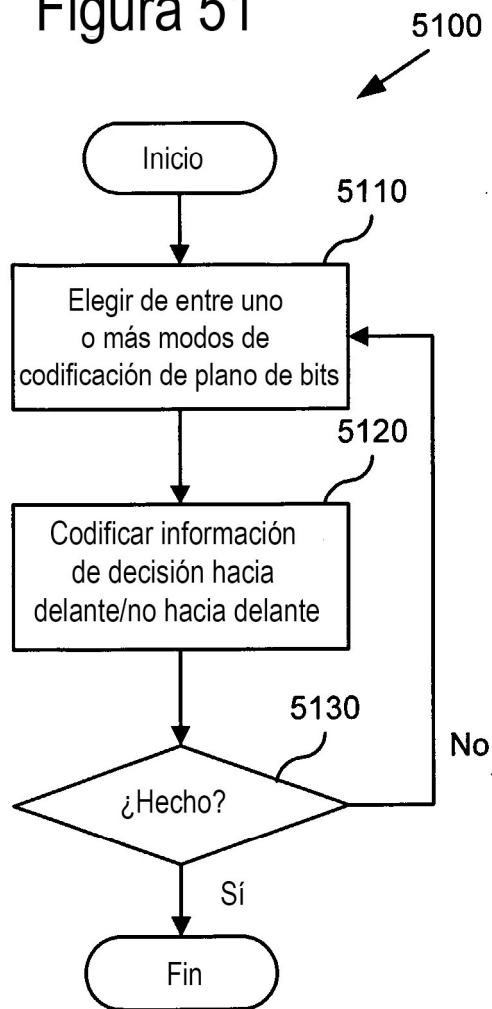


Figura 52

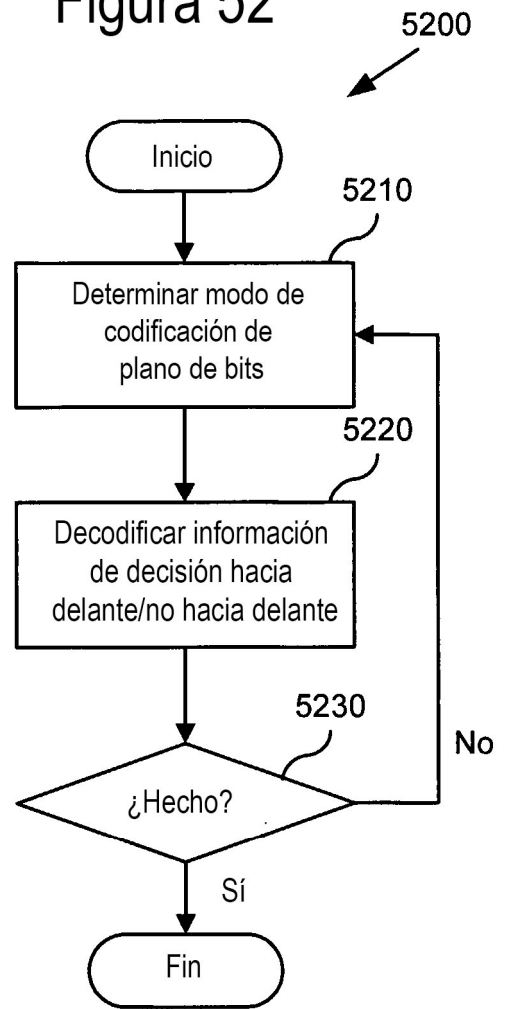


Figura 53

5300

```

MotionVector SelectDirectModeMVFromColocatedMB ()
{
    MotionVector SelectedMV;

    if el correspondiente MB usó 1 MV
        then SelectedMV = SingleMV
    else // 4 MV desde los que tomar
    {
        Contar el número de MV del mismo campo y campo opuesto

        if (OppFieldCount > SameFieldCount)
            entonces usar únicamente los MV de campo opuesto en siguiente etapa
            //es decir opuesta es la polaridad elegida
        else usar únicamente el mismo MV de campo en la siguiente etapa
            //es decir la misma es la polaridad elegida

        Contar el número de MV de la polaridad elegida

        if ( MV elegidos   = 3) {
            SelectedMV = Mediana de 3 de los MV elegidos
        }
        else if ( MV elegidos   = 2) {
            SelectedMV = Promedio de los MV elegidos
        }
        else if ( MV elegidos   = 1) {
            SelectedMV = El MV elegido
        }
        else { // todos los 4 son de la polaridad elegida
            SelectedMV = Mediana de 4 de los MV elegidos
        }
    }

    return (SelectedMV);
}

