

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 016 814**

51 Int. Cl.:

H04N 19/119 (2014.01)
H04N 19/52 (2014.01)
H04N 19/147 (2014.01)
H04N 19/176 (2014.01)
H04N 19/196 (2014.01)
H04N 19/463 (2014.01)
H04N 19/70 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **10.10.2011** **E 20201843 (8)**

97 Fecha y número de publicación de la concesión europea: **01.01.2025** **EP 3833025**

54 Título: **Codificación de imágenes que soporta partición de bloques y fusión de bloques**

30 Prioridad:

08.10.2010 US 39147310 P

45 Fecha de publicación y mención en BOPI de la
traducción de la patente:
09.05.2025

73 Titular/es:

GE VIDEO COMPRESSION, LLC (100.00%)
1 Research Circle
Niskayuna, NY 12309, US

72 Inventor/es:

SCHWARZ, HEIKO;
HELLE, PHILIPP;
MARPE, DETLEV;
WIEGAND, THOMAS;
UDIN, SIMON y
BROSS, BENJAMIN

74 Agente/Representante:

RUÍZ GALLEGOS, Natalia

ES 3 016 814 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 de imágenes que soporta partición de bloques y fusión de bloques

- 5 La presente solicitud se refiere a la codificación de imagen y/o vídeo y, en particular, códecs que soportan partición de bloques y fusión de bloques.

Muchos códecs de imagen y/o vídeo tratan las imágenes en conjuntos de bloques. Por ejemplo, los códecs predictivos usan una granularidad de bloque con el fin de lograr un buen compromiso entre parámetros de predicción establecidos con mucha precisión establecidos a una alta resolución espacial gastando, pero gastando demasiada información conexa para los parámetros de predicción por un lado y parámetros de predicción establecidos demasiado de manera demasiado aproximada, haciendo que la cantidad de bits necesarios para codificar el residuo de predicción aumente debido a la menor resolución espacial de los parámetros de predicción, por otro lado. En efecto, el ajuste óptimo para los parámetros de predicción se encuentra en algún lugar entre ambos extremos.

15 Se han hecho varios intentos con el fin de obtener la solución óptima para el problema descrito anteriormente. Por ejemplo, en lugar de usar una subdivisión regular de una imagen en bloques dispuestos regularmente en filas y columnas, la subdivisión de partición multiárbol busca aumentar la libertad de subdividir una imagen en bloques a una demanda razonable de información de subdivisión. Sin embargo, incluso la subdivisión multiárbol requiere la señalización de una cantidad notable de datos y la libertad de subdividir una imagen está bastante restringida, incluso en caso de usar tal subdivisión multiárbol.

25 Con el fin de permitir una mejor solución de compromiso entre la cantidad de información conexa necesaria con el fin de señalar la subdivisión de imagen por un lado y la libertad de subdividir la imagen por otro lado, se puede usar la fusión de bloques con el fin de aumentar el número de subdivisiones de imagen posibles a una cantidad razonable de datos adicionales necesarios con el fin de señalar la información de fusión. Para los bloques que se fusionan, los parámetros de codificación deben transmitirse dentro del flujo de bits en su totalidad solo una vez, de manera similar como si el grupo fusionado resultante de bloques fuera una porción directamente subdividida de la imagen. Sin embargo, todavía hay una necesidad de lograr mejor eficiencia de codificación, debido a las redundancias restantes recién causadas por la combinación de fusión de bloques y subdivisión de bloques.

35 LI B. ET AL, "Redundancy reduction in Cbf and merge coding", JCTVC-C277, describe una propuesta para el estándar HEVC de acuerdo con la cual la codificación de fusión se restringe en la medida en que una partición de una unidad de codificación en dos unidades de predicción se restringe en la medida en que una primera subpartición en orden de partición puede no ser un candidato de fusión de la segunda unidad de predicción en orden de partición.

40 DE FORNI R. ET AL, "On the Benefits of Leaf Merging in Quad-Tree Motion Models", IMAGE PROCESSING, 2005, describe un procedimiento de fusión de hojas como una extensión de una subdivisión de árbol cuádruple con el fin de codificar el movimiento de un vídeo. Con el fin de evitar redundancias entre la subdivisión de árbol cuádruple por un lado y la fusión de hojas por otro, este artículo propone permitir la fusión solo con hojas vecinas inmediatas de un tamaño mayor, o del mismo tamaño, pero con un padre diferente.

45 Por lo tanto, el objeto de la presente invención es proporcionar un concepto de codificación que tiene una eficiencia de codificación aumentada. Este objeto se soluciona por las reivindicaciones independientes pendientes.

La idea subyacente a la presente invención es que se puede lograr un aumento adicional de la eficiencia de codificación si para un bloque actual de una imagen, para el cual el flujo de bits señala uno de los patrones de partición soportados, se evita una inversión de la partición mediante fusión de bloques. En particular, si el señalado de los patrones de partición soportados especifica una subdivisión del bloque en dos o más bloques adicionales, se realiza una eliminación de determinados candidatos a parámetros de codificación para todos los bloques adicionales, excepto un primer bloque adicional de los bloques adicionales en un orden de codificación. En particular, esos candidatos a parámetros de codificación se eliminan del conjunto de candidatos a parámetros de codificación para el bloque adicional respectivo, cuyos parámetros de codificación son los mismos que los parámetros de codificación asociados con cualquiera de los bloques adicionales que, al fusionarse con el bloque adicional respectivo, darían como resultado uno del patrón de partición soportado. Mediante esta medida, se evita la redundancia entre codificación de partición y codificación de fusión y la tara de señalización para señalar la información de fusión puede reducirse adicionalmente explotando el tamaño reducido del conjunto de candidatos a parámetros de codificación. Además, se mantienen los efectos positivos de combinar la partición de bloques con la fusión de bloques. Es decir, debido a la combinación de la partición de bloques con la fusión de bloques, la variedad de patrones de partición alcanzables aumenta con respecto al caso sin fusión de bloques. El aumento de la tara de señalización se mantiene en límites razonables. Por último, la fusión de bloques permite unir bloques adicionales más allá del límite del bloque actual, ofreciendo así granularidades que no serían posibles sin fusión de bloques.

Implementaciones ventajosas de la presente invención son el tema de las reivindicaciones dependientes adjuntas.

A continuación, se describen realizaciones preferidas de la presente solicitud con más detalle con respecto a las figuras, entre las que:

- 5 la Fig. 1 muestra un diagrama de bloques de un codificador de acuerdo con una realización;
- la Fig. 2 muestra un diagrama de bloques de un decodificador de acuerdo con una realización;
- la Fig. 3 muestra un diagrama de bloques de una posible estructura interna del codificador de la Fig. 1;
- 10 la Fig. 4 muestra un diagrama de bloques de una posible estructura interna del decodificador de la Fig. 2;
- la Fig. 5a muestra esquemáticamente una posible subdivisión de una imagen en bloques de raíz arbórea, conjuntos de codificación (bloques) y conjuntos de predicción (particiones);
- 15 la Fig. 5b muestra un árbol de subdivisión del bloque de raíz arbórea mostrado en la Fig. 5a, hasta el nivel de las particiones, de acuerdo con un ejemplo ilustrativo;
- La Fig. 6 muestra una realización para un conjunto de posibles patrones de partición soportados de acuerdo con una realización;
- 20 la Fig. 7 muestra posibles patrones de partición que resultan efectivamente de combinar la fusión de bloques y la partición de bloques cuando se usa la partición de bloques de acuerdo con la Fig. 6;
- 25 la Fig. 8 muestra esquemáticamente bloques candidatos para un modo SALTO/DIRECTO de acuerdo con una realización;
- la Fig. 9-11 muestra porciones de sintaxis de una sintaxis de acuerdo con una realización; y
- 30 la Fig. 12 muestra esquemáticamente la definición de particiones vecinas para una partición de acuerdo con una realización.

Con respecto a la siguiente descripción, se observa que siempre que se utilice el mismo signo de referencia en relación con diferentes figuras, las explicaciones con respecto al elemento respectivo presentado con respecto a una de estas figuras se aplicarán igualmente a las otras figuras, siempre que dicha transferencia de explicaciones de una figura a la otra no entre en conflicto con la descripción restante de esta otra figura. Las realizaciones de la presente invención se describen, entre otros, en las figuras 1 y 2 y en los pasajes correspondientes de la descripción.

Cabe señalar que, en general, en la descripción, la expresión "realización" debe entenderse como que significa "ejemplo" a menos que se use para referirse a la materia que está dentro del alcance de las reivindicaciones adjuntas. En otras palabras, cualquier enseñanza que no esté dentro del alcance de las reivindicaciones adjuntas se presenta solo para proporcionar ejemplos adicionales y no pertenece a la invención.

La Fig. 1 muestra un codificador 10 de acuerdo con una realización de la presente invención. El codificador 10 está configurado para codificar una imagen 20 en un flujo de bits 30. Naturalmente, la imagen 20 podría ser parte de un vídeo, en cuyo caso el codificador sería un codificador de vídeo.

La imagen 20 comprende un bloque 40 que actualmente va a ser codificado por el codificador 10. Tal como se muestra en la Fig. 1, la imagen 20 puede comprender más de un bloque 40. Por ejemplo, la imagen 20 puede subdividirse en una disposición regular de bloques 40 de modo que los bloques 40 estén dispuestos en filas y columnas como se muestra de manera ejemplar en la Fig. 1. Sin embargo, también puede ser posible cualquier otra subdivisión de la imagen 20 en bloques 40. En particular, la subdivisión de la imagen 20 en bloques 40 puede ser fija, es decir, conocida por el decodificador de forma predeterminada o puede señalarse dentro del flujo de bits 30 al decodificador. En particular, los bloques 40 de la imagen 20 pueden variar de tamaño. Por ejemplo, una subdivisión de múltiples árboles, tal como una subdivisión de árbol cuádruple puede aplicarse a la imagen 20 o a una presubdivisión regular de la imagen 20 en bloques de raíz arbórea dispuestos regularmente para obtener bloques 40 que, en este caso, forman los bloques de hojas de la subdivisión de múltiples árboles.

En cualquier caso, el codificador 10 está configurado para señalar dentro del flujo de bits 30 uno de los patrones de partición soportados para el bloque actual 40. Es decir, el codificador 10 decide si en algunos, por ejemplo, el sentido de optimización de distorsión de velocidad es mejor para particionar adicionalmente el bloque 40, y en cuanto a cuál de los patrones de partición soportados se debería usar para un bloque actual 40 con el fin de adaptar la granularidad en la que se establecen determinados parámetros de codificación dentro del bloque actual 40 de la imagen 20. Como se describirá con más detalle a continuación, los parámetros de codificación pueden, por ejemplo, representar

parámetros de predicción tales como parámetros de predicción inter. Tales parámetros de predicción inter pueden, por ejemplo, comprender un índice de imagen de referencia, un vector de movimiento y similares. Los patrones de partición soportados pueden, por ejemplo, comprender un modo de no partición, es decir, una opción de acuerdo con la cual el bloque actual 40 no se particiona adicionalmente, un modo de partición horizontal, es decir, una opción de acuerdo con la cual el bloque actual 40 se subdivide a lo largo de una línea que se extiende horizontalmente en una porción más alta o superior y una porción inferior o más baja y un modo de partición vertical, es decir, una opción de acuerdo con la cual el bloque actual 40 se subdivide verticalmente a lo largo de una línea que se extiende verticalmente en una porción izquierda y una porción derecha. Más allá de esto, los patrones de partición soportados también pueden comprender una opción de acuerdo con la cual el bloque actual 40 se subdivide además regularmente en cuatro bloques adicionales, cada uno asumiendo una cuarta parte del bloque actual 40. Además, la partición puede pertenecer a todos los bloques 40 de la imagen 20 o simplemente un subconjunto adecuado de los mismos, tales como los que tienen un cierto modo de codificación asociado con los mismos, tal como el modo de predicción inter. Además, el conjunto de posibles bloques, para los cuales se aplicará la fusión para la o las particiones del bloque, puede limitarse adicionalmente mediante señalización de flujo de bits para cada bloque 40 para el cual se podría realizar la fusión, en cuanto a si la fusión estará disponible para las particiones del bloque o no. Naturalmente, tal señalización también podría hacerse para cada partición candidata a fusión potencial individualmente. Además, diferentes subconjuntos de los modos de partición soportados pueden estar disponibles para los bloques 40, dependiendo, por ejemplo, del tamaño del bloque, el nivel de subdivisión del bloque 40 en caso de que el mismo sea un bloque de hoja de subdivisión de múltiples árboles, en combinación o individualmente.

Es decir, mientras que la subdivisión de la imagen 20 en bloques para obtener, entre otras cosas, el bloque 40 puede fijarse o señalarse dentro del flujo de bits, el patrón de partición que se usará para el bloque actual 40 se señala dentro del flujo de bits 30 en forma de información de partición. Por consiguiente, la información de partición puede, por lo tanto, considerarse como una especie de extensión de la subdivisión de la imagen 20 en bloques 40. Por otro lado, aún puede permanecer una relevancia adicional de la granularidad original de subdivisión de la imagen 20 en bloques 40. Por ejemplo, el codificador 10 puede configurarse para señalar dentro del flujo de bits 30 el modo de codificación que se usará para la porción respectiva o el bloque 40 de la imagen 20 en la granularidad definida por el bloque 40 mientras que el codificador 10 está configurado para variar los parámetros de codificación del modo de codificación respectivo dentro del bloque respectivo 40 en una granularidad aumentada (más fina) definida por el patrón de partición respectivo elegido para el bloque respectivo 40. Por ejemplo, el modo de codificación señalado en la granularidad de los bloques 40 puede distinguir entre el modo de predicción intra, el modo de predicción inter y similares, tal como el modo de predicción inter temporal, el modo de predicción inter-vista, etc. El tipo de parámetros de codificación asociados con el uno o más sub-bloques (particiones) resultantes de la partición del bloque respectivo 40, entonces depende del modo de codificación asignado al bloque respectivo 40. Por ejemplo, para un bloque intracodificado 40, los parámetros de codificación pueden comprender una dirección espacial a lo largo de la cual se usa el contenido de imagen de porciones previamente decodificadas de la imagen 20 para llenar el bloque respectivo 40. En el caso de un bloque intercodificado 40, los parámetros de codificación pueden comprender, entre otras cosas, un vector de movimiento para la predicción de movimiento compensado.

La Fig. 1 muestra de manera ejemplar que el bloque actual 40 está subdividido en dos bloques adicionales (más pequeños) 50 y 60. En particular, se muestra de manera ejemplar un modo de partición vertical. Los bloques más pequeños 50 y 60 también se pueden denominar sub-bloques 50 y 60 o particiones 50 y 60 o conjuntos de predicción 50 y 60. En particular, el codificador 10 está configurado para eliminar, en aquellos casos en los que el señalado de los patrones de partición soportados especifica una subdivisión del bloque actual 40 en dos o más bloques adicionales 50 y 60, para todos los bloques adicionales excepto un primer bloque adicional de los bloques adicionales 50 y 60 en un orden de codificación, de un conjunto de candidatos a parámetros de codificación para el bloque adicional respectivo, los candidatos a parámetros de codificación que tienen parámetros de codificación que son los mismos que los parámetros de codificación asociados con cualquiera de los bloques adicionales que, al fusionarse con los bloques adicionales respectivos, darían como resultado uno de los patrones de partición soportados. Para ser más precisos, para cada uno de los patrones de partición soportados se define un orden de codificación entre la una o más particiones resultantes 50 y 60. En el caso de la Fig. 1, el orden de codificación se ilustra de manera ejemplar mediante una flecha 70, que define que la partición izquierda 50 se codifica antes de la partición derecha 60. En caso de un modo de partición horizontal, se podría definir que la partición superior se codifica antes de la partición inferior. En cualquier caso, el codificador 10 está configurado para eliminar para la segunda partición 60 en orden de codificación 70, del conjunto de candidatos a parámetros de codificación para la segunda partición respectiva 60, candidatos a parámetros de codificación que tienen parámetros de codificación que son los mismos que los parámetros de codificación asociados con la primera partición 50 con el fin de evitar el resultado de esta fusión, concretamente el hecho de que ambas particiones 50 y 60 tendrían los mismos parámetros de codificación asociados con los mismos que, de hecho, podrían producir igualmente al elegir el modo de no partición para el bloque actual 40 a una velocidad de codificación más baja.

Para ser más precisos, el codificador 10 está configurado para usar la fusión de bloques de una manera eficaz junto con la partición de bloques. En lo que respecta a la fusión de bloques, el codificador 10 determina, para cada partición 50 y 60, un conjunto respectivo de candidatos a parámetros de codificación. El codificador puede configurarse para

determinar los conjuntos de candidatos a parámetros de codificación para cada una de las particiones 50 y 60 en función de los parámetros de codificación asociados con bloques previamente decodificados. En particular, al menos algunos de los candidatos a parámetros de codificación dentro de los conjuntos de candidatos a parámetros de codificación pueden ser iguales, es decir, pueden adoptarse a partir de los parámetros de codificación de particiones previamente decodificadas. Adicional o alternativamente, al menos algunos de los candidatos a parámetros de codificación pueden derivarse de candidatos a parámetros de codificación asociados con más de una partición previamente codificada, a través de una combinación adecuada, tal como una mediana, media o similar. Sin embargo, dado que el codificador 10 está configurado para realizar la determinación del conjunto reducido de candidatos a parámetros de codificación y, si más de uno de tales candidatos a parámetros de codificación permanece después de la eliminación, la elección entre los candidatos a parámetros de codificación no eliminados restantes, para cada una de las particiones no primeras 60 con el fin de establecer parámetros de codificación asociados con la partición respectiva dependiendo del candidato a parámetro de codificación no eliminado o elegido, el codificador 10 está configurado para realizar la eliminación de modo que se eliminen candidatos a parámetros de codificación que conducirían, efectivamente, a una reunificación de las particiones 50 y 60. Es decir, las constelaciones de sintaxis se evitan efectivamente de acuerdo con las cuales una situación de partición efectivo se codifica más compleja que en el caso de señalar directamente este partición simplemente mediante el uso de la información de partición solamente.

Además, como los conjuntos de candidatos a parámetros de codificación se reducen, la cantidad de información conexa necesaria para codificar la información de fusión en el flujo de bits 30 puede disminuir debido al menor número de elementos en estos conjuntos de candidatos. En particular, como el decodificador es capaz de determinar y posteriormente reducir los conjuntos de candidatos a parámetros de codificación de la misma manera que lo hace el codificador de la Fig. 1, el codificador 10 de la Fig. 1 es capaz de explotar los conjuntos reducidos de candidatos a parámetros de codificación mediante, por ejemplo, el uso de menos bits con el fin de insertar un elemento de sintaxis en el flujo de bits 30, especificando cuál de los candidatos a parámetros de codificación no eliminados se empleará para la fusión. Naturalmente, la introducción del elemento de sintaxis en el flujo de bits 30 puede suprimirse completamente en caso de que el número de candidatos a parámetros de codificación no eliminados para la partición respectiva sea simplemente uno. En cualquier caso, debido a la fusión, es decir, establecer los parámetros de codificación asociados con la partición respectiva dependiendo del restante, o elegido, de los candidatos a parámetros de codificación no eliminados, el codificador 10 es capaz de suprimir la inserción completamente nueva de parámetros de codificación para la partición respectiva en el flujo de bits 30, reduciendo así la información conexa también. De acuerdo con algunas realizaciones de la presente solicitud, el codificador 10 puede configurarse para señalar dentro del flujo de bits 30 información de refinamiento para refinar el restante, o elegido de los candidatos a parámetros de codificación para las particiones respectivas.

De acuerdo con la descripción de la Fig. 1, como se estableció anteriormente, el codificador 10 está configurado para determinar los candidatos a fusión que se eliminarán mediante una comparación de sus parámetros de codificación con los parámetros de codificación de la partición, la fusión con los cuales produciría otro patrón de partición soportado. Esta forma de tratar los candidatos a parámetros de codificación eliminaría, efectivamente, al menos un candidato a parámetro de codificación en el caso ilustrativo de la Fig. 1, por ejemplo, siempre que los parámetros de codificación de la partición izquierda 50 formen un elemento del conjunto de candidatos a parámetros de codificación para la partición derecha 60. Sin embargo, los candidatos a parámetros de codificación adicionales también pueden eliminarse en caso de que sean iguales a los parámetros de codificación de la partición izquierda 50. Sin embargo, de acuerdo con otra realización de la presente invención, el codificador 10 podría configurarse para determinar un conjunto de bloques candidatos para cada segunda partición y las siguientes en orden de codificación, con la eliminación de ese o esos bloques candidatos de este conjunto de bloques candidatos que, al fusionarse con la partición respectiva, darían como resultado uno de los patrones de partición soportados. En cierto sentido, esto significa lo siguiente. El codificador 10 puede ser configurado para determinar candidatos a fusión para una partición respectiva 50 o 60 (es decir, la primera y las siguientes en orden de codificación) de modo que cada elemento del conjunto de candidatos tenga exactamente una partición del bloque actual 40 o cualquiera de los bloques 40 previamente codificados, asociada con el mismo en el sentido de que el candidato adopte los parámetros de codificación respectivos de la partición asociada. Por ejemplo, cada elemento del conjunto de candidatos podría ser igual, es decir, adoptado a partir de, uno de tales parámetros de codificación de particiones previamente codificadas, o al menos podría derivarse de los parámetros de codificación de simplemente una de tales particiones previamente codificadas, tal como mediante escalado o refinamiento adicionales usando información de refinamiento enviada adicionalmente. Sin embargo, el codificador 10 también podría configurarse para acompañar tal conjunto de candidatos con otros elementos o candidatos, concretamente candidatos a parámetros de codificación que se han derivado de una combinación de parámetros de codificación de más de una partición previamente codificada, o que se han derivado - por modificación - de parámetros de codificación de una partición previamente codificada, tal como tomando simplemente los parámetros de codificación de una lista de parámetros de movimiento. Para los elementos "combinados", no hay asociación 1:1 entre los parámetros de codificación del elemento candidato respectivo y una partición respectiva. De acuerdo con la primera alternativa de la descripción de la Fig. 1, el codificador 10 podría configurarse para eliminar todos los candidatos de todo el conjunto de candidatos, cuyos parámetros de codificación son iguales a los parámetros de codificación de la partición 50. De acuerdo con la última alternativa de la descripción de la Fig. 1, el codificador 10 podría configurarse para eliminar simplemente el elemento del conjunto candidato que está asociado con la partición

50. Al armonizar ambos puntos de vista, el codificador 10 podría configurarse para eliminar candidatos de la parte del conjunto de candidatos, mostrando una asociación 1:1 con algunas particiones previamente codificadas (por ejemplo, vecinas), sin extender la eliminación (y la búsqueda de candidatos que tengan parámetros de codificación iguales) a la parte restante del conjunto de candidatos que tiene parámetros de codificación que se obtienen por combinación.

- 5 Pero, por supuesto, si una combinación también condujera a una representación redundante, esto podría resolverse eliminando los parámetros de codificación redundantes de la lista o realizando también la comprobación de redundancia para las fechas límite combinadas.

- 10 Después de haber descrito un codificador de acuerdo con una realización de la presente invención, con referencia a la Fig. 2, se describe un decodificador 80 de acuerdo con una realización. El decodificador 80 de la Fig. 2 está configurado para decodificar el flujo de bits 30 que, como se describió anteriormente, señala uno de los patrones de partición soportados para un bloque actual 40 de la imagen 20. El decodificador 80 está configurado para, si el señalado del patrón de partición soportado especifica una subdivisión del bloque actual 40 en dos o más particiones 15 50 y 60, eliminar para todas las particiones excepto la primera partición 50 de las particiones en orden de codificación 70, es decir, para la partición 60 en el ejemplo ilustrado de las Figs. 1 y 2, de un conjunto de candidatos a parámetros de codificación para los candidatos a parámetros de codificación de partición respectivos que tienen parámetros de codificación que son los mismos que, o iguales a parámetros de codificación asociados con cualquiera de las particiones que, al fusionarse con la partición respectiva, darían como resultado uno de los patrones de partición soportados, concretamente uno que no se ha señalado dentro del flujo de bits 30 pero que es, sin embargo, uno de 20 los patrones de partición soportados.

- Es decir, la funcionalidad del decodificador coincide en gran medida con la del codificador descrito con respecto a la Fig. 1. Por ejemplo, el decodificador 80 puede configurarse para, si un número de los candidatos a parámetros de codificación no eliminados es distinto de cero, establecer parámetros de codificación asociados con la partición 25 respectiva 60 dependiendo de uno de los candidatos a parámetros no eliminados. Por ejemplo, el decodificador 80 establece los parámetros de codificación de la partición 60 para que sean iguales a uno del candidato a parámetro de codificación no eliminado, con o sin refinamiento adicional y/o con o sin escalado de acuerdo con una distancia temporal a la que se refieren los parámetros de codificación, respectivamente. Por ejemplo, el candidato a parámetro de codificación que ha de fusionarse con los candidatos no eliminados, puede tener otro índice de imagen de referencia asociado con el mismo distinto de un índice de imagen de referencia señalado explícitamente dentro del flujo de bits 30 para la partición 60. En ese caso, los parámetros de codificación de los candidatos a parámetros de codificación pueden definir vectores de movimiento, cada uno relacionado con un índice de imagen de referencia respectivo, y el decodificador 80 puede configurarse para escalar el vector de movimiento del candidato a parámetro de codificación no eliminado finalmente elegido de acuerdo con la relación entre ambos índices de imagen de referencia. Por lo tanto, 30 de acuerdo con la alternativa recién mencionada, los parámetros de codificación que son sometidos a fusión abarcarían los parámetros de movimiento, mientras que los índices de imagen de referencia estarían separados de los mismos. Sin embargo, como se indicó anteriormente, de acuerdo con realizaciones alternativas, los índices de imagen de referencia también podrían ser una parte de los parámetros de codificación que son sometidos a fusión.