```

Figura 54

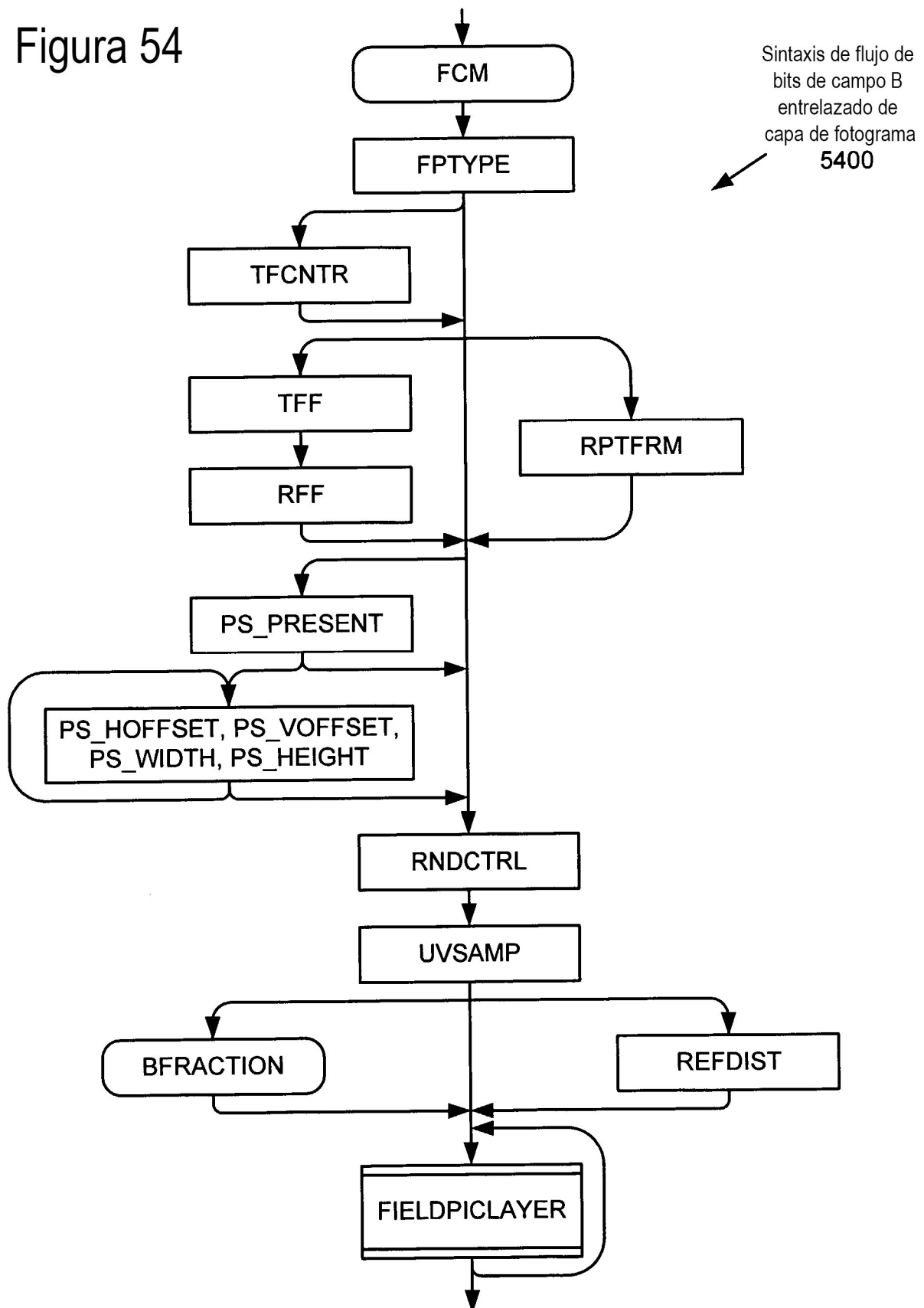


Figura 55

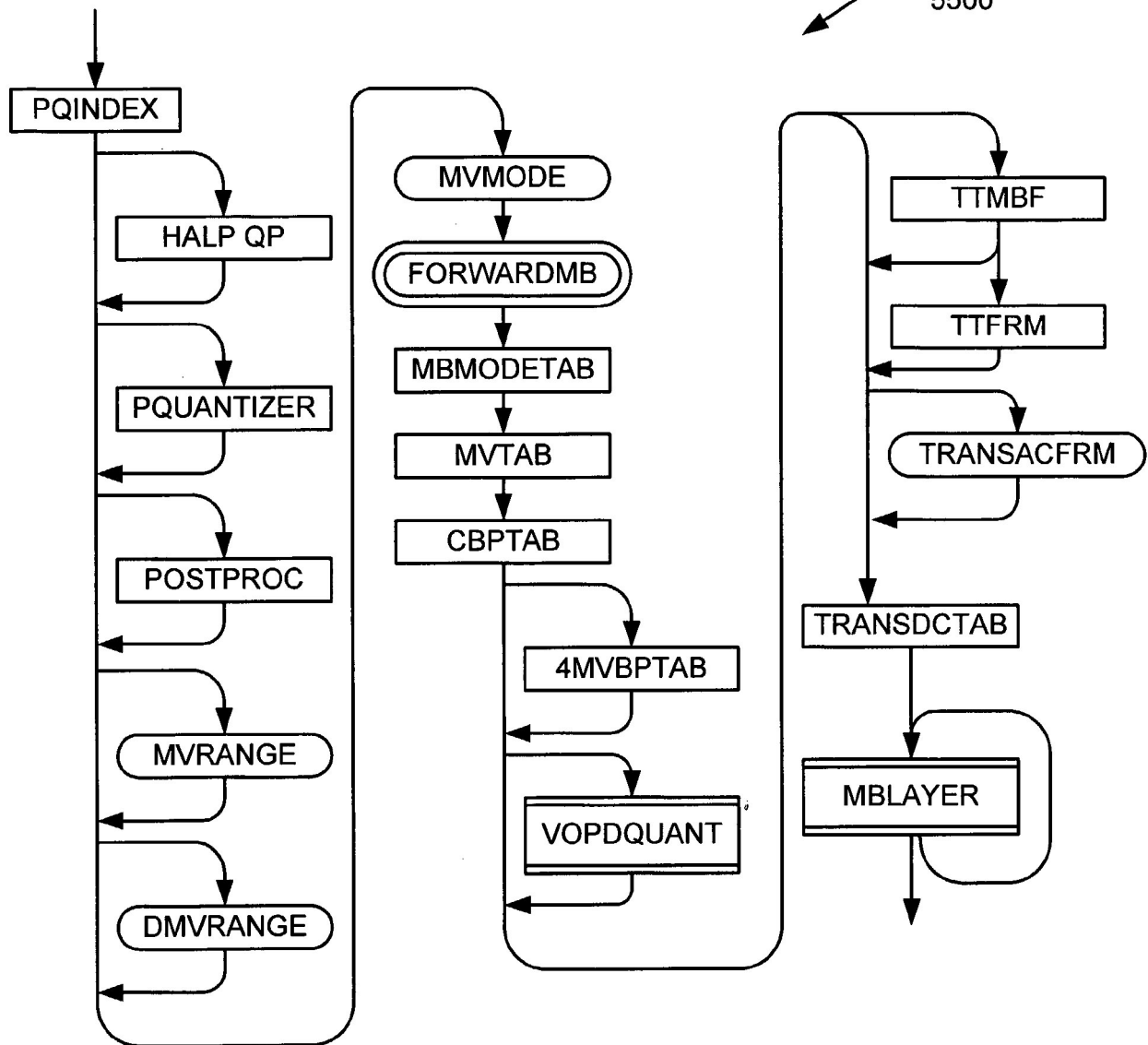


Figura 56

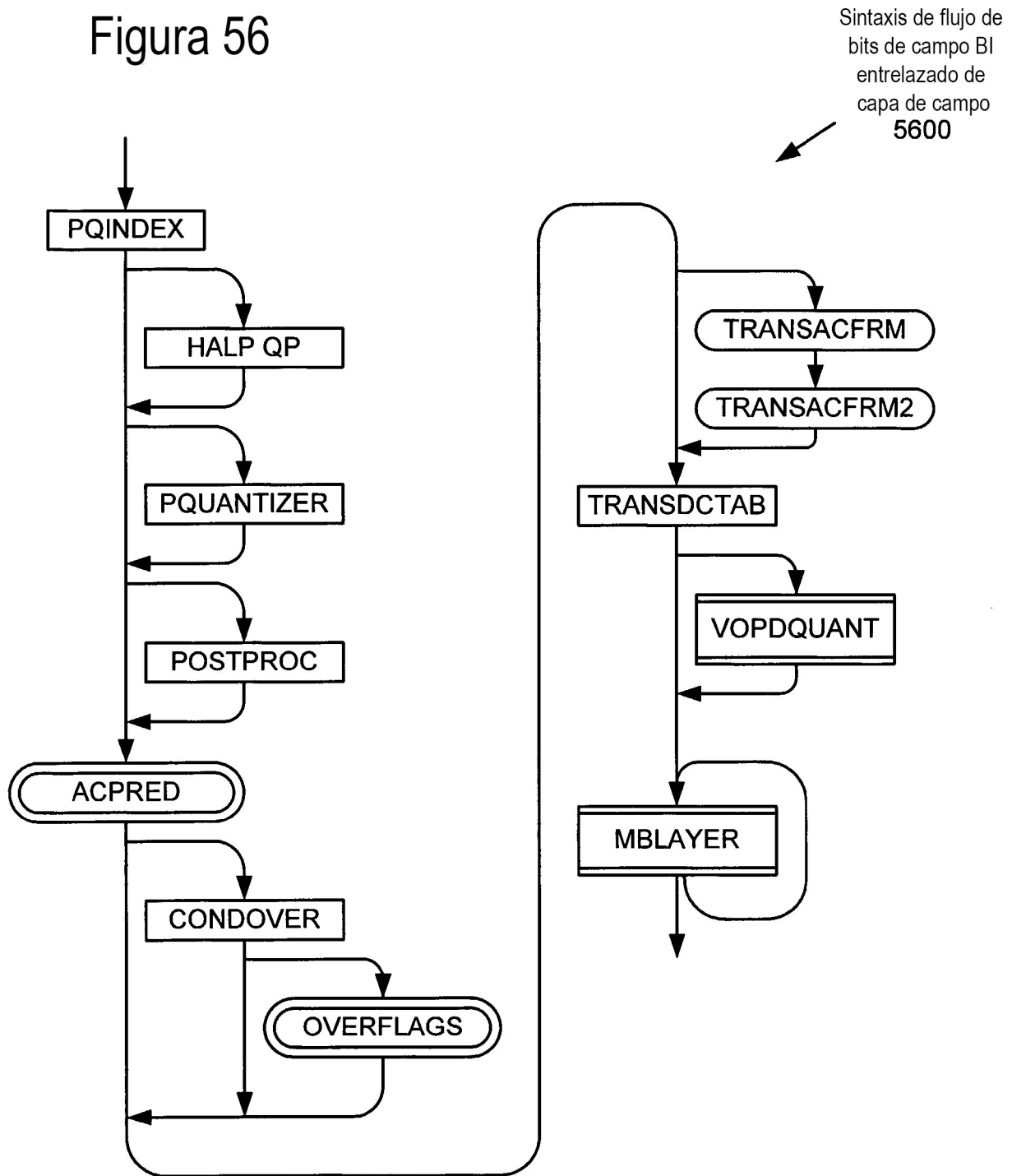


Figura 57

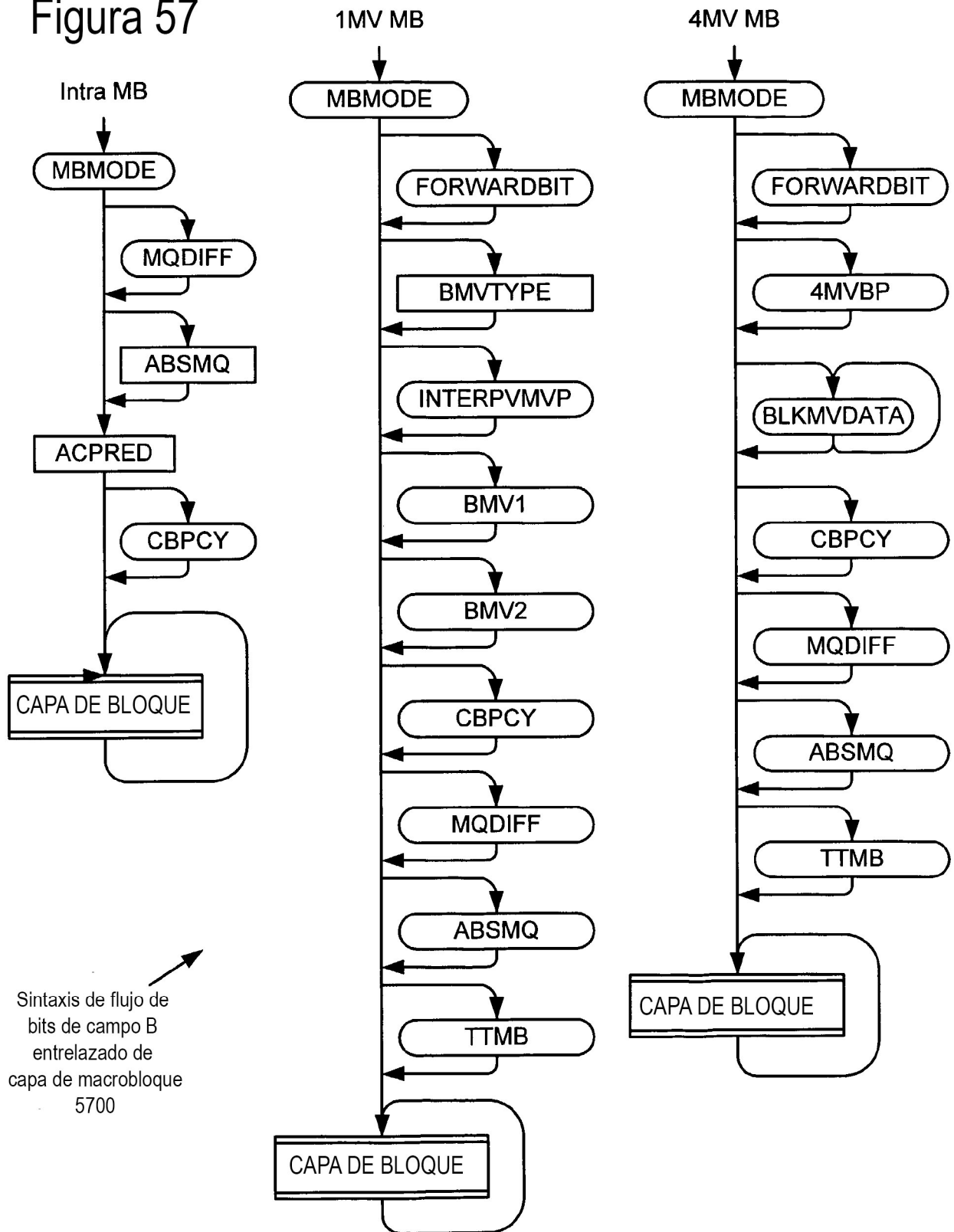


Figura 58

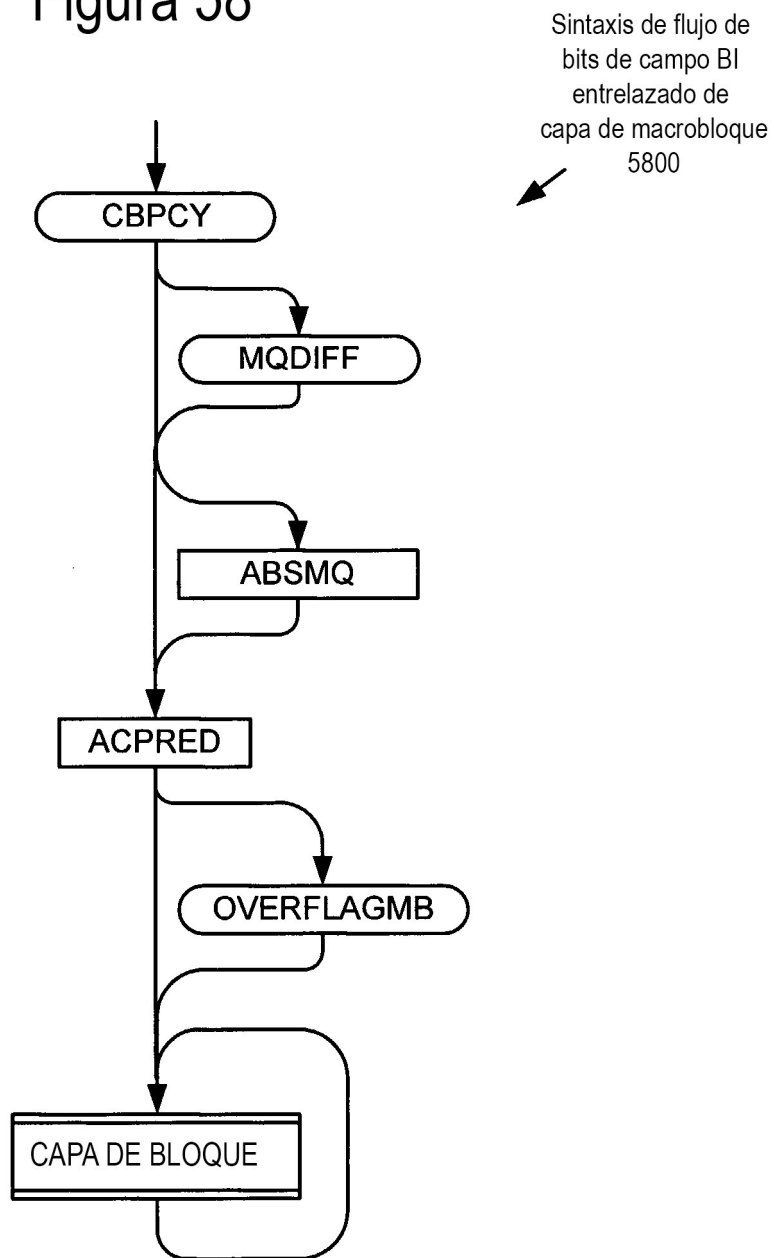


Figura 59

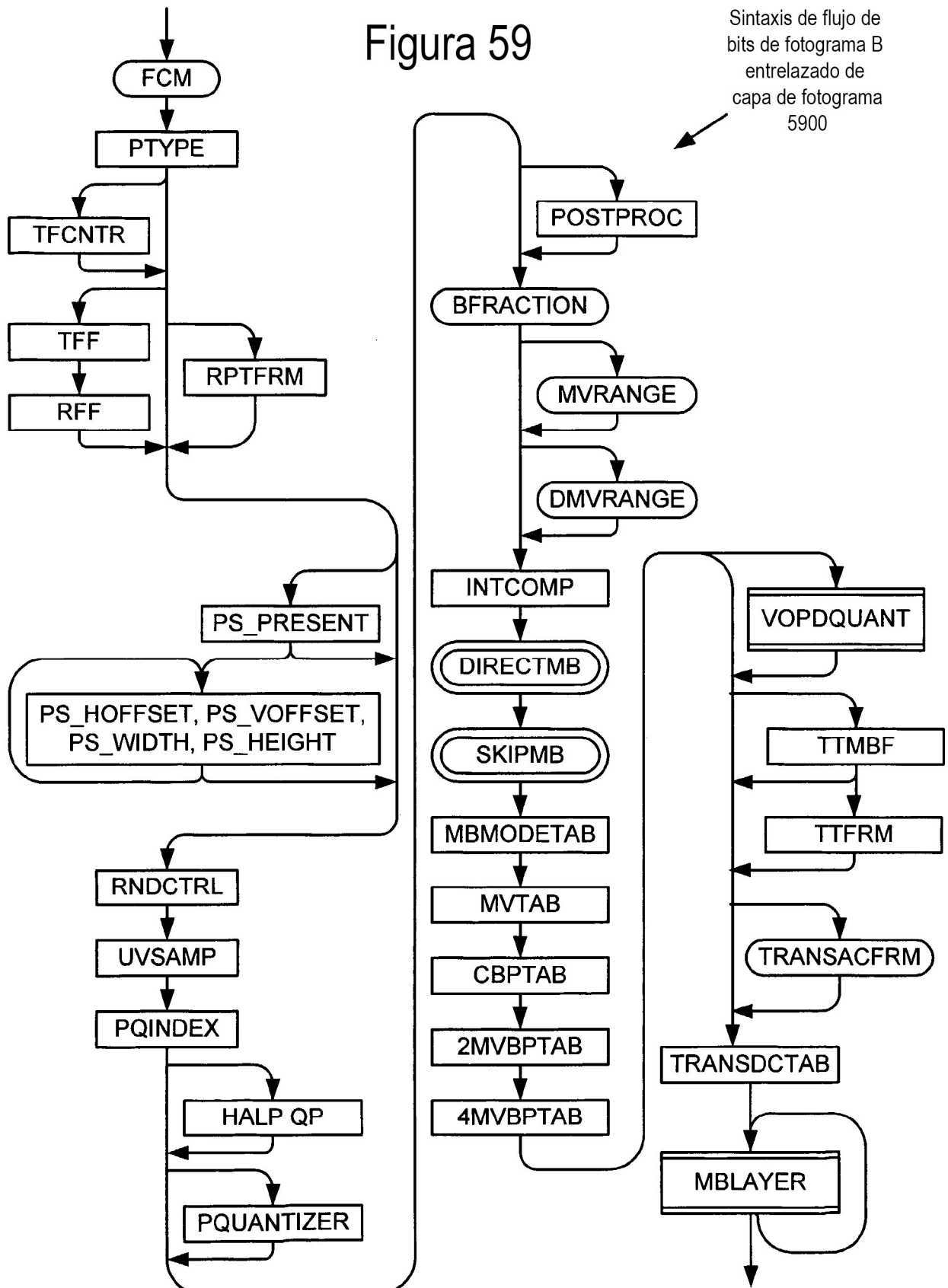
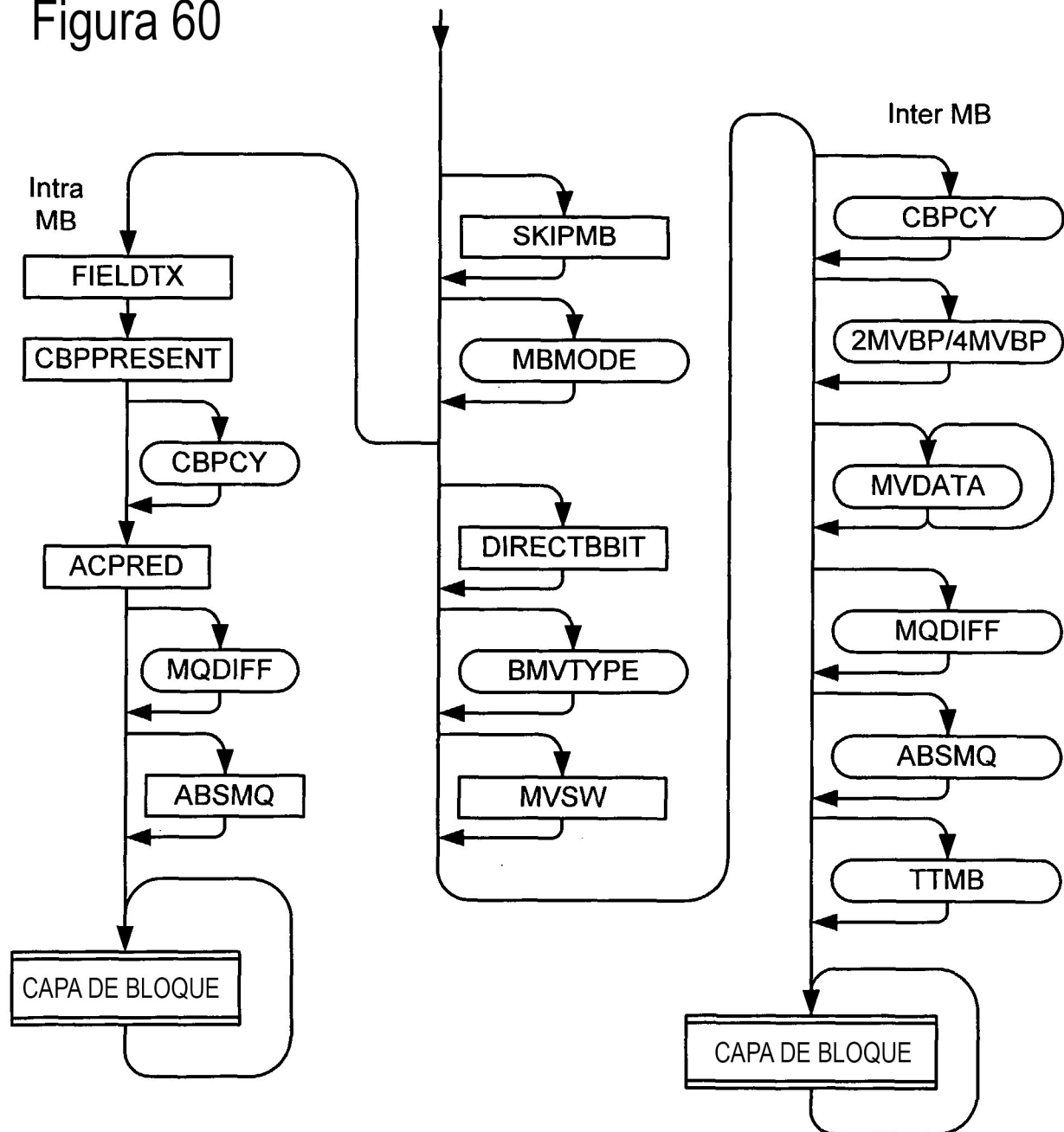


Figura 60



Sintaxis de flujo de
bits de fotograma B
entrelazado de
capa de macrobloque
6000

Figura 61A

6100

```

size_table[16] = {0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7}
offset_table1[9] = {0, 1, 2, 4, 8, 16, 32, 64, 128}
offset_table2[9] = {0, 1, 3, 7, 15, 31, 63, 127, 255}
index = vlc_decode() // Usar la tabla de Huffman indicada por MVTAB
if (index == 125)
{
    dmv_x = get_bits(k_x)
    dmv_y = get_bits(k_y)
    predictor_flag = dmv_y & 1
    dmv_y = dmv_y >> 1
}
else
{
    if (extend_x == 1)
        offset_table = offset_table2
    else
        offset_table = offset_table1
    index1 = (index + 1) % 9
    if (index1 != 0)
    {
        val = get_bits (index1 + extend_x)
        sign = 0 - (val & 1)
        dmv_x = sign ^ ((val >> 1) + offset_table[index1])
        dmv_x = dmv_x - sign
    }
    else
        dmv_x = 0
    if (extend_y == 1)
        offset_table = offset_table2
    else
        offset_table = offset_table1
    index1 = (index + 1) / 9
    if (index1 != 0)
    {
        val = get_bits (size_table[index1 + 2 * extend_y])
        sign = 0 - (val & 1)
        dmv_y = sign ^ ((val >> 1) + offset_table[index1 >> 1])
        dmv_y = dmv_y - sign
        predictor_flag = index1 & 1
    }
    else