- 40 Se aplica igualmente para el codificador de la Fig. 1 y el decodificador de la Fig. 2 que el comportamiento de fusión puede restringirse a los bloques interpredichos 40. Por consiguiente, el decodificador 80 y el codificador 10 pueden configurarse para soportar modos de predicción intra e inter para el bloque actual 40 y realizar la fusión y eliminación de candidatos simplemente en caso de que el bloque actual 40 se codifique en modo de predicción inter. Por consiguiente, simplemente los parámetros de codificación/predicción de tales particiones previamente codificadas 45 interpredichas se pueden usar para determinar/construir la lista de candidatos.

- Como ya se discutió anteriormente, los parámetros de codificación pueden ser parámetros de predicción y el decodificador 80 puede configurarse para usar los parámetros de predicción de las particiones 50 y 60 con el fin de derivar una señal de predicción para la partición respectiva. Naturalmente, el codificador 10 también realiza la 50 derivación de la señal de predicción de la misma manera. Sin embargo, el codificador 10 establece adicionalmente los parámetros de predicción junto con todos los demás elementos de sintaxis dentro del flujo de bits 30 con el fin de lograr cierta optimización en un sentido de optimización adecuado.

- Además, como ya se describió anteriormente, el codificador puede configurarse para insertar un índice en un candidato a parámetro de codificación no eliminado simplemente en caso de que el número de candidatos a parámetros de codificación no eliminados para una partición respectiva sea mayor que uno. Por consiguiente, el decodificador 80 puede configurarse para, dependiendo del número de candidatos a parámetros de codificación no eliminados para, por ejemplo, la partición 60, simplemente esperar que el flujo de bits 30 comprenda un elemento de sintaxis que especifique cuál de los candidatos a parámetros de codificación no eliminados se emplea para fusionarse, si el número 60 de candidatos a parámetros de codificación no eliminados es mayor que uno. Sin embargo, que se produzca el caso de que el conjunto de candidatos se vuelva inferior a dos, generalmente podría excluirse al extender, como se describió anteriormente, la lista/conjunto de candidatos usando parámetros de codificación combinados, es decir, parámetros que se han derivado por combinación de los parámetros de codificación de más de una - o más de dos - particiones previamente codificadas, con la restricción del rendimiento de la reducción del conjunto de candidatos a aquellos

candidatos que se han obtenido al adoptar, o derivar de, los parámetros de codificación de exactamente una partición previamente codificada. También es posible lo contrario, es decir, eliminar generalmente todos los candidatos a parámetros de codificación que tienen el mismo valor que los de la partición, lo que da como resultado otro patrón de partición soportado.

5

Con respecto a la determinación, el decodificador 80 actúa como lo hace el codificador 10. Es decir, el decodificador 80 puede configurarse para determinar el conjunto de candidatos a parámetros de codificación para la partición o las particiones después de la primera partición 50 en orden de codificación 70 en función de los parámetros de codificación asociados con particiones previamente decodificadas. Es decir, un orden de codificación no solo se define entre las

10 particiones 50 y 60 de un bloque respectivo 40, sino también entre los bloques 40 de la propia imagen 20. Todas las particiones que han sido codificadas antes de la partición 60 pueden, por lo tanto, servir de base para la determinación del conjunto de candidatos a parámetros de codificación para cualquiera de las particiones posteriores, tal como la partición 60 en el caso de la Fig. 2. Como también se describió anteriormente, el codificador y decodificador pueden restringir la determinación del conjunto de candidatos a parámetros de codificación a particiones en un determinado

15 vecindario espacial y/o temporal. Por ejemplo, el decodificador 80 puede configurarse para determinar el conjunto de candidatos a parámetros de codificación para una partición no primera 60 en función de los parámetros de codificación asociados con particiones previamente decodificadas vecinas a la partición no primera respectiva, donde tales particiones pueden estar fuera y dentro del bloque actual 40. Naturalmente, la determinación de los candidatos a fusión también se puede realizar para la primera partición en orden de codificación. Simplemente no se realiza la extracción.

20

Coincidiendo con la descripción de la Fig. 1, el decodificador 80 puede configurarse para determinar el conjunto de candidatos a parámetros de codificación para la partición no primera respectiva 60 de un conjunto inicial de particiones previamente decodificadas, excluyendo las que se codifican en un modo de predicción intra.

25

Además, en caso de que el codificador introduzca información de subdivisión en el flujo de bits con el fin de subdividir la imagen 20 en los bloques 40, el decodificador 80 puede configurarse para recuperar la subdivisión de la imagen 20 en tales bloques de codificación 40 de acuerdo con la información de subdivisión en el flujo de bits 30.

30

Con respecto a la Fig. 1 y la Fig. 2, cabe señalar que la señal residual para el bloque actual 40 puede transmitirse a través del flujo de bits 30 en una granularidad que puede diferir de la granularidad definida por las particiones con respecto a los parámetros de codificación. Por ejemplo, el codificador 10 de la Fig. 1 puede configurarse para subdividir el bloque 40 en uno o más bloques de transformación de manera paralela o independiente de la partición en las particiones 50 y 60. El codificador puede señalar la subdivisión de bloque de transformación respectiva para el bloque 40 a través de información de subdivisión adicional. El decodificador 80, a su vez, puede configurarse para recuperar esta subdivisión adicional del bloque 40 en uno o más bloques de transformación de acuerdo con la información de subdivisión adicional en el flujo de bits, y para derivar una señal residual del bloque actual 40 del flujo de bits en conjuntos de estos bloques de transformación. La importancia de la partición de bloques de transformación puede ser que la transformación, tal como DCT, en el codificador y la transformación inversa correspondiente tal como IDCT en el decodificador se realizan dentro de cada bloque de transformación del bloque 40 individualmente. Con el fin de

35 reconstruir la imagen 20 como el bloque 40, el codificador 10, a continuación, combina, tal como añade, la señal de predicción derivada mediante la aplicación de los parámetros de codificación en las respectivas particiones 50 y 60, y la señal residual, respectivamente. Sin embargo, se observa que la codificación residual puede no implicar ninguna transformación y transformación inversa respectivamente, y que el residuo de predicción se codifica en el dominio espacial en su lugar, por ejemplo.

45

Antes de describir más detalles posibles de realizaciones adicionales a continuación, se describirá una posible estructura interna de codificador y decodificador de las Figs. 1 y 2 con respecto a las Figs. 3 y 4. La Fig. 3 muestra de manera ejemplar cómo se puede construir internamente el codificador 10. Tal como se muestra, el codificador 10 puede comprender un sustractor 108, un transformador 100 y un generador de flujo de bits 102, que pueden, tal como se indica en la Fig. 3, realizar una codificación de entropía. Los elementos 108, 100 y 102 están conectados en serie entre una entrada 112 que recibe la imagen 20 y una salida 114 que produce como salida el flujo de bits 30 mencionado anteriormente. En particular, el sustractor 108 tiene su entrada no inversora conectada a la entrada 112 y el transformador 100 está conectado entre una salida del sustractor 108 y una primera entrada del generador de flujo de bits 102 que, a su vez, tiene una salida conectada a la salida 114. El codificador 10 de la Fig. 3 comprende además

50 un transformador inverso 104 y un sumador 110 conectados en serie, en el orden mencionado, a la salida del transformador 100. El codificador 10 comprende además un predictor 106, que está conectado entre una salida del sumador 110 y una entrada adicional del sumador 110 y la entrada inversora del sustractor 108.

55

60

Los elementos de la Fig. 3 interactúan de la siguiente manera: El predictor 106 predice porciones de la imagen 20 con el resultado de la predicción, es decir, la señal de predicción, que se aplica a la entrada inversora del sustractor 108. La salida del sustractor 108, a su vez, representa la diferencia entre la señal de predicción y la porción respectiva de la imagen 20, es decir, una señal residual. La señal residual es sometida a codificación de transformación en el transformador 100. Es decir, el transformador 100 puede realizar una transformación, tal como una DCT o similar, y una cuantificación posterior en la señal residual transformada, es decir, los coeficientes de transformación, para

obtener niveles de coeficiente de transformación. El transformador inverso 104 reconstruye la salida de señal residual final producida como salida por el transformador 100 para obtener una señal residual reconstruida que corresponde a la señal residual introducida en el transformador 100, excepto por la pérdida de información debido a la cuantificación en el transformador 100. La adición de la señal residual reconstruida y la señal de predicción como salida del predictor 106 da como resultado una reconstrucción de la porción respectiva de la imagen 20 y se reenvía desde la salida del sumador 110 a la entrada del predictor 106. El predictor 106 funciona en diferentes modos, tal como se describió anteriormente, tal como un modo de predicción intra, modo de predicción inter y similares. El modo de predicción y los parámetros de codificación o predicción correspondientes aplicados por el predictor 106 con el fin de obtener la señal de predicción, se reenvían mediante el predictor 106 al codificador de entropía 102 para su inserción en el flujo de bits.

En la Fig. 4 se muestra una posible implementación de la estructura interna del decodificador 80 de la Fig. 2, que corresponde a la posibilidad mostrada en la Fig. 3 con respecto al codificador. Como se muestra en la misma, el decodificador 80 puede comprender un extractor de flujo de bits 150 que, como se muestra en la Fig. 4, puede implementarse como un decodificador de entropía, un transformador inverso 152 y un sumador 154, que están, en el orden mencionado, conectados entre una entrada 158 y una salida 160 del decodificador. Además, el decodificador de la Fig. 4 comprende un predictor 156 conectado entre una salida del sumador 154 y una entrada adicional del mismo. El decodificador de entropía 150 está conectado a una entrada de parámetro del predictor 156.

Describiendo brevemente la funcionalidad del decodificador de la Fig. 4, el decodificador de entropía 150 es para extraer toda la información contenida en el flujo de bits 30. El esquema de codificación de entropía usado puede ser codificación de longitud variable o codificación aritmética. Mediante esto, el decodificador de entropía 150 recupera del flujo de bits los niveles de coeficiente de transformación que representan la señal residual y los reenvía al transformador inverso 152. Además, el decodificador de entropía 150 recupera del flujo de bits todos los modos de codificación y parámetros de codificación asociados y los reenvía al predictor 156. De manera adicional, la información de partición y la información de fusión se extraen del flujo de bits mediante el extractor 150. La señal residual transformada inversamente, es decir, reconstruida y la señal de predicción tal como se deriva mediante el predictor 156 se combinan, tal como se añaden, mediante el sumador 154 que, a su vez, produce como salida la señal reconstruida así recuperada en la salida 160 y la reenvía al predictor 156.

Como queda claro al comparar las Figs. 3 y 4, los elementos 152, 154 y 156 corresponden funcionalmente a los elementos 104, 110 y 106 de la Fig. 3.