    {
        dmv_y = 0
        predictor_flag = 0
    }
}
}

```

Figura 61B

6110

```

size_table[14] = {0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6}
offset_table[9] = {0, 1, 2, 4, 8, 16, 32, 64, 128}
index = vlc_decode() // Usar la tabla de Huffman indicada por MVTAB en
la capa de instantánea
if (index == 0) {
    dmv_x = 1 - 2 * get_bits(1)
    dmv_y = 0
    predictor_flag = 0
}
if (index == 125)
{
    dmv_x = get_bits(k_x - halfpel_flag)
    dmv_y = get_bits(k_y - halfpel_flag)
    predictor_flag = dmv_y & 1
    dmv_y = dmv_y >> 1
}
else
{
    index1 = (index + 1) % 9
    val = get_bits (index1)
    sign = 0 - (val & 1)
    dmv_x = sign ^ ((val >> 1) + offset_table[index1])
    dmv_x = dmv_x - sign

    index1 = (index + 1) / 9
    val = get_bits (size_table[index1])
    sign = 0 - (val & 1)
    dmv_y = sign ^ ((val >> 1) + offset_table[index1])
    dmv_y = dmv_y - sign
    predictor_flag = index1 & 1
}

```

Figura 62A

6200

```

samecount = 0;
oppositecount = 0;
if (predictorA no está fuera de los límites) {
  if (predictorC no está fuera de los límites) {
    if (predictorA es intra) {
      samefieldpred_x = oppositefieldpred_x = samefieldpredA_x = oppositefieldpredA_x = 0
      samefieldpred_y = oppositefieldpred_y = samefieldpredA_y = oppositefieldpredA_y = 0
    }
    if (predictorB es intra) {
      samefieldpred_x = oppositefieldpred_x = samefieldpredB_x = oppositefieldpredB_x = 0
      samefieldpred_y = oppositefieldpred_y = samefieldpredB_y = oppositefieldpredB_y = 0
    }
    if (predictorC es intra) {
      samefieldpred_x = oppositefieldpred_x = samefieldpredC_x = oppositefieldpredC_x = 0
      samefieldpred_y = oppositefieldpred_y = samefieldpredC_y = oppositefieldpredC_y = 0
    }
    if (predictorA no es intra) {
      if (predictorA es desde mismo campo) {
        samecount = samecount + 1
        samefieldpred_x = samefieldpredA_x = predictorA_x
        samefieldpred_y = samefieldpredA_y = predictorA_y
        oppositefieldpred_x = oppositefieldpredA_x = scaleforopposite_x(predictorA_x)
        oppositefieldpred_y = oppositefieldpredA_y = scaleforopposite_y(predictorA_y)
      }
      else {
        oppositecount = oppositecount + 1
        oppositefieldpred_x = oppositefieldpredA_x = predictorA_x
        oppositefieldpred_y = oppositefieldpredA_y = predictorA_y
        samefieldpred_x = samefieldpredA_x = scaleforsame_x(predictorA_x)
        samefieldpred_y = samefieldpredA_y = scaleforsame_y(predictorA_y)
      }
    }
  }
  if (predictorB no es intra) {
    if (predictorB es desde mismo campo) {
      samecount = samecount + 1
      samefieldpred_x = samefieldpredB_x = predictorB_x
      samefieldpred_y = samefieldpredB_y = predictorB_y
      oppositefieldpred_x = oppositefieldpredB_x = scaleforopposite_x(predictorB_x)
      oppositefieldpred_y = oppositefieldpredB_y = scaleforopposite_y(predictorB_y)
    }
    else {
      oppositecount = oppositecount + 1
      oppositefieldpred_x = oppositefieldpredB_x = predictorB_x
      oppositefieldpred_y = oppositefieldpredB_y = predictorB_y
      samefieldpred_x = samefieldpredB_x = scaleforsame_x(predictorB_x)
      samefieldpred_y = samefieldpredB_y = scaleforsame_y(predictorB_y)
    }
  }
}