En la descripción anterior de las Figs. 1 a 4, se han presentado varias posibilidades diferentes con respecto a posibles subdivisiones de la imagen 20 y la granularidad correspondiente al variar algunos de los parámetros involucrados en la codificación de la imagen 20. Una de tales posibilidades se describe nuevamente con respecto a la Fig. 5a y la Fig. 5b. La Fig. 5a muestra una porción de una imagen 20. De acuerdo con la realización de la Fig. 5a, el codificador y el decodificador están configurados para subdividir en primer lugar la imagen 20 en bloques de raíz arborea 200. Uno de tales bloques de raíz arborea se muestra en la Fig. 5a. La subdivisión de la imagen 20 en bloques de raíz arborea se realiza regularmente en filas y columnas como se ilustra con líneas punteadas. El tamaño de los bloques de raíz arborea 200 puede ser seleccionado por el codificador y señalado al decodificador por el flujo de bits 30. Alternativamente, el tamaño de estos bloques de raíz arborea 200 puede fijarse de forma predeterminada. Los bloques de raíz arborea 200 se subdividen mediante el uso de partición de árbol cuádruple con el fin de producir los bloques 40 identificados anteriormente que pueden denominarse bloques de codificación o conjuntos de codificación. Estos bloques de codificación o conjuntos de codificación se dibujan con líneas continuas delgadas en la Fig. 5a. Mediante esto, el codificador acompaña cada bloque de raíz arborea 200 con información de subdivisión e inserta la información de subdivisión en el flujo de bits. Esta información de subdivisión indica cómo el bloque de raíz arborea 200 debe subdividirse en bloques 40. A una granularidad de, y en conjuntos de, estos bloques 40, el modo de predicción varía dentro de la imagen 20. Tal como se indicó anteriormente, cada bloque 40 - o cada bloque que tiene un cierto modo de predicción, tal como modo de predicción inter - va acompañado de información de partición en cuanto a qué patrón de partición soportado se usa para el bloque respectivo 40. En el caso ilustrativo de la Fig. 5a, para muchos bloques de codificación 40, se ha elegido el modo de no partición de modo que el bloque de codificación 40 coincida espacialmente con la partición correspondiente. En otras palabras, el bloque de codificación 40 es, simultáneamente, una partición que tiene un conjunto respectivo de parámetros de predicción asociados con la misma. El tipo de parámetros de predicción, a su vez, depende del modo asociado con el bloque de codificación respectivo 40. Sin embargo, se muestra ejemplarmente que otros bloques de codificación se particionan adicionalmente. El bloque de codificación 40 en la esquina superior derecha del bloque de raíz arborea 200, por ejemplo, se muestra que está particionado en cuatro particiones, mientras que el bloque de codificación en la esquina inferior derecha del bloque de raíz arborea 200 se muestra de manera ejemplar que está subdividido verticalmente en dos particiones. La subdivisión para partición en particiones se ilustra mediante líneas punteadas. La Fig. 5a también muestra el orden de codificación entre las particiones así definidas. Como se muestra, se usa un primer orden transversal en profundidad. A través de los bordes del bloque de raíz de árbol, el orden de codificación se puede continuar en un orden de escaneo de acuerdo con el cual las filas de bloques de raíz arborea 200 se escanean en fila de arriba a abajo de la imagen 20. Mediante esta medida, es posible tener la máxima probabilidad de que una determinada partición tenga una partición

previamente codificada adyacente a su borde superior y su borde izquierdo. Cada bloque 40 - o cada bloque que tiene un determinado modo de predicción, tal como el modo de predicción inter - puede tener un indicador de interruptor de fusión dentro del flujo de bits que indica si la fusión se activa para las particiones correspondientes en el mismo o no. Cabe señalar que la partición de los bloques en particiones/conjuntos de predicción podría restringirse a una partición

de un máximo de dos particiones, haciéndose simplemente una excepción de esta regla para el tamaño de bloque más pequeño posible de los bloques 40. En caso de usar la subdivisión de árbol cuádruple con el fin de obtener los bloques 40, esto podría evitar la redundancia entre la información de subdivisión para subdividir la imagen 20 en el bloque 40 y la información de partición para subdividir el bloque 40 en particiones. Alternativamente, se podrían permitir simplemente particionamientos en una o dos particiones, incluyendo o no incluyendo asimétricos.

La Fig. 5b muestra un árbol de subdivisión. Con líneas continuas, se ilustra la subdivisión del bloque de raíz arbórea 200, mientras que las líneas punteadas simbolizan la partición de los bloques de hojas de la subdivisión de árbol cuádruple, que son los bloques de codificación 40. Es decir, la partición de los bloques de codificación representa una especie de extensión de la subdivisión cuádruple.

Como ya se señaló anteriormente, cada bloque de codificación 40 puede subdividirse paralelamente en bloques de transformación de modo que los bloques de transformación puedan representar una subdivisión diferente del bloque de codificación 40 respectivo. Para cada uno de estos bloques de transformación, que no se muestran en las Figs. 5a y 5b, una transformación con el fin de transformar la señal residual de los bloques de codificación se puede realizar por separado.

A continuación, se describen realizaciones adicionales de la presente invención. Si bien las realizaciones anteriores se concentraron en la relación entre la fusión de bloques, por un lado, y la partición de bloques, por otro lado, la siguiente descripción también incluye aspectos de la presente solicitud relacionados con otros principios de codificación conocidos en los presentes códecs, tales como los modos de SALTO/DIRECTO. Sin embargo, la descripción posterior no se considerará como una mera descripción de realizaciones separadas, es decir, realizaciones separadas de las descritas anteriormente. Más bien, la siguiente descripción también revela posibles detalles de implementación para las realizaciones descritas anteriormente. Por consiguiente, la descripción a continuación usa signos de referencia de las figuras ya descritas anteriormente, de modo que una posible implementación respectiva descrita a continuación, también definirá posibles variaciones de las realizaciones descritas anteriormente. La mayoría de estas variaciones se pueden transferir individualmente a las realizaciones anteriores.

En otras palabras, las realizaciones de la presente solicitud describen procedimientos para reducir la velocidad de información conexa en aplicaciones de codificación de imagen y vídeo mediante la fusión de los elementos de sintaxis asociados con conjuntos particulares de muestras, es decir, bloques, con el fin de transmitir parámetros de codificación asociados. Las realizaciones de la presente solicitud son particularmente capaces de considerar la combinación de la fusión de elementos de sintaxis con una partición de partes de una imagen en varios patrones de partición y la combinación con modos de SALTO/DIRECTO, en los que los parámetros de codificación se infieren de un vecindario espacial y/o temporal de un bloque actual. En la medida en que, las realizaciones descritas anteriormente pueden modificarse para implementar la fusión para conjuntos de muestras, es decir, bloques, en combinación con diferentes patrones de partición y modos de SALTO/DIRECTO.

Además, antes de describir estas variaciones y más detalles, se presenta una visión general sobre los códecs de imagen y vídeo.

En aplicaciones de codificación de imagen y vídeo, las matrices de muestras asociadas con una imagen generalmente se particionan en conjuntos particulares de muestras (o conjuntos de muestras), que pueden representar bloques rectangulares o cuadráticos o cualquier otra colección de muestras que incluya regiones con forma arbitraria, triángulos o cualquier otra forma. La subdivisión de las matrices de muestras puede fijarse mediante la sintaxis o la subdivisión se señala (al menos parcialmente) dentro del flujo de bits. Para mantener la velocidad de información conexa para señalar la información de subdivisión pequeña, la sintaxis generalmente permite solo un número limitado de opciones que dan como resultado partición simple, tal como la subdivisión de bloques en bloques más pequeños. Un esquema de partición usado a menudo es la partición de bloques cuadrados en cuatro bloques cuadrados más pequeños, o en dos bloques rectangulares del mismo tamaño, o en dos bloques rectangulares de diferentes tamaños, donde la partición empleada realmente se señala dentro del flujo de bits. Los conjuntos de muestras están asociados con parámetros de codificación particulares que pueden especificar información de predicción o modos de codificación residuales, etc. En las aplicaciones de codificación de vídeo, a menudo se realiza una partición con el fin de representación de movimiento. Todas las muestras de un bloque (dentro de un patrón de partición) están asociadas con el mismo conjunto de parámetros de movimiento, que pueden incluir parámetros que especifiquen el tipo de predicción (por ejemplo, lista 0, lista 1 o bipredicción; y/o predicción traslacional o afín o una predicción con un modelo de movimiento diferente), parámetros que especifiquen las imágenes de referencia empleadas, parámetros que especifican el movimiento con respecto a las imágenes de referencia (por ejemplo, vectores de desplazamiento, vectores de parámetros de movimiento afín o vectores de parámetros de movimiento para cualquier otro modelo de movimiento), que generalmente se transmiten como una diferencia a un predictor, parámetros que especifican la

precisión de los parámetros de movimiento (por ejemplo, precisión de media muestra o cuarto de muestra), parámetros que especifican la ponderación de la señal de muestra de referencia (por ejemplo, con el propósito de compensación de iluminación) o parámetros que especifican el filtro de interpolación que se emplea para derivar la señal de predicción con compensación de movimiento del bloque actual. Se asume que para cada conjunto de muestras, se transmiten parámetros de codificación individuales (por ejemplo, para especificar la predicción y/o codificación residual). Con el fin de lograr una eficiencia de codificación mejorada, la presente invención presenta un procedimiento y realizaciones particulares para fusionar dos o más conjuntos de muestras en los llamados grupos de conjuntos de muestras. Todos los conjuntos de muestras de tal grupo comparten los mismos parámetros de codificación, que se pueden transmitir junto con uno de los conjuntos de muestras en el grupo. Al hacerlo, los parámetros de codificación no necesitan transmitirse para cada conjunto de muestras del grupo de conjuntos de muestras individualmente, sino que, en cambio, los parámetros de codificación se transmiten solo una vez para todo el grupo de conjuntos de muestras. Como resultado, la velocidad de información conexa para transmitir los parámetros de codificación se reduce y la eficiencia de codificación general se mejora. Como estrategia alternativa, se puede transmitir un refinamiento adicional para uno o más de los parámetros de codificación para uno o más de los conjuntos de muestras de un grupo de conjuntos de muestras. El refinamiento se puede aplicar a todos los conjuntos de muestras de un grupo o solo al conjunto de muestras para el que se transmite.

Las realizaciones de la presente invención se refieren particularmente a la combinación del procedimiento de fusión con una partición de un bloque en varios sub-bloques 50, 60 (como se mencionó anteriormente). Por lo general, los sistemas de codificación de imagen o vídeo soportan varios patrones de partición para un bloque 40. Como ejemplo, un bloque cuadrado puede no partitionarse o puede partitionarse en cuatro bloques cuadrados del mismo tamaño, o en dos bloques rectangulares del mismo tamaño (donde el bloque cuadrado puede dividirse vertical u horizontalmente), o en bloques rectangulares de diferentes tamaños (horizontal o verticalmente). Los patrones de partición ejemplares descritos se ilustran en la Fig. 6. Además de la descripción anterior, la partición puede implicar incluso más de un nivel de partición. Por ejemplo, los sub-bloques cuadrados también pueden partitionarse adicionalmente de manera opcional usando los mismos patrones de partición. El problema que surge cuando tal procedimiento de partición se combina con un procedimiento de fusión que permite la fusión de un bloque (cuadrado o rectangular) con, por ejemplo, uno de sus bloques vecinos es la misma partición resultante se puede lograr mediante diferentes combinaciones de los patrones de partición y señales de fusión. Por lo tanto, la misma información se puede transmitir en el flujo de bits usando diferentes palabras de código, lo que es claramente subóptimo con respecto a la eficiencia de codificación. Como ejemplo simple, consideramos un bloque cuadrado que no está partitionado adicionalmente (como se ilustra en la esquina superior izquierda de la Fig. 6). Esta partición se puede señalar directamente mediante el envío de un elemento de sintaxis de que este bloque 40 no está subdividido. Pero el mismo patrón también se puede señalar mediante el envío de un elemento de sintaxis que especifica que este bloque está, por ejemplo, subdividido en dos bloques rectangulares alineados vertical (u horizontalmente) 50, 60. A continuación podemos transmitir información de fusión que especifica que el segundo de estos bloques rectangulares se fusiona con el primer bloque rectangular, lo que da como resultado exactamente la misma partición que cuando señalamos que el bloque no se divide adicionalmente. También puede lograrse lo mismo especificando primero que el bloque se subdivide en cuatro sub-bloques cuadrados y a continuación transmitir información de fusión que fusiona efectivamente todos estos cuatro bloques. Este concepto es claramente subóptimo (ya que tenemos diferentes palabras de código para señalar lo mismo).