```

Figura 62B

6200

```

if (predictorC no es intra) {
  if (predictorC es desde mismo campo ) {
    samecount = samecount + 1
    samefieldpred_x = samefieldpredC_x = predictorC_x
    samefieldpred_y = samefieldpredC_y = predictorC_y
    oppositefieldpred_x = oppositefieldpredC_x = scaleforopposite_x(predictorC_x)
    oppositefieldpred_y = oppositefieldpredC_y = scaleforopposite_y(predictorC_y)
  }
  else {
    oppositecount = oppositecount + 1
    oppositefieldpred_x = oppositefieldpredC_x = predictorC_x
    oppositefieldpred_y = oppositefieldpredC_y = predictorC_y
    samefieldpred_x = samefieldpredC_x = scaleforsame_x(predictorC_x)
    samefieldpred_y = samefieldpredC_y = scaleforsame_y(predictorC_y)
  }
}
if ((samecount + oppositecount) > 1) {
  samefieldpred_x =
    median (samefieldpredA_x, samefieldpredB_x, samefieldpredC_x)
  samefieldpred_y =
    median (samefieldpredA_y, samefieldpredB_y, samefieldpredC_y)
  oppositefieldpred_x =
    median (oppositefieldpredA_x, oppositefieldpredB_x, oppositefieldpredC_x)
  oppositefieldpred_y =
    median (oppositefieldpredA_y, oppositefieldpredB_y, oppositefieldpredC_y)
}

if (samecount > oppositecount)
  dominantpredictor = samefield
else
  dominantpredictor = oppositefield
}

```

Figura 62C

6200

```

else {
    // predictorC está fuera de los límites
    if (únicamente 1 macrobloque por fila) {
        if (predictorA es intra) {
            samefieldpred_x = oppositefieldpred_x = 0
            samefieldpred_y = oppositefieldpred_y = 0
            dominantpredictor = oppositefield
        }
        else {
            // Usar predictorA
            if (predictorA es desde mismo campo) {
                samefieldpred_x = predictorA_x
                samefieldpred_y = predictorA_y
                oppositefieldpred_x = scaleforopposite_x(predictorA_x)
                oppositefieldpred_y = scaleforopposite_y(predictorA_y)
                dominantpredictor = samefield
            }
            else {
                oppositefieldpred_x = predictorA_x
                oppositefieldpred_y = predictorA_y
                samefieldpred_x = scaleforsame_x(predictorA_x)
                samefieldpred_y = scaleforsame_y(predictorA_y)
                dominantpredictor = oppositefield
            }
        }
    }
}

```


Figura 62D

6200

```

else {
    // Predictor C está fuera de los límites, usar Predictor y PredictorB
    predictorC_x = 0
    predictorC_y = 0
    if (predictorA es intra) {
        samefieldpred_x = oppositefieldpred_x = samefieldpredA_x = oppositefieldpredA_x = 0
        samefieldpred_y = oppositefieldpred_y = samefieldpredA_y = oppositefieldpredA_y = 0
    }
    if (predictorB es intra) {
        samefieldpred_x = oppositefieldpred_x = samefieldpredB_x = oppositefieldpredB_x = 0
        samefieldpred_y = oppositefieldpred_y = samefieldpredB_y = oppositefieldpredB_y = 0
    }
    if (predictorC es intra) {
        samefieldpred_x = oppositefieldpred_x = samefieldpredC_x = oppositefieldpredC_x = 0
        samefieldpred_y = oppositefieldpred_y = samefieldpredC_y = oppositefieldpredC_y = 0
    }
    if (predictorA no es intra) {
        if (predictorA es desde mismo campo) {
            samecount = samecount + 1
            samefieldpred_x = samefieldpredA_x = predictorA_x
            samefieldpred_y = samefieldpredA_y = predictorA_y
            oppositefieldpred_x = oppositefieldpredA_x = scaleforopposite_x(predictorA_x)
            oppositefieldpred_y = oppositefieldpredA_y = scaleforopposite_y(predictorA_y)
        }
        else {
            oppositecount = oppositecount + 1
            oppositefieldpred_x = oppositefieldpredA_x = predictorA_x
            oppositefieldpred_y = oppositefieldpredA_y = predictorA_y
            samefieldpred_x = samefieldpredA_x = scaleforsame_x(predictorA_x)
            samefieldpred_y = samefieldpredA_y = scaleforsame_y(predictorA_y)
        }
    }
}

```

Figura 62E

6200

```

if (predictorB no es intra) {
  if (predictorB es desde mismo campo) {
    samecount = samecount + 1
    samefieldpred_x = samefieldpredB_x = predictorB_x
    samefieldpred_y = samefieldpredB_y = predictorB_y
    oppositefieldpred_x = oppositefieldpredB_x = scaleforopposite_x(predictorB_x)
    oppositefieldpred_y = oppositefieldpredB_y = scaleforopposite_y(predictorB_y)
  }
  else {
    oppositecount = oppositecount + 1
    oppositefieldpred_x = oppositefieldpredB_x = predictorB_x
    oppositefieldpred_y = oppositefieldpredB_y = predictorB_y
    samefieldpred_x = samefieldpredB_x = scaleforsame_x(predictorB_x)
    samefieldpred_y = samefieldpredB_y = scaleforsame_y(predictorB_y)
  }
}
if ((samecount + oppositecount) > 1) {
  samefieldpred_x =
    median (samefieldpredA_x, samefieldpredB_x, samefieldpredC_x)
  samefieldpred_y =
    median (samefieldpredA_y, samefieldpredB_y, samefieldpredC_y)
  oppositefieldpred_x =
    median (oppositefieldpredA_x, oppositefieldpredB_x, oppositefieldpredC_x)
  oppositefieldpred_y =
    median (oppositefieldpredA_y, oppositefieldpredB_y, oppositefieldpredC_y)
}
if (samecount > oppositecount)
  dominantpredictor = samefield
else
  dominantpredictor = oppositfield
}
}
}

```

Figura 62F

6200

```

else {
    // Predictor A está fuera de los límites
    if (predictorC está fuera de los límites) {
        samefieldpred_x = oppositefieldpred_x = 0
        samefieldpred_y = oppositefieldpred_y = 0
        dominantpredictor = oppositefield
    }
    else {
        // Usar predictorC
        if (predictorC es desde el mismo campo) {
            samefieldpred_x = predictorC_x
            samefieldpred_y = predictorC_y
            oppositefieldpred_x = scaleforopposite_x(predictorC_x)
            oppositefieldpred_y = scaleforopposite_y(predictorC_y)
            dominantpredictor = samefield
        }
        else {
            oppositefieldpred_x = predictorC_x
            oppositefieldpred_y = predictorC_y
            samefieldpred_x = scaleforsame_x(predictorC_x)
            samefieldpred_y = scaleforsame_y(predictorC_y)
            dominantpredictor = oppositefield
        }
    }
}