Las realizaciones de la presente invención describen un concepto y posibilidades para reducir la velocidad de información conexa y así aumentar la eficiencia de codificación para una combinación del concepto de fusión con el concepto de proporcionar diferentes patrones de partición para un bloque. Si observamos los patrones de partición de ejemplo en la Fig. 6, la "simulación" del bloque no dividido adicionalmente por cualquiera de los patrones de partición con dos bloques rectangulares puede evitarse cuando prohibimos (es decir, excluimos de la especificación de sintaxis de flujo de bits) el caso de que un bloque rectangular se fusione con un primer bloque rectangular. Cuando se analiza más profundamente el problema, también es posible «simular» el patrón no subdividido mediante la fusión del segundo bloque rectangular con cualquier otro vecino (es decir, no el primer bloque rectangular) que esté asociado con los mismos parámetros (por ejemplo, información para especificar la predicción) que el primer bloque rectangular. Las realizaciones de la presente invención condicionan el envío de información de fusión de manera que el envío de parámetros de fusión particulares se excluye de la sintaxis de flujo de bits cuando estos parámetros de fusión dan como resultado un patrón que también se puede lograr señalando uno de los patrones de partición soportados. Como ejemplo, si el patrón de partición actual especifica la subdivisión en dos bloques rectangulares, como se muestra en la Fig. 1 y 2, por ejemplo, antes de enviar la información de fusión para el segundo bloque, es decir, 60 en el caso de la Fig. 1 y 2, se puede verificar cuál de los posibles candidatos a fusión tiene los mismos parámetros (por ejemplo, parámetros para especificar la señal de predicción) que el primer bloque rectangular, es decir, 50 en el caso de la Fig. 1 y 2. Y todos los candidatos que tienen los mismos parámetros de movimiento (incluido el primer bloque rectangular en sí) se eliminan del conjunto de candidatos a fusión. Las palabras de código o banderas que se transmiten para señalar la información de fusión se adaptan al conjunto de candidatos resultante. Si el conjunto de candidatos queda vacío debido a la verificación de parámetros, no se transmite información de fusión. Si el conjunto de candidatos consiste en una sola entrada, solo se señala si el bloque está fusionado o no, pero no es necesario señalar al candidato, ya que se puede derivar en el lado del decodificador, etc. Para el ejemplo anterior, el mismo concepto

también se emplea para el patrón de partición que divide un bloque cuadrado en cuatro bloques cuadrados más pequeños. Aquí, el envío de banderas de fusión se adapta de manera que ni el patrón de partición que no especifica ninguna subdivisión ni ninguno de los dos patrones de partición especifican una subdivisión en dos bloques rectangulares del mismo tamaño se pueden lograr mediante una combinación de banderas de fusión. Aunque describimos el concepto más en el ejemplo anterior con patrones de partición específicos, debería quedar claro que el mismo concepto (evitar la especificación de un patrón de partición particular mediante una combinación de otro patrón de partición y la información de fusión correspondiente) puede emplearse para cualquier otro conjunto de patrones de partición.

- 10 La ventaja de la invención descrita con respecto a un concepto en el que solo se permite la partición es que se proporciona una libertad mucho mayor para señalar la partición de una imagen en partes que están asociadas con los mismos parámetros (por ejemplo, para especificar la señal de predicción). Como ejemplo, los patrones de partición adicionales que resultan de la fusión de bloques cuadrados que de un bloque más grande subdividido se representan en la Fig. 7. Sin embargo, cabe señalar que se pueden lograr muchos más patrones resultantes mediante la fusión con bloques vecinos adicionales (fuera del bloque subdividido anteriormente). Con solo unas pocas palabras de código para señalar la información de partición y fusión, se proporciona una variedad de posibilidades de partición y un codificador puede seleccionar la mejor opción (para una complejidad de codificador dada) en sentido de distorsión de velocidad (por ejemplo, minimizando una medida de distorsión de velocidad particular). La ventaja en comparación con una estrategia en la que solo se proporciona un patrón de partición (por ejemplo, una subdivisión en cuatro bloques del mismo tamaño) en combinación con la estrategia de fusión es que los patrones usados a menudo (como, por ejemplo, formas rectangulares de diferentes tamaños) pueden señalarse mediante una palabra de código corta en lugar de varias banderas de subdivisión y fusión.

- 25 Otro aspecto que debe considerarse es que el concepto de fusión es, en cierto sentido, similar a los modos de SALTO o DIRECTO que se encuentran en los diseños de codificación de vídeo. En los modos de SALTO/DIRECTO, básicamente no se transmiten parámetros de movimiento para un bloque actual, sino que se infieren de un vecindario espacial y/o temporal. En un concepto eficiente particular de los modos de SALTO/DIRECTO, se crea una lista de candidatos a parámetros de movimiento (índices de trama de referencia, vectores de desplazamiento, etc.) a partir de un vecindario espacial y/o temporal y se transmite un índice a esta lista que especifica cuál de los parámetros candidatos se elige. Para bloques bipredichos (o tramas de múltiples hipótesis), se puede señalar un candidato separado para cada lista de referencia. Los posibles candidatos pueden incluir el bloque en la parte superior del bloque actual, el bloque a la izquierda del bloque actual, el bloque en la parte superior izquierda del bloque actual, el bloque en la parte superior derecha del bloque actual, el predictor de mediana de varios de estos candidatos, el bloque coubicado en una o más tramas de referencia anteriores (o cualquier otro bloque ya codificado, o una combinación obtenida de bloques ya codificados). Al combinar el concepto de fusión con el modo de SALTO/DIRECTO, debería garantizarse que tanto el modo de SALTO/DIRECTO como el modo de fusión no incluyan los mismos candidatos. Esto se puede lograr mediante diferentes configuraciones. Es posible habilitar el modo de SALTO/DIRECTO (por ejemplo, con más candidatos que el modo de fusión) solo para bloques particulares (por ejemplo, con un tamaño mayor que un tamaño especificado, o solo para bloques cuadrados, etc.) y no soportar el modo de fusión para estos bloques. O se puede eliminar el modo de SALTO/DIRECTO y todos los candidatos (incluidos los parámetros que representan una combinación de parámetros para los bloques vecinos espaciales/temporales) se añaden al modo de fusión como candidatos. Esta opción también se había mencionado anteriormente con respecto a las Figs. 1-5. El conjunto de candidatos aumentado solo se podría usar para bloques particulares (con un tamaño mayor que un tamaño mínimo dado, o bloques cuadrados, etc.), donde para otros bloques se usa un conjunto de candidatos reducido. O como una variante adicional, el modo de fusión se usa con un conjunto de candidatos reducido (por ejemplo, solo el vecino superior e izquierdo) y se usan candidatos adicionales (por ejemplo, el modo superior izquierdo, el bloque coubicado, etc.) para el modo de SALTO/DIRECTO. También en tales configuraciones, los modos de SALTO/DIRECTO solo se pueden permitir para bloques particulares (con un tamaño mayor que un tamaño mínimo dado, o bloques cuadrados, etc.), mientras que el modo de fusión se permite para un conjunto más grande de bloques. La ventaja de tales combinaciones es que se proporcionan múltiples opciones para señalar la reutilización de parámetros ya transmitidos (por ejemplo, para especificar la predicción) para diferentes tamaños de bloque. Como ejemplo, para bloques cuadrados más grandes se proporcionan más opciones, ya que aquí la velocidad binaria de gasto adicional proporciona un aumento en la eficiencia de distorsión de velocidad. Para bloques más pequeños, se proporciona un conjunto más pequeño de opciones. Un aumento del conjunto de candidatos aquí no traería ninguna ganancia en la eficiencia de distorsión de velocidad debido a la pequeña proporción de muestras por bit requerida para señalar el candidato seleccionado.

- 60 Como se mencionó anteriormente, las realizaciones de la presente invención también proporcionan un codificador con una mayor libertad para crear un flujo de bits, ya que la estrategia de fusión aumenta significativamente las posibilidades de número para seleccionar una partición para las matrices de muestras de una imagen. Dado que el codificador puede elegir entre más opciones, por ejemplo, para minimizar una medida de distorsión de velocidad particular, se puede mejorar la eficiencia de codificación. Como ejemplo, algunos de los patrones adicionales que se pueden representar mediante una combinación de subparticionamiento y fusión (por ejemplo, los patrones de la Fig. 7) se pueden probar adicionalmente (usando los tamaños de bloque correspondientes para la estimación del

movimiento y la decisión de modo) y el mejor de los patrones proporcionados mediante partición puramente (Fig. 6) y mediante partición y fusión (Fig. 7) se puede seleccionar en función de una medida de distorsión de velocidad particular. Además, para cada bloque se puede probar si una fusión con cualquiera de los conjuntos de candidatos ya codificados produce una disminución de una medida de distorsión de velocidad particular y a continuación se establecen las banderas de fusión correspondientes durante el procedimiento de codificación. En resumen, hay varias posibilidades de operar un codificador. En una estrategia simple, el codificador podría determinar primero la mejor subdivisión de las matrices de muestras (como en los esquemas de codificación de última generación). Y luego podría verificar para cada conjunto de muestras, si una fusión con otro conjunto de muestras u otro grupo de conjuntos de muestras reduce una medida de costo de distorsión de velocidad particular. En este caso, los parámetros de predicción asociados con el grupo fusionado de conjuntos de muestras pueden volver a estimarse (por ejemplo, realizando una nueva búsqueda de movimiento) o los parámetros de predicción que ya se han determinado para el conjunto de muestras actual y el conjunto de muestras candidato (o grupo de conjuntos de muestras) para la fusión podrían evaluarse para el grupo considerado de conjuntos de muestras. En una estrategia más amplia, podría evaluarse una medida de distorsión de costo de distorsión de velocidad particular para grupos de candidatos adicionales de conjuntos de muestras. Como ejemplo particular, cuando se prueban los diversos patrones de partición posibles (véase la Fig. 6, por ejemplo), parte o la totalidad del patrón que se puede representar mediante una combinación de partición y fusión (véase la Fig. 7, por ejemplo) se puede probar adicionalmente. Es decir, para todos los patrones se lleva a cabo un procedimiento específico de estimación de movimiento y decisión de modo y se selecciona el patrón que produce la medida de distorsión de velocidad más pequeña. Este procedimiento también se puede combinar con el procedimiento de baja complejidad descrito anteriormente, de modo que para los bloques resultantes se prueba adicionalmente si una fusión con bloques ya codificados (por ejemplo, fuera de los patrones de la Fig. 6 y la Fig. 7) produce una disminución en una medida de distorsión de velocidad.

A continuación, se describen alguna posible implementación detallada para las realizaciones descritas anteriormente, tales como para los codificadores de las Figs. 1 y 3 y los decodificadores de las Figs. 2 y 4. Como ya se ha señalado anteriormente, los mismos son utilizables en la codificación de imagen y vídeo. Tal como se describió anteriormente, las imágenes o conjuntos particulares de matrices de muestras para las imágenes pueden descomponerse en bloques, que están asociados con parámetros de codificación particulares. Las imágenes generalmente consisten en múltiples matrices de muestras. Además, una imagen también puede asociarse con matrices de muestras auxiliares adicionales, que pueden, por ejemplo, especificar información de transparencia o mapas de profundidad. Las matrices de muestras de una imagen (que incluyen matrices de muestras auxiliares) se pueden agrupar en uno o más denominados grupos de planos, donde cada grupo de planos consiste en una o más matrices de muestras. Los grupos de planos de una imagen se pueden codificar independientemente o, si la imagen está asociada con más de un grupo de planos, con predicción de otros grupos de planos de la misma imagen. Cada grupo de planos generalmente se descompone en bloques. Los bloques (o los bloques correspondientes de las matrices de muestras) se predicen mediante predicción interimagen o predicción intraimagen. Los bloques pueden tener diferentes tamaños y pueden ser cuadráticos o rectangulares. La partición de una imagen en bloques puede ser fijado por la sintaxis, o puede ser (al menos parcialmente) señalado dentro del flujo de bits. A menudo se transmiten elementos de sintaxis que señalan la subdivisión para bloques de tamaños predefinidos. Tales elementos de sintaxis pueden especificar si y cómo un bloque se subdivide en bloques más pequeños y se asocia con parámetros de codificación, por ejemplo, con fines de predicción. Un ejemplo de posibles patrones de partición se muestra en la Fig. 6. Para todas las muestras de un bloque (o los bloques correspondientes de conjuntos de muestras), la decodificación de los parámetros de codificación asociados se especifica de cierta manera. En el ejemplo, todas las muestras de un bloque se predicen usando el mismo conjunto de parámetros de predicción, tales como índices de referencia (identificar una imagen de referencia en el conjunto de imágenes ya codificadas), parámetros de movimiento (especificar una medida para el movimiento de un bloque entre una imagen de referencia y la imagen actual), parámetros para especificar el filtro de interpolación, modos de predicción intra, etc. Los parámetros de movimiento se pueden representar mediante vectores de desplazamiento con una componente horizontal y vertical o mediante parámetros de movimiento de orden superior, tales como parámetros de movimiento afín que consisten en seis componentes. También es posible que más de un conjunto de parámetros de predicción particulares (tales como índices de referencia y parámetros de movimiento) estén asociados con un solo bloque. En ese caso, para cada conjunto de estos parámetros de predicción particulares, se genera una única señal de predicción intermedia para el bloque (o los bloques correspondientes de conjuntos de muestras), y la señal de predicción final se construye mediante una combinación que incluye superponer las señales de predicción intermedias. Los parámetros de ponderación correspondientes y potencialmente también un desplazamiento constante (que se añade a la suma ponderada) se pueden fijar para una imagen, o una imagen de referencia, o un conjunto de imágenes de referencia, o se pueden incluir en el conjunto de parámetros de predicción para el bloque correspondiente. La diferencia entre los bloques originales (o los bloques correspondientes de matrices de muestras) y sus señales de predicción, también denominada señal residual, generalmente se transforma y cuantifica. A menudo, se aplica una transformación bidimensional a la señal residual (o las matrices de muestras correspondientes para el bloque residual). Para la codificación de transformaciones, los bloques (o los bloques correspondientes de matrices de muestras), para los cuales se ha usado un conjunto particular de parámetros de predicción, se pueden dividir adicionalmente antes de aplicar la transformación. Los bloques de transformación pueden ser iguales o más pequeños que los bloques que se usan para la predicción. También es posible que un bloque de transformación incluya más de uno de los bloques que se usan para la predicción. Diferentes bloques de

transformación pueden tener diferentes tamaños y los bloques de transformación pueden representar bloques cuadráticos o rectangulares. En el ejemplo anterior para las Figs. 1-5, se ha observado que es posible que los nodos de hoja de la primera subdivisión, es decir, los bloques de codificación 40, se puedan particionar adicionalmente de forma paralela en la partición que define la granularidad de los parámetros de codificación, por un lado, y los bloques de transformación sobre los cuales se aplica la transformación bidimensional individualmente, por otro lado. Después de la transformación, se cuantifican los coeficientes de transformación resultantes y se obtienen los llamados niveles de coeficiente de transformación. Los niveles del coeficiente de transformación, así como los parámetros de predicción y, si están presentes, la información de subdivisión son codificados por entropía.