```

Figura 63

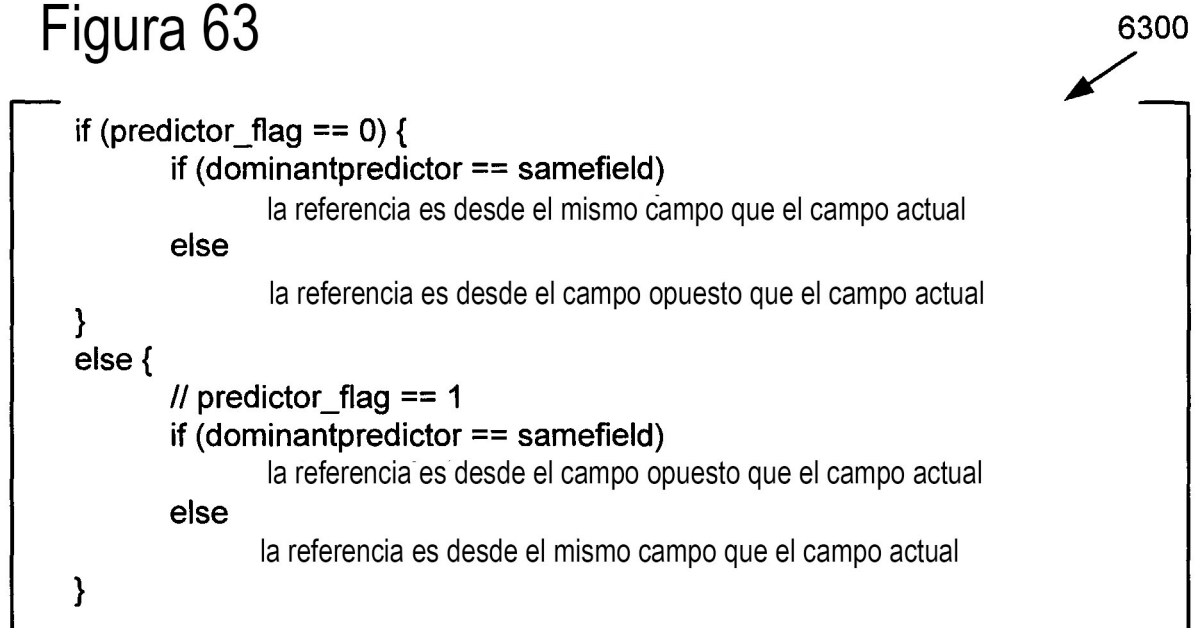


Figura 64

6400

```

if (A existe y A no está intra codificado) {
  if (A es 1 MV) {
    Añadir MV de A al conjunto de MV candidatos
  } else if (A es MV de 4 fotogramas) {
    Añadir el MV de bloque superior derecho de A al conjunto de MV candidatos
  } else if (A es MV de 2 campos) {
    Promediar los MV de dos campos de A y añadir el MV resultante al conjunto de MV candidatos
  } else if (A es MV de 4 campos) {
    Promediar el MV de campo de bloque superior derecho y MV de campo de bloque
    inferior derecho de A y añadir el MV resultante al conjunto de MV candidatos
  }
}

if (B existe y B no está intra codificado) {
  if (B es 1 MV) {
    Añadir MV de B al conjunto de MV candidatos
  } else if (B es MV de 4 fotogramas) {
    Añadir el MV de bloque superior izquierdo de B al conjunto de MV candidatos
  } else if (B es MV de 2 campos) {
    Promediar los MV de dos campos de B y añadir el MV resultante al conjunto de MV candidatos
  } else if (B es MV de 4 campos) {
    Promediar el MV de campo de bloque superior izquierdo y MV de campo de bloque
    inferior izquierdo de B y añadir el MV resultante al conjunto de MV candidatos
  }
}

if (C existe y C no está intra codificado) {
  if (C es 1 MV) {
    Añadir MV de C al conjunto de MV candidatos
  } else if (C es MV de 4 fotogramas) {
    if (C es el MB superior derecho) {
      Añadir el MV de bloque inferior izquierdo de C al conjunto de MV candidatos
    } else { //C es el MB superior izquierdo
      Añadir el MV de bloque inferior derecho de C al conjunto de MV candidatos
    }
  } else if (C es MV de 2 campos) {
    Promediar los MV de dos campos de C y añadir el MV resultante al conjunto de MV candidatos
  } else if (C es MV de 4 campos) {
    if (C es el MB superior derecho) {
      Promediar el MV de campo de bloque superior izquierdo y el MV de campo de bloque
      inferior izquierdo de C y añadir el MV resultante al conjunto de MV candidatos
    } else { //C es el MB superior izquierdo
      Promediar el MV de campo de bloque superior derecho y el MV de campo de bloque
      inferior derecho de C y añadir el MV resultante al conjunto de MV candidatos
    }
  }
}
}

```

Figura 65

6500

```

//MV de bloque superior izquierdo
if (A existe y A no está intra codificado) {
  if (A es 1 MV) {
    Añadir MV de A al conjunto de MV candidatos
  } else if (A es MV de 4 fotogramas) {
    Añadir el MV de bloque superior derecho de A al conjunto de MV candidatos
  } else if (A es MV de 2 campos) {
    Promediar los MV de dos campos de A y añadir el MV resultante al conjunto de MV candidatos
  } else if (A es MV de 4 campos) {
    Promediar el MV de campo de bloque superior derecho y MV de campo de bloque
    inferior derecho de A y añadir el MV resultante al conjunto de MV candidatos
  }
}
if (B existe y B no está intra codificado) {
  if (B es 1 MV) {
    Añadir MV de B al conjunto de MV candidatos
  } else if (B es MV de 4 fotogramas) {
    Añadir el MV de bloque inferior izquierdo de B al conjunto de MV candidatos
  } else if (B es MV de 2 campos) {
    Promediar los MV de dos campos de B y añadir el MV resultante al conjunto de MV candidatos
  } else if (B es MV de 4 campos) {
    Promediar el MV de campo de bloque superior izquierdo y MV de campo de bloque
    inferior izquierdo de B y añadir el MV resultante al conjunto de MV candidatos
  }
}
if (C existe y C no está intra codificado) {
  if (C es 1 MV) {
    Añadir MV de C al conjunto de MV candidatos
  } else if (C es MV de 4 fotogramas) {
    if (C es el MB superior derecho) {
      Añadir el MV de bloque inferior izquierdo de C al conjunto de MV candidatos
    } else { //C es el MB superior izquierdo
      Añadir el MV de bloque inferior derecho de C al conjunto de MV candidatos
    }
  } else if (C es MV de 2 campos) {
    Promediar los MV de dos campos de C y añadir el MV resultante al conjunto de MV candidatos
  } else if (C es MV de 4 campos) {
    if (C es el MB superior derecho) {
      Promediar el MV de campo de bloque superior izquierdo y el MV de campo de bloque
      inferior izquierdo de C y añadir el MV resultante al conjunto de MV candidatos
    } else { //C es el MB superior izquierdo
      Promediar el MV de campo de bloque superior derecho y el MV de campo de bloque
      inferior derecho de C y añadir el MV resultante al conjunto de MV candidatos
    }
  }
}
}

```

Figura 66

6600

```

//MV de bloque superior derecho
Añadir el MV de bloque superior izquierdo del MB actual al conjunto de MV candidatos

if (B existe y B no está intra codificado) {
  if (B es 1 MV) {
    Añadir MV de B al conjunto de MV candidatos
  } else if (B es MV de 4 fotogramas){
    Añadir el MV de bloque inferior derecho de B al conjunto de MV candidatos
  } else if (B es MV de 2 campos) {
    Promediar los MV de dos campos de B y añadir el MV resultante al conjunto de MV candidatos
  } else if (B es MV de 4 campos){
    Promediar el MV de campo de bloque superior derecho y MV de campo de bloque
    inferior derecho de B y añadir el MV resultante al conjunto de MV candidatos
  }
}

if (C existe y C no está intra codificado) {
  if (C es 1 MV) {
    Añadir MV de C al conjunto de MV candidatos
  } else if (C es MV de 4 fotogramas) {
    if (C es el MB superior derecho){
      Añadir el MV de bloque inferior izquierdo de C al conjunto de MV candidatos
    } else { //C es el MB superior izquierdo
      Añadir el MV de bloque inferior derecho de C al conjunto de MV candidatos
    }
  } else if (C es MV de 2 campos) {
    Promediar los MV de dos campos de C y añadir el MV resultante al conjunto de MV candidatos
  } else if (C es MV de 4 campos) {
    if (C es el MB superior derecho){
      Promediar el MV de campo de bloque superior izquierdo y el MV de campo de bloque
      inferior izquierdo de C y añadir el MV resultante al conjunto de MV candidatos
    } else { //C es el MB superior izquierdo
      Promediar el MV de campo de bloque superior derecho y el MV de campo de bloque
      inferior derecho de C y añadir el MV resultante al conjunto de MV candidatos
    }
  }
}

```

Figura 67

6700

```

// MV de bloque inferior izquierdo
if (A existe y A no está intra codificado) {
  if (A es 1 MV) {
    Añadir MV de A al conjunto de MV candidatos.
  } else if (A es MV de 4 fotogramas) {
    Añadir el MV de bloque inferior derecho de A al conjunto de MV candidatos.
  } else if (A es MV de 2 campos) {
    Promediar los MV de dos campos de A y añadir el MV resultante al
    conjunto de MV candidatos.
  } else if (A es MV de 4 campos) {
    Promediar el MV de campo de bloque superior derecho y MV de campo de
    bloque inferior derecho de A y añadir el MV resultante al conjunto de MV candidatos.
  }
}

Añadir el MV de bloque superior izquierdo del MB actual al conjunto de MV candidatos

Añadir el MV de bloque superior derecho del MB actual al conjunto de MV candidatos

```

Figura 68

6800

```

// MV de bloque inferior derecho
Añadir el MV de bloque inferior izquierdo del MB actual al conjunto de MV candidatos

Añadir el MV de bloque superior izquierdo del MB actual al conjunto de MV candidatos

Añadir el MV de bloque superior derecho del MB actual al conjunto de MV candidatos

```


Figura 69

6900

```

//MV de campo superior
if (A existe y A no está intra codificado)  {
    if (A es 1 MV) {
        Añadir MV de A al conjunto de MV candidatos
    } else if (A es MV de 4 fotogramas) {
        Añadir el MV de bloque superior derecho de A al conjunto de MV candidatos
    } else if (A es MV de 2 campos) {
        Añadir el MV de campo superior de A al conjunto de MV candidatos
    } else if (A es MV de 4 campos) {
        Añadir el MV de bloque de campo superior derecho de A al conjunto de MV candidatos
    }
}
if (B existe y B no está intra codificado)  {
    if (B es 1 MV) {
        Añadir MV de B al conjunto de MV candidatos
    } else if (B es MV de 4 fotogramas) {
        Añadir el MV de bloque inferior izquierdo de B al conjunto de MV candidatos
    } else if (B es MV de 2 campos) {
        Añadir el MV de campo superior de B al conjunto de MV candidatos
    } else if (B es MV de 4 campos) {
        Añadir el MV de bloque de campo superior izquierdo de B al conjunto de MV candidatos
    }
}
if (C existe y C no está intra codificado)  {
    if (C es 1 MV) {
        Añadir MV de C al conjunto de MV candidatos
    } else if (C es MV de 4 fotogramas) {
        if (C es el MB superior derecho){
            Añadir el MV de bloque inferior izquierdo de C al conjunto de MV candidatos
        } else { //C es el MB superior izquierdo
            Añadir el MV de bloque inferior derecho de C al conjunto de MV candidatos
        }
    } else if (C es MV de 2 campos) {
        Añadir el MV de campo superior de C al conjunto de MV candidatos
    } else if (C es MV de 4 campos) {
        if (C es el MB superior derecho){
            Añadir el MV de bloque de campo superior izquierdo de C al conjunto de MV candidatos
        } else { //C es el MB superior derecho
            Añadir el MV de bloque de campo superior derecho de C al conjunto de MV candidatos
        }
    }
}
}

```

Figura 70

7000

```

//MV de campo inferior
if (A existe y A no está intra codificado) {
    if (A es 1 MV) {
        Añadir MV de A al conjunto de MV candidatos
    } else if (A es MV de 4 fotogramas) {
        Añadir el MV de bloque inferior derecho de A al conjunto de MV candidatos
    } else if (A es MV de 2 campos) {
        Añadir el MV de campo inferior de A al conjunto de MV candidatos
    } else if (A es MV de 4 campos) {
        Añadir el MV de bloque de campo inferior derecho de A al conjunto de MV candidatos
    }
}
if (B existe y B no está intra codificado) {
    if (B es 1 MV) {
        Añadir MV de B al conjunto de MV candidatos
    } else if (B es MV de 4 fotogramas) {
        Añadir el MV de bloque inferior izquierdo de B al conjunto de MV candidatos
    } else if (B es MV de 2 campos) {
        Añadir el MV de campo inferior de B al conjunto de MV candidatos
    } else if (B es MV de 4 campos) {
        Añadir el MV de bloque de campo inferior izquierdo de B al conjunto de MV candidatos
    }
}
if (C existe y C no está intra codificado) {
    if (C es 1 MV) {
        Añadir MV de C al conjunto de MV candidatos
    } else if (C es MV de 4 fotogramas) {
        if (C es el MB superior derecho) {
            Añadir el MV de bloque inferior izquierdo de C al conjunto de MV candidatos
        } else { //C es el MB superior izquierdo
            Añadir el MV de bloque inferior derecho de C al conjunto de MV candidatos
        }
    } else if (C es MV de 2 campos) {
        Añadir el MV de campo inferior de C al conjunto de MV candidatos
    } else if (C es MV de 4 campos) {
        if (C es el MB superior derecho) {
            Añadir el MV de bloque de campo inferior izquierdo de C al conjunto de MV candidatos
        } else { //C es el MB superior izquierdo
            Añadir el MV de bloque de campo inferior derecho de C al conjunto de MV candidatos
        }
    }
}
}

```

Figura 71

7100

```

//MV de bloque de campo superior izquierdo
if (A existe y A no está intra codificado) {
  if (A es 1 MV) {
    Añadir MV de A al conjunto de MV candidatos
  } else if (A es MV de 4 fotogramas) {
    Añadir el MV de bloque superior derecho de A al conjunto de MV candidatos
  } else if (A es MV de 2 campos) {
    Añadir el MV de campo superior de A al conjunto de MV candidatos
  } else if (A es MV de 4 campos) {
    Añadir el MV de bloque de campo superior derecho de A al conjunto de MV candidatos
  }
}
if (B existe y B no está intra codificado) {
  if (B es 1 MV) {
    Añadir MV de B al conjunto de MV candidatos
  } else if (B es MV de 4 fotogramas) {
    Añadir el MV de bloque inferior izquierdo de B al conjunto de MV candidatos
  } else if (B es MV de 2 campos) {
    Añadir el MV de campo superior de B al conjunto de MV candidatos
  } else if (B es MV de 4 campos) {
    Añadir el MV de bloque de campo superior izquierdo de B al conjunto de MV candidatos
  }
}
if (C existe y C no está intra codificado) {
  if (C es 1 MV) {
    Añadir MV de C al conjunto de MV candidatos
  } else if (C es MV de 4 fotogramas) {
    if (C es el MB superior derecho) {
      Añadir el MV de bloque inferior izquierdo de C al conjunto de MV candidatos
    } else { //C es el MB superior izquierdo
      Añadir el MV de bloque inferior derecho de C al conjunto de MV candidatos
    }
  } else if (C es MV de 2 campos) {
    Añadir el MV de campo superior de C al conjunto de MV candidatos
  } else if (C es MV de 4 campos) {
    if (C es el MB superior derecho) {
      Añadir el MV de bloque de campo superior izquierdo de C al conjunto de MV candidatos
    } else { //C es el MB superior izquierdo
      Añadir el MV de bloque de campo superior derecho de C al conjunto de MV candidatos
    }
  }
}
}

```

Figura 72

7200

```

//MV de bloque de campo superior derecho
Añadir el MV de bloque de campo superior izquierdo del MB actual al conjunto de MV candidatos

if (B existe y B no está intra codificado) {
    if (B es 1 MV) {
        Añadir MV de B al conjunto de MV candidatos
    } else if (B es MV de 4 fotogramas) {
        Añadir el MV de bloque inferior derecho de B al conjunto de MV candidatos
    } else if (B es MV de 2 campos) {
        Añadir el MV de campo superior de B al conjunto de MV candidatos
    } else if (B es MV de 4 campos) {
        Añadir el MV de bloque de campo superior derecha de B al conjunto de MV candidatos
    }
}

if (C existe y C no está intra codificado) {
    if (C es 1 MV) {
        Añadir MV de C al conjunto de MV candidatos
    } else if (C es MV de 4 fotogramas) {
        if (C es el MB superior derecho) {
            Añadir el MV de bloque inferior izquierdo de C al conjunto de MV candidatos
        } else { //C es el MB superior izquierdo
            Añadir el MV de bloque inferior derecho de C al conjunto de MV candidatos
        }
    } else if (C es MV de 2 campos) {
        Añadir el MV de campo superior de C al conjunto de MV candidatos
    } else if (C es MV de 4 campos) {
        if (C es el MB superior derecho) {
            Añadir el MV de bloque de campo superior izquierdo de C al conjunto de MV candidatos
        } else { //C es el MB superior izquierdo
            Añadir el MV de bloque de campo superior derecho de C al conjunto de MV candidatos
        }
    }
}