- En los estándares de codificación de imagen y vídeo de última generación, las posibilidades de subdividir una imagen (o un grupo de planos) en bloques proporcionados por la sintaxis son muy limitadas. Por lo general, solo se puede especificar si y (potencialmente cómo) un bloque de un tamaño predefinido puede subdividirse en bloques más pequeños. Como ejemplo, el tamaño de bloque más grande en H.264 es 16x16. Los bloques de 16x16 también se conocen como macrobloques y cada imagen se particiona en macrobloques en una primera etapa. Para cada macrobloque de 16x16, se puede señalar si está codificado como bloque de 16x16, o como dos bloques de 16x8, o como dos bloques de 8x16, o como cuatro bloques de 8x8. Si un bloque de 16x16 se subdivide en cuatro bloques de 8x8, cada uno de estos bloques de 8x8 puede codificarse como un bloque de 8x8, o como dos bloques de 8x4, o como dos bloques 4x8, o como cuatro bloques de 4x4. El pequeño conjunto de posibilidades para especificar la partición en bloques en estándares de codificación de imagen y vídeo de última generación tiene la ventaja de que la velocidad de información conexa para señalar la información de subdivisión puede mantenerse pequeña, pero tiene la desventaja de que la velocidad binaria requerida para transmitir los parámetros de predicción para los bloques puede llegar a ser significativa como se explica a continuación. La velocidad de información conexa para señalar la información de predicción generalmente representa una cantidad significativa de la velocidad binaria general para un bloque. Y la eficiencia de codificación podría aumentarse cuando esta información conexa se reduce, lo que, por ejemplo, podría lograrse mediante el uso de tamaños de bloque más grandes. También es posible aumentar el conjunto de patrones de partición soportados en comparación con H.264. Por ejemplo, los patrones de partición representados en la Fig. 6 pueden proporcionarse para bloques cuadrados de todos los tamaños (o tamaños seleccionados). Las imágenes reales o imágenes de una secuencia de vídeo consisten en objetos de forma arbitraria con propiedades específicas. Como ejemplo, dichos objetos o partes de los objetos se caracterizan por una textura única o un movimiento único. Y generalmente, el mismo conjunto de parámetros de predicción se puede aplicar para tal objeto o parte de un objeto. Pero los límites de objeto generalmente no coinciden con los límites de bloque posibles para bloques de predicción grandes (por ejemplo, macrobloques de 16x16 en H.264). Un codificador generalmente determina la subdivisión (entre el conjunto limitado de posibilidades) que da como resultado el mínimo de una medida de coste de distorsión de velocidad particular. Para objetos de forma arbitraria, esto puede dar como resultado un gran número de bloques pequeños. Esta afirmación también sigue siendo cierta cuando se proporcionan más patrones de partición (como se mencionó anteriormente). Cabe señalar que la cantidad de patrones de partición no debería ser demasiado grande, ya que entonces se requiere mucha información conexa y/o complejidad de codificador/decodificador para señalar y procesar estos patrones. Por lo tanto, los objetos con forma arbitraria a menudo dan como resultado un gran número de bloques pequeños debido a la partición. Y dado que cada uno de estos bloques pequeños está asociado con un conjunto de parámetros de predicción, que deben transmitirse, la velocidad de información conexa puede convertirse en una parte significativa de la velocidad binaria general. Pero dado que varios de los bloques pequeños todavía representan áreas del mismo objeto o parte de un objeto, los parámetros de predicción para un número de bloques obtenidos son iguales o muy similares. Intuitivamente, la eficiencia de codificación podría aumentarse cuando la sintaxis se extiende de manera que no solo permita subdividir un bloque, sino también fusionar dos o más de los bloques que se obtienen después de la subdivisión. Como resultado, se obtendría un grupo de bloques que se codifican con los mismos parámetros de predicción. Los parámetros de predicción para tal grupo de bloques deben codificarse solo una vez. En los ejemplos anteriores de las Figs. 1-5, por ejemplo, los parámetros de codificación para el reloj actual 40 no se transmiten siempre que tenga lugar la fusión, es decir, que el conjunto reducido de candidatos no desaparezca. Es decir, el codificador no transmite los parámetros de codificación asociados con el bloque actual, y el decodificador no espera que el flujo de bits 30 contenga parámetros de codificación para el bloque actual 40. Más bien, de acuerdo con sus realizaciones específicas, se puede transmitir simplemente información de refinamiento para el bloque actual fusionado 40. Como una determinación de un conjunto de candidatos y la reducción del mismo, así como la fusión y así sucesivamente, también se realiza para los otros bloques de codificación 40 de la imagen 20. Los bloques de codificación de alguna manera forman grupos de bloques de codificación a lo largo de una cadena de codificación, donde los parámetros de codificación para estos grupos se transmiten dentro del flujo de bits en su totalidad simplemente una vez.

- Si la velocidad binaria que se ahorra al reducir el número de parámetros de predicción codificados es mayor que la velocidad binaria que se gasta adicionalmente para codificar la información de fusión, la fusión descrita da como resultado una mayor eficiencia de codificación. Cabe mencionar además que la extensión de sintaxis descrita (para la fusión) proporciona al codificador libertad adicional al seleccionar la partición de una imagen o grupo de planos en bloques. El codificador no está restringido a hacer la subdivisión primero y, a continuación, verificar si algunos de los bloques resultantes tienen el mismo conjunto de parámetros de predicción. Como una alternativa simple, el codificador podría determinar primero la subdivisión como en las técnicas de codificación de última generación. Y, a continuación,

podría verificar para cada bloque, si una fusión con uno de sus bloques vecinos (o el grupo de bloques ya determinado asociado) reduce una medida de coste de distorsión de velocidad. En este caso, los parámetros de predicción asociados con el nuevo grupo de bloques pueden volver a estimarse (por ejemplo, realizando una nueva búsqueda de movimiento) o los parámetros de predicción que ya se han determinado para el bloque actual y el bloque o grupo de bloques vecinos podrían evaluarse para el nuevo grupo de bloques. Un codificador también puede verificar directamente (un subconjunto de) los patrones que se proporcionan mediante una combinación de división y fusión; es decir, la estimación del movimiento y la decisión de modo se pueden hacer con las formas resultantes como ya se mencionó anteriormente. La información de fusión se puede señalar en bloque. Efectivamente, la fusión también podría interpretarse como inferencia de los parámetros de predicción para un bloque actual, donde los parámetros de predicción inferidos se establecen igual a los parámetros de predicción de uno de los bloques vecinos.

En este caso, cabe señalar que la combinación de diferentes patrones de partición e información de fusión puede dar como resultado las mismas formas (que se asocian con los mismos parámetros). Esto es claramente subóptimo, ya que el mismo mensaje se puede transmitir con diferentes combinaciones de palabras de código. Con el fin de evitar (o reducir) este inconveniente, las realizaciones de la presente invención describen un concepto, que prohíbe que la misma forma (asociada con un conjunto particular de parámetros) pueda señalarse mediante diferentes elementos de sintaxis de partición y fusión. Por lo tanto, para todos los bloques de un bloque previamente subdividido, excepto el primero en orden de codificación - se verifica en codificadores y decodificadores como 10 y 50, para todos los candidatos a fusión si una fusión daría como resultado un patrón que pudiera ser señalado por una partición sin información de fusión. Todos los bloques candidatos para los que esto es cierto se eliminan del conjunto de candidatos de fusión y la información de fusión transmitida se adapta al conjunto de candidatos resultante. Si no queda ningún candidato, no se transmite información de fusión; si queda un candidato, se transmite solo una bandera que especifica si el bloque se fusiona o no, etc. Para una ilustración adicional de este concepto, a continuación se describe una realización preferida. La ventaja de las realizaciones descritas con respecto a un concepto en el que solo se permite la partición es que se proporciona una libertad mucho mayor para señalar la partición de una imagen en partes que están asociadas con los mismos parámetros (por ejemplo, para especificar la señal de predicción). La ventaja en comparación con una estrategia en el que solo se proporciona un patrón de partición (por ejemplo, una subdivisión en cuatro bloques del mismo tamaño) en combinación con la estrategia de fusión es que los patrones usados a menudo (como, por ejemplo, formas rectangulares de diferentes tamaños) pueden señalarse mediante una palabra de código corta en lugar de varias banderas de subdivisión y fusión.

Los estándares de codificación de vídeo de última generación como H.264 también contienen modos intercódigos particulares llamados modo de SALTO y DIRECTO, en los que los parámetros que especifican la predicción se infieren completamente de bloques espaciales y/o temporalmente vecinos. La diferencia entre SALTO y DIRECTO es que el modo de SALTO indica además que no se transmite ninguna señal residual. En varias mejoras propuestas del modo de SALTO/DIRECTO, en lugar de un solo candidato (como en H.264), se infiere una lista de posibles candidatos de un vecindario espacial y/o temporal del bloque actual. Los posibles candidatos pueden incluir el bloque en la parte superior del bloque actual, el bloque a la izquierda del bloque actual, el bloque en la parte superior izquierda del bloque actual, el bloque en la parte superior derecha del bloque actual, el predictor de mediana de varios de estos candidatos, el bloque cúbicado en una o más tramas de referencia anteriores (o cualquier otro bloque ya codificado, o una combinación obtenida de bloques ya codificados). Para una combinación con el modo de fusión, debería garantizarse que tanto el modo de SALTO/DIRECTO como el modo de fusión no deberían incluir los mismos candidatos. Esto se puede lograr mediante diferentes configuraciones como se mencionó anteriormente. La ventaja de las combinaciones descritas es que se proporcionan múltiples opciones para señalar la reutilización de parámetros ya transmitidos (por ejemplo, para especificar la predicción) para diferentes tamaños de bloque.

Una ventaja de las realizaciones de la presente invención es reducir la velocidad binaria que se requiere para transmitir los parámetros de predicción mediante la fusión de bloques vecinos en un grupo de bloques, donde cada grupo de bloques está asociado con un conjunto único de parámetros de codificación, por ejemplo, parámetros de predicción o parámetros de codificación residuales. La información de fusión se señala dentro del flujo de bits (además de la información de subdivisión, si está presente). En combinación con diferentes patrones de división y modos de SALTO/DIRECTO, se puede garantizar que el modo de SALTO/DIRECTO y ninguno de los patrones proporcionados se «simula» mediante el envío de la información de fusión correspondiente. La ventaja de las realizaciones de la presente invención es una mayor eficiencia de codificación resultante de una velocidad de información conexa disminuida para los parámetros de codificación. Las realizaciones de la presente invención son aplicables en aplicaciones de codificación de imagen y vídeo, en las que conjuntos de muestras están asociados con parámetros de codificación o predicción particulares. El procedimiento de fusión descrito actualmente también se extiende a una tercera dimensión o más dimensiones. Por ejemplo, un grupo de bloques en varias imágenes de vídeo podría fusionarse en un grupo de bloques. También podría aplicarse a la compresión 4D en codificación de campo de luz. Por otro lado, también se puede usar para la compresión en señales de 1D, donde la señal de 1D se particiona y se fusionan particiones dadas.

Las realizaciones de la presente invención también se refieren a un procedimiento para reducir la velocidad de información conexa en aplicaciones de codificación de imagen y vídeo. En aplicaciones de codificación de imagen y

vídeo, conjuntos particulares de muestras (que pueden representar bloques rectangulares o cuadráticos o regiones de forma arbitraria o cualquier otra colección de muestras) generalmente se asocian con un conjunto particular de parámetros de codificación. Para cada uno de estos conjuntos de muestras, los parámetros de codificación se incluyen en el flujo de bits. Los parámetros de codificación pueden representar parámetros de predicción, que especifican cómo se predice el conjunto correspondiente de muestras usando muestras ya codificadas. La partición de las matrices de muestras de una imagen en conjuntos de muestras puede fijarse mediante la sintaxis o puede señalarse mediante la información de subdivisión correspondiente dentro del flujo de bits. Se pueden permitir múltiples patrones de partición para un bloque. Los parámetros de codificación para los conjuntos de muestras se transmiten en un orden predefinido, que es dado por la sintaxis. Las realizaciones de la presente invención también representan un procedimiento mediante el cual se puede señalar para un conjunto actual de muestras que se fusiona (por ejemplo, con fines de predicción) con uno o más conjuntos de muestras adicionales en un grupo de conjuntos de muestras. Por lo tanto, el posible conjunto de valores para la información de fusión correspondiente se adapta al patrón de partición empleado, de manera que los patrones de partición particulares no pueden representarse mediante una combinación de otros patrones de partición y los datos de fusión correspondientes. Los parámetros de codificación para un grupo de conjuntos de muestras deben transmitirse solo una vez. En una realización particular, los parámetros de codificación de un conjunto de muestras actual no se transmiten si el conjunto de muestras actual se fusiona con un conjunto de muestras (o un grupo de conjuntos de muestras) para el cual los parámetros de codificación ya se han transmitido; en su lugar, los parámetros de codificación para el conjunto actual de muestras se establecen igual a los parámetros de codificación del conjunto de muestras (o grupo de conjuntos de muestras) con el que se fusiona el conjunto actual de muestras. Como estrategia alternativa, se puede transmitir un refinamiento adicional para uno o más de los parámetros de codificación para un conjunto de muestras actual; el refinamiento se puede aplicar a todos los conjuntos de muestras de un grupo o solo al conjunto de muestras para el que se transmite.

En una realización preferida, para cada conjunto de muestras, el conjunto de todos los conjuntos de muestras previamente codificados se denomina el «conjunto de conjuntos de muestras causales». Los conjuntos de muestras que se pueden usar para la fusión con un conjunto actual de muestras se denominan el "conjunto de conjuntos de muestras candidatos" y siempre es un subconjunto del "conjunto de conjuntos de muestras causales". La forma en que se forma este subconjunto puede ser conocida por el decodificador o puede especificarse dentro del flujo de bits. En cualquier caso, el codificador 10 y el decodificador 80 determinan que el conjunto de candidatos se reduzca. Si un conjunto actual particular de muestras está codificado y su conjunto de conjuntos de muestras candidatos no está vacío, se señala (o deriva) si el conjunto actual de muestras se fusiona con un conjunto de muestras de este conjunto de conjuntos de muestras candidatos y, en caso afirmativo, con cuál de ellos (si existe más de un candidato). De lo contrario, la fusión no se puede usar para este bloque. Los bloques candidatos para los cuales una fusión daría como resultado una forma que también podría especificarse directamente mediante un patrón de partición se excluyen del conjunto de candidatos, con el fin de evitar que la misma forma se pueda representar mediante diferentes combinaciones de información de partición y datos de fusión. Es decir, el conjunto de candidatos se reduce, mediante la eliminación de candidatos respectivos como se describió anteriormente con respecto a las Figs. 1-5.

En una realización preferida, un número del conjunto de conjuntos de muestras candidatos es cero o más conjuntos de muestras que contienen al menos un número particular distinto de cero de muestras (que puede ser uno o dos o incluso más) que representan vecinos espaciales directos de cualquier muestra dentro del conjunto actual de muestras. En otra realización preferida de la invención, el conjunto de conjuntos de muestras candidatos puede incluir adicionalmente (o exclusivamente) conjuntos de muestras que contienen un número particular distinto de cero de muestras (que puede ser uno o dos o incluso más) que tienen la misma ubicación espacial, es decir, están comprendidos tanto por los conjuntos de muestras candidatos como por el conjunto de muestras actual sometido actualmente a fusión, pero están contenidos en una imagen diferente. En otra realización preferida de la invención, el conjunto de conjuntos de muestras candidatos se puede derivar de datos procesados previamente dentro de la imagen actual o en otras imágenes. El procedimiento de derivación puede incluir información direccional espacial, tal como coeficientes de transformación, asociados con una dirección particular y gradientes de imagen de la imagen actual o puede incluir información direccional temporal tal como representaciones de movimiento vecinas. A partir de tales datos disponibles en el receptor y otros datos e información conexa (si están presentes), se puede derivar el conjunto de conjuntos de muestras candidatos. La eliminación de candidatos (del conjunto de candidatos original) que daría como resultado la misma forma que podría estar representada por un patrón de partición particular se deriva de la misma manera en el codificador y decodificador, de modo que el codificador y decodificador derivan el conjunto de candidatos finales para fusión exactamente de la misma manera.

En una realización preferida, los conjuntos de muestras considerados son bloques rectangulares o cuadráticos. Entonces, los conjuntos fusionados de muestras representan una colección de bloques rectangulares y/o cuadráticos. En otra realización preferida de la invención, los conjuntos considerados de muestras son regiones de imagen de forma arbitraria y los conjuntos fusionados de muestras representan una colección de regiones de imagen de forma arbitraria.

En una realización preferida, se transmiten uno o más elementos de sintaxis para cada conjunto de muestras, que especifican si el conjunto de muestras se fusiona con otro conjunto de muestras (que puede ser parte de un grupo ya

fusionado de conjuntos de muestras) y cuál del conjunto de conjuntos de muestras candidatos se emplea para la fusión. Sin embargo, el elemento de sintaxis no se transmite si el conjunto de candidatos está vacío (por ejemplo, debido a una eliminación de los candidatos que produciría una partición que podría ser señalado por diferentes patrones de partición sin fusión).

5

En una realización preferida, se transmiten uno o dos elementos de sintaxis para especificar la información de fusión. El primer elemento de sintaxis especifica si el conjunto actual de muestras se fusiona con otro conjunto de muestras. El segundo elemento de sintaxis, que solo se transmite si el primer elemento de sintaxis especifica que el conjunto actual de muestras se fusiona con otro conjunto de muestras, especifica cuál de los conjuntos de conjuntos de muestras candidatos se emplea para la fusión. En una realización preferida, el primer elemento de sintaxis solo se transmite si un conjunto derivado de conjuntos de muestras candidatos no está vacío (después de la eliminación potencial de los candidatos que producirían una partición que podría señalarse mediante un patrón de partición diferente sin fusión). En otra realización preferida, el segundo elemento de sintaxis solo se transmite si un conjunto derivado de conjuntos de muestras candidatos contiene más de un conjunto de muestras. En otra realización preferida de la invención, el segundo elemento de sintaxis solo se transmite si al menos dos conjuntos de muestras de un conjunto derivado de conjuntos de muestras candidatos están asociados con parámetros de codificación diferentes.

10

15

En una realización preferida de la invención, la información de fusión para un conjunto de muestras se codifica antes de los parámetros de predicción (o, más generalmente, los parámetros de codificación particulares que están asociados con los conjuntos de muestras). Los parámetros de predicción o codificación solo se transmiten si la información de fusión señala que el conjunto actual de muestras no se fusiona con otro conjunto de muestras.

20

En otra realización preferida, la información de fusión para un conjunto de muestras se codifica después de que se ha transmitido un subconjunto de los parámetros de predicción (o, más generalmente, los parámetros de codificación particulares que están asociados con los conjuntos de muestras). El subconjunto de parámetros de predicción puede consistir en uno o más índices de imagen de referencia o uno o más componentes de un vector de parámetro de movimiento o un índice de referencia y uno o más componentes de un vector de parámetro de movimiento, etc. El subconjunto ya transmitido de parámetros de predicción o codificación se puede usar para derivar un conjunto (reducido) de conjuntos de muestras candidatos. Como ejemplo, se puede calcular una medida de diferencia entre los parámetros de predicción o codificación ya codificados y los parámetros de predicción o codificación correspondientes de un conjunto original de conjuntos de muestras candidatos. Y solo aquellos conjuntos de muestras, para los cuales la medida de diferencia calculada es menor o igual a un umbral predefinido o derivado, se incluyen en el conjunto final (reducido) de conjuntos de muestras candidatos. El umbral puede derivarse en función de las medidas de diferencia calculadas. O como otro ejemplo, solo se seleccionan aquellos conjuntos de muestras para los cuales se minimiza la medida de diferencia. O solo se selecciona un conjunto de muestras en función de la medida de diferencia. En este último caso, la información de fusión puede reducirse de tal manera que solo especifique si el conjunto actual de muestras se fusiona con el único conjunto de candidatos de muestras.

25

30

35

Se describen las siguientes realizaciones preferidas para conjuntos de muestras que representan bloques rectangulares y cuadráticos, pero se puede extender a regiones de forma arbitraria u otras colecciones de muestras de una manera sencilla.

40

1. Derivación del conjunto inicial de bloques candidatos

La derivación del conjunto inicial de muestras descrita en estas secciones se refiere a la derivación de un conjunto de candidatos iniciales. Algunos de todos los bloques candidatos pueden eliminarse posteriormente mediante el análisis de los parámetros asociados (por ejemplo, información de predicción) y la eliminación de aquellos bloques candidatos para los cuales una fusión daría como resultado una partición final que también podría obtenerse mediante el uso de otros patrones de partición. Este procedimiento se describe en la siguiente subsección.

45

50

En una realización preferida, el conjunto de bloques candidatos iniciales se forma de la siguiente manera. Partiendo de la posición de muestra superior izquierda del bloque actual, se deriva su posición de muestra vecina izquierda y su posición de muestra vecina superior. El conjunto de bloques candidatos iniciales solo puede tener hasta dos elementos, concretamente aquellos bloques del conjunto de bloques causales que contienen una de las dos posiciones de muestra. Por lo tanto, el conjunto de bloques candidatos iniciales solo puede tener los dos bloques directamente vecinos de la posición de muestra superior izquierda del bloque actual como sus elementos.

55

En otra realización preferida de la invención, el conjunto de bloques candidatos iniciales viene dado por todos los bloques que han sido codificados antes del bloque actual y contienen una o más muestras que representan vecinos espaciales directos (los vecinos espaciales directos pueden restringirse a vecinos izquierdos directos y/o vecinos superiores directos y/o vecinos derechos directos y/o vecinos inferiores directos) de cualquier muestra del bloque actual. En otra realización preferida de la invención, el conjunto de bloques candidatos iniciales incluye adicionalmente (o exclusivamente) bloques que contienen una o más muestras que están ubicadas en la misma posición que cualquiera de las muestras del bloque actual pero están contenidas en una imagen diferente (ya codificada). En otra

60

realización preferida de la invención, el conjunto de candidatos iniciales de bloques representa un subconjunto de los conjuntos de bloques (vecinos) descritos anteriormente. El subconjunto de bloques candidatos puede ser fijo, señalado o derivado. La derivación del subconjunto de bloques candidatos puede considerar las decisiones tomadas para otros bloques en la imagen o en otras imágenes. Como ejemplo, los bloques que están asociados con los mismos parámetros de codificación (o muy similares) que otros bloques candidatos podrían no incluirse en el conjunto de candidatos iniciales de bloques.

En una realización preferida de la invención, conjunto de candidatos iniciales de bloques se deriva como para una de las realizaciones descritas anteriormente, pero con la siguiente restricción: Solo los bloques que usan predicción de movimiento compensado (predicción inter) pueden ser elementos del conjunto de bloques candidatos. Es decir, los bloques intracodificados no se incluyen en el conjunto de candidatos (inicial).

Como ya se ha indicado anteriormente, es posible ampliar la lista de candidatos con candidatos adicionales para la fusión en bloque, como por ejemplo mediante la combinación de candidatos a fusión bipredictivos combinados, candidatos a fusión bipredictivos no escalados y un vector de movimiento cero.

La derivación del conjunto inicial de bloques candidatos se realiza tanto por el codificador como por el decodificador de la misma manera.

2. Derivación del conjunto final de bloques candidatos.

Después de derivar el conjunto de candidatos iniciales, se analizan los parámetros asociados de los bloques candidatos dentro del conjunto de candidatos iniciales y se eliminan los candidatos a fusión para los cuales una fusión daría como resultado una partición que podría representarse mediante el uso de un patrón de partición diferente. Si las matrices de muestras que se pueden fusionar son de diferente forma y/o tamaño, pueden existir particionamientos idénticos que se pueden describir mediante al menos dos palabras de código diferentes. Por ejemplo, si el codificador decide dividir una matriz de muestras en dos matrices de muestras, esta división se revertiría fusionando las dos matrices de muestras. Para evitar tales descripciones redundantes, el conjunto de bloques candidatos para la fusión se restringe en función de las formas de bloque y divisiones particulares que se permiten. Por otro lado, las formas permitidas de matrices de muestras se pueden restringir dependiendo de las listas de candidatos particulares usadas para la fusión. Las dos instalaciones de división y fusión tienen que diseñarse juntas de modo que, en combinación con las dos, se eviten descripciones redundantes.