```

Figura 73

7300

```

//MV de bloque de campo inferior izquierdo
if (A existe y A no está intra codificado) {
    if (A es 1 MV) {
        Añadir MV de A al conjunto de MV candidatos
    } else if (A es MV de 4 fotogramas) {
        Añadir el MV de bloque inferior derecho de A al conjunto de MV candidatos
    } else if (A es MV de 2 campos) {
        Añadir el MV de campo inferior de A al conjunto de MV candidatos
    } else if (A es MV de 4 campos) {
        Añadir el MV de bloque de campo inferior derecho de A al conjunto de MV candidatos
    }
}
if (B existe y B no está intra codificado) {
    if (B es 1 MV) {
        Añadir MV de B al conjunto de MV candidatos
    } else if (B es MV de 4 fotogramas) {
        Añadir el MV de bloque inferior izquierdo de B al conjunto de MV candidatos
    } else if (B es MV de 2 campos) {
        Añadir el MV de campo inferior de B al conjunto de MV candidatos
    } else if (B es MV de 4 campos) {
        Añadir el MV de bloque de campo inferior izquierdo de B al conjunto de MV candidatos
    }
}
if (C existe y C no está intra codificado) {
    if (C es 1 MV) {
        Añadir MV de C al conjunto de MV candidatos
    } else if (C es MV de 4 fotogramas) {
        if (C es el MB superior derecho) {
            Añadir el MV de bloque inferior izquierdo de C al conjunto de MV candidatos
        } else { //C es el MB superior izquierdo
            Añadir el MV de bloque inferior derecho de C al conjunto de MV candidatos
        }
    } else if (C es MV de 2 campos) {
        Añadir el MV de campo inferior de C al conjunto de MV candidatos
    } else if (C es MV de 4 campos) {
        if (C es el MB superior derecho) {
            Añadir el MV de bloque de campo inferior izquierdo de C al conjunto de MV candidatos
        } else { //C es el MB superior izquierdo
            Añadir el MV de bloque de campo inferior derecho de C al conjunto de MV candidatos
        }
    }
}
}

```

Figura 74

7400

```

// MV de bloque de campo inferior derecho
Añadir el MV de bloque de campo inferior izquierdo del MB actual al conjunto de MV candidatos

if (B existe y B no está intra codificado)  {
    if (B es 1 MV) {
        Añadir MV de B al conjunto de MV candidatos.
    } else if (B es MV de 4 fotogramas) {
        Añadir el MV de bloque inferior derecho de B al conjunto de MV candidatos.
    } else if (B es MV de 2 campos) {
        Añadir el MV de campo inferior de B al conjunto de MV candidatos.
    } else if (B es MV de 4 campos) {
        Añadir el MV de bloque de campo inferior derecho de B al conjunto de MV candidatos.
    }
}

if (C existe y C no está intra codificado)
    if (C es 1 MV) {
        Añadir MV de C al conjunto de MV candidatos.
    } else if (C es MV de 4 fotogramas) {
        if (C es el MB superior derecho){
            Añadir el MV de bloque inferior izquierdo de C al conjunto de MV candidatos
        } else { // C es el MB superior izquierdo
            Añadir el MV de bloque inferior derecho de C al conjunto de MV candidatos
        }
    } else if (C es MV de 2 campos){
        Añadir el MV de campo inferior de C al conjunto de MV candidatos
    } else if (C es MV de 4 campos){
        if (C es el MB superior derecho){
            Añadir el MV de bloque de campo inferior izquierdo de C al conjunto de candidatos
        } else { // C es el MB superior izquierdo
            Añadir el MV de bloque de campo inferior derecho de C al conjunto de candidatos
        }
    }
}

```

Figura 75

```

if (TotalValidMV >= 2) {
    // Obsérvese que si hay únicamente dos MV válidos, entonces el
    // tercer ValidMV se establece para que sea (0, 0)
    PMVx = median3 (ValidMVx [0], ValidMVx [1], ValidMVx [2]);
    PMVy = median3 (ValidMVy [0], ValidMVy [1], ValidMVy [2]);
} else if (TotalValidMV es 1) {
    PMVx = ValidMVx [0];
    PMVy = ValidMVy [0];
} else {
    PMVx = 0;
    PMVy = 0;
}

```

7500

Figura 76

```

if (TotalValidMV == 3) {
    if (NumSameFieldMV == 3 || NumOppFieldMV == 3) {
        PMVx = median3 (ValidMVx [0], ValidMVx [1], ValidMVx [2]);
        PMVy = median3 (ValidMVy [0], ValidMVy [1], ValidMVy [2]);
    } else if (NumSameFieldMV >= NumOppFieldMV) {
        PMVx = SameFieldMVx [0];
        PMVy = SameFieldMVy [0];
    } else {
        PMVx = OppFieldMVx [0];
        PMVy = OppFieldMVy [0];
    }
} else if (TotalValidMV == 2) {
    if (NumSameFieldMV >= NumOppFieldMV) {
        PMVx = SameFieldMVx [0];
        PMVy = SameFieldMVy [0];
    } else {
        PMVx = OppFieldMVx [0];
        PMVy = OppFieldMVy [0];
    }
} else if (TotalValidMV == 1) {
    PMVx = ValidMVx [0];
    PMVy = ValidMVy [0];
} else {
    PMVx = 0;
    PMVy = 0;
}

```

7600

Figura 77A

7700


```

offset_table1[9] = {0, 1, 2, 4, 8, 16, 32, 64, 128,}
offset_table2[9] = {0, 1, 3, 7, 15, 31, 63, 127, 255}
index = vlc_decode() // Usar la tabla de Huffman indicada por MVTAB en
la capa de instantánea
if (index == 71)
{
    dmv_x = get_bits(k_x)
    dmv_y = get_bits(k_y)
}
else
{
    if (extend_x == 1)
        offset_table = offset_table2
    else
        offset_table = offset_table1
    index1 = (index + 1) % 9
    if (index1 != 0)
    {
        val = get_bits (index1 + extend_x)
        sign = 0 - (val & 1)
        dmv_x = sign ^ ((val >> 1) + offset_table[index1])
        dmv_x = dmv_x - sign
    }
    else
        dmv_x = 0
    if (extend_y == 1)
        offset_table = offset_table2
    else
        offset_table = offset_table1
    index1 = (index + 1) / 9
    if (index1 != 0)
    {
        val = get_bits (index1 + extend_y)
        sign = 0 - (val & 1)
        dmv_y = sign ^ ((val >> 1) + offset_table[index1])
        dmv_y = dmv_y - sign
    }
    else
        dmv_y = 0
}
}

```


Figura 77B

7710

```

offset_table[9] = {0, 1, 2, 4, 8, 16, 32, 64, 128}
index = vlc_decode()      // Usar la tabla de Huffman indicada por MVTAB en
la capa de instantánea
if (index == 0) {
    dmv_x = 1 - 2 * get_bits(1)
    dmv_y = 0
}
if (index == 125)
{
    dmv_x = get_bits(k_x - halfpel_flag)
    dmv_y = get_bits(k_y - halfpel_flag)
}
else
{
    index1 = (index + 1) % 9
    val = get_bits (index1)
    sign = 0 - (val & 1)
    dmv_x = sign ^ ((val >> 1) + offset_table[index1])
    dmv_x = dmv_x - sign

    index1 = (index + 1) / 9
    val = get_bits (index1)
    sign = 0 - (val & 1)
    dmv_y = sign ^ ((val >> 1) + offset_table[index1])
    dmv_y = dmv_y - sign
}

```

Figura 78

7800


```
Int s_RndTbl [] = {0, 0, 0, 1};  
Int s_RndTblField [] = {0, 0, 1, 2, 4, 4, 5, 6, 2, 2, 3, 8, 6, 6, 7, 12};  
CMVX = (LMVX + s_RndTbl[LMVX & 3]) >> 1;  
if (LMV es un vector de movimiento de campo) {  
    CMVY = (LMVY >> 4)*8 + s_RndTblField [LMVY & 0xF];  
} else {  
    CMVY = (LMVY + s_RndTbl[LMVY & 3]) >> 1;  
}
```

Figura 79A

7900

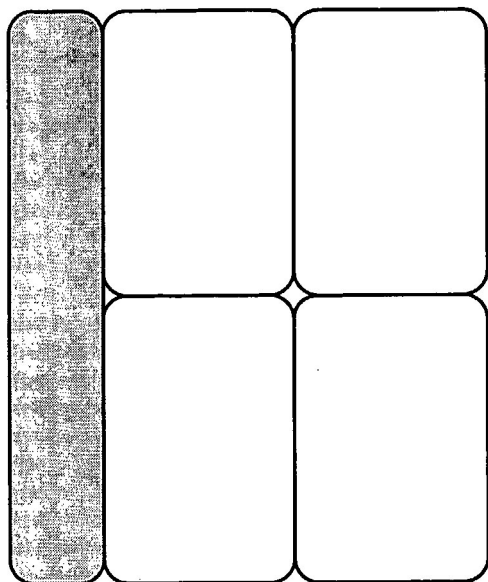


Figura 79B

7901

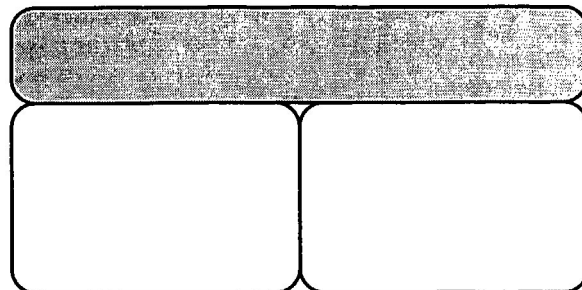


Figura 79C

7902