En una realización preferida de la invención, el conjunto de modos de división (o modos de partición) representados en la Fig. 6 son soportados para bloques cuadrados. Si un bloque cuadrado de un tamaño particular se divide en cuatro bloques cuadrados más pequeños del mismo tamaño (patrón inferior izquierdo en la Fig. 6), el mismo conjunto de patrones de partición se puede aplicar a los cuatro bloques cuadrados resultantes de modo que se pueda especificar una partición jerárquico.

Después de derivar el conjunto de bloques candidatos iniciales, la reducción de las listas de candidatos se realiza de la siguiente manera.

Si el bloque actual no se particiona más (patrón superior izquierdo en la Fig. 6), la lista de candidatos iniciales no se reduce. Es decir, todos los candidatos iniciales representan los candidatos finales para la fusión.

Si el bloque actual se particiona en exactamente dos bloques de tamaño arbitrario, uno de estos dos bloques se codifica antes que el otro, lo que se determina mediante la sintaxis. Para el primer bloque codificado, el conjunto de candidatos iniciales no se reduce. Pero para el segundo bloque codificado, todos los bloques candidatos que tienen los mismos parámetros asociados que el primer bloque se eliminan del conjunto de candidatos (esto incluye el primer bloque codificado).

Si un bloque se particiona en cuatro bloques cuadrados del mismo tamaño, la lista de candidatos iniciales de los tres primeros bloques (en orden de codificación) no se reduce. Todos los bloques de la lista de candidatos iniciales también están presentes en la lista de candidatos finales. Pero para el cuarto (último) bloque en orden de codificación, se aplica lo siguiente:

Si los bloques que están en una fila diferente (en el esquema de partición como se ilustra en la parte inferior izquierda de la Fig. 6) que el bloque actual tienen los mismos parámetros asociados (por ejemplo, parámetros de movimiento), todos los candidatos que tienen los mismos parámetros de movimiento que el bloque ya codificado en la misma fila que el bloque actual se eliminan del conjunto de candidatos (esto incluye el bloque en la misma fila).

Si los bloques que están en una columna diferente (en el esquema de partición como se ilustra en la parte inferior izquierda de la Fig. 6) que el bloque actual tienen los mismos parámetros asociados (por ejemplo, parámetros de movimiento), todos los candidatos que tienen los mismos parámetros de movimiento que el bloque ya codificado en la

misma columna que el bloque actual se eliminan del conjunto de candidatos. (esto incluye el bloque en la misma columna)

- 5 En una variación de baja complejidad de la realización (utilizando los patrones de partición de la Fig. 6), la reducción de las listas de candidatos se realiza de la siguiente manera.

Si el bloque actual no se particiona más (patrón superior izquierdo en la Fig. 6), la lista de candidatos iniciales no se reduce. Es decir, todos los candidatos iniciales representan los candidatos finales para la fusión.

- 10 Si el bloque actual se particiona en exactamente dos bloques de tamaño arbitrario, uno de estos dos bloques se codifica antes que el otro, lo que se determina mediante la sintaxis. Para el primer bloque codificado, el conjunto de candidatos iniciales no se reduce. Pero para el segundo bloque codificado, el primer bloque codificado del patrón de partición se elimina del conjunto de candidatos.

- 15 Si un bloque se particiona en cuatro bloques cuadrados del mismo tamaño, la lista de candidatos iniciales de los tres primeros bloques (en orden de codificación) no se reduce. Todos los bloques de la lista de candidatos iniciales también están presentes en la lista de candidatos finales. Pero para el cuarto (último) bloque en orden de codificación, se aplica lo siguiente:

- 20 Si para el bloque en la otra fila (que no sea el bloque actual) que se codifica más tarde, la información de fusión indica que se fusiona con el primer bloque codificado de esa fila, el bloque en la misma fila que el bloque actual se elimina del conjunto de candidatos.

- 25 Si para el bloque en la otra columna (que no sea el bloque actual) que se codifica más tarde, la información de fusión señala que se fusiona con el primer bloque codificado de esa columna, el bloque en la misma columna que el bloque actual se elimina del conjunto de candidatos.

- 30 En otra realización preferida, se soportan los mismos patrones de partición que se representan en la Fig. 6, pero sin los patrones que particionan el bloque cuadrado en dos bloques rectangulares del mismo tamaño. La reducción de la lista de candidatos procede como se describe en cualquiera de las realizaciones descritas anteriormente, con la excepción del patrón que divide el bloque en cuatro bloques cuadrados. Aquí, todos los candidatos iniciales están permitidos para todos los sub-bloques o solo la lista de candidatos del último sub-bloque codificado está restringida de la siguiente manera. Si los tres bloques previamente codificados están asociados con los mismos parámetros, todos los candidatos que están asociados con estos parámetros se eliminan de la lista de candidatos. En una versión de baja complejidad, el último sub-bloque codificado no se puede fusionar con ninguno de los tres sub-bloques previamente codificados si estos tres sub-bloques se han fusionado juntos.

- 40 En otra realización preferida, se soporta un conjunto diferente de patrones de partición para un bloque (o cualquier otra forma de conjunto de matrices de muestras). Para conjuntos de matrices de muestras que no están particionados, todos los candidatos de las listas de candidatos iniciales se pueden usar para la fusión. Si una matriz de muestras está particionada en exactamente dos matrices de muestras, para las matrices de muestras que están primero en orden de codificación, todos los candidatos del conjunto de candidatos iniciales se insertan en el conjunto de candidatos finales. Para la segunda matriz de muestras en orden de codificación, se eliminan todos los candidatos que tienen los mismos parámetros asociados que la primera matriz de muestras. O en una variación de baja complejidad, solo se elimina la primera matriz de muestras del conjunto de candidatos. Para patrones de partición que dividen una matriz de muestras en más de 2 matrices de muestras, la eliminación de candidatos depende de si se puede simular otro patrón de partición con el patrón de partición actual y la información de fusión correspondiente. El procedimiento de eliminación de candidatos sigue el concepto explícitamente descrito anteriormente, pero considera los patrones de candidatos realmente soportados.

- 50 En otra realización preferida, si el modo de SALTO/DIRECTO es soportado para un bloque en particular, los candidatos a fusión que también son candidatos presentes para los modos de SALTO/DIRECTO se eliminan de la lista de candidatos. Esta eliminación puede reemplazar las eliminaciones de bloques candidatos descritas anteriormente o usarse junto con las eliminaciones de bloques candidatos descritas anteriormente.

55 3. Combinación con modos de SALTO/DIRECTO

- 60 Los modos de SALTO/DIRECTO pueden ser soportados para todos o solo para un tamaño de bloque y/o forma de bloque particular. Se usa un conjunto de bloques candidatos para los modos de SALTO/DIRECTO. La diferencia entre SALTO y DIRECTO es si se envía o no información residual. Se infiere que los parámetros (por ejemplo, para la predicción) de SALTO y DIRECTO son iguales a cualquiera de los candidatos correspondientes. El candidato se elige transmitiendo un índice a la lista de candidatos.

En una realización preferida, la lista de candidatos para SALTO/DIRECTO puede contener diferentes candidatos. En

la Fig. 8 se ilustra un ejemplo. La lista de candidatos puede incluir los siguientes candidatos (el bloque actual se indica por Xi):

- 5 Mediana (entre Izquierda, Arriba, Esquina)
- Bloque izquierdo (Li)
- Bloque de arriba (Ai)
- 10 Bloques de esquina (En orden: Arriba a la derecha (Ci1), Abajo a la izquierda (Ci2), Arriba a la izquierda (Ci3))
- Bloque coubicado en una imagen diferente, pero ya codificada
- 15 En una realización preferida, los candidatos para la fusión incluyen Li (bloque izquierdo) y Ai (bloque de arriba). La elección de estos candidatos para la fusión requiere una pequeña cantidad de información conexa para señalar con qué bloque se fusiona el bloque actual.

Para describir las siguientes realizaciones se usa la siguiente notación:

- 20 set_mvp_ori es un conjunto de candidatos usados para el modo de SALTO/DIRECTO. Este conjunto está compuesto por {Mediana, Izquierda, Arriba, Esquina, Coubicado}, donde Mediana es la mediana (valor medio en un conjunto ordenado de Izquierda, Arriba y Esquina), y coubicado está dado por la trama de referencia más cercana y se escala de acuerdo con la distancia temporal.
- 25 set_mvp_comb es un conjunto de candidatos usados para el modo de SALTO/DIRECTO en combinación con el procedimiento de fusión de bloques.

- 30 Para la realización preferida, la combinación entre el modo de SALTO/DIRECTO y el modo de fusión de bloques se puede procesar con el conjunto original de candidatos. Esto significa que el modo de SALTO/DIRECTO tiene el mismo conjunto de candidatos que cuando se activa solo. El interés de combinar estos dos modos proviene de su complementariedad en la señalización de la información conexa en intertrama. A pesar del hecho de que estos dos modos están usando información de los vecinos con el fin de mejorar la señalización del bloque actual, la fusión de bloques está procesando solo los vecinos de la izquierda y de arriba y el modo de SALTO/DIRECTO está procesando hasta 5 candidatos. La principal complementariedad reside en la diferente estrategia de procesamiento de la información del vecino. El procedimiento de fusión de bloques mantiene el conjunto completo de información de sus vecinos para todas las listas de referencia. Esto significa que la fusión de bloques mantiene la información conexa completa de estos vecinos y no solo sus vectores de movimiento por lista de referencia, mientras que el modo de SALTO/DIRECTO procesa por separado los parámetros de predicción para cada lista de referencia y transmite un índice a una lista de candidatos para cada lista de referencia. Es decir, para las imágenes bipredichas, se transmiten dos índices para señalar un candidato para la lista de referencia 0 y un candidato para la lista de referencia 1.
- 40

- 45 En otra realización preferida, se puede encontrar un conjunto combinado de candidatos, llamado set_mvp_comb, para el modo de SALTO/DIRECTO en combinación con el modo de fusión de bloques. Este conjunto combinado forma parte del conjunto original (set_mvp_ori) y permite una reducción de la señalización para el modo de SALTO/DIRECTO, debido a la reducción de la lista de candidatos: set_mvp_comb. Los candidatos que deberían eliminarse de la lista original (set_mvp_ori) son aquellos que podrían ser redundantes con el procedimiento de fusión de bloques o que no se usan a menudo.

- 50 En otra realización preferida, la combinación entre el modo de SALTO/DIRECTO y el procedimiento de fusión de bloques se puede procesar con el conjunto combinado de candidatos (set_mvp_comb), que es el conjunto original (set_mvp_ori) sin la mediana. Debido a una baja eficiencia observada para la Mediana para el modo de SALTO/DIRECTO, su reducción de la lista original, trae una mejora en la eficiencia de codificación.

- 55 En otra realización preferida, la combinación del modo de SALTO/DIRECTO y la fusión de bloques se puede procesar con el conjunto combinado de candidatos (set_mvp_comb), que es el conjunto original (set_mvp_ori) solo con la Esquina y/o con el Coubicado como candidatos.

- 60 En otra realización preferida, la combinación del modo de SALTO/DIRECTO y el procedimiento de fusión de bloques se puede procesar con el conjunto combinado de candidatos, que es el set_mvp_ori con solo la Esquina y el Coubicado como candidatos. A pesar de la complementariedad entre el modo de SALTO/DIRECTO y la fusión de bloques, como ya se ha mencionado, los candidatos que deberían eliminarse de la lista son los que podrían ser redundantes con los candidatos del procedimiento de fusión de bloques. Estos candidatos son Izquierda y Arriba. El conjunto combinado de candidatos (set_mvp_comb) se ha reducido a sólo dos candidatos: Esquina y Coubicado. El modo salto/DIRECTO que usa este conjunto de candidatos set_mvp_comb, combinado con el proceso de fusión de bloques, proporciona

un alto aumento en la eficiencia de la señalización de la información secundaria en inter tramas. En esta realización, el modo SALTO/DIRECTO y el modo de fusión no comparten ningún bloque candidato.

- En realizaciones adicionales, se puede usar una combinación ligeramente diferente del modo de SALTO/DIRECTO y de fusión. Es posible habilitar el modo de SALTO/DIRECTO (por ejemplo, con más candidatos que el modo de fusión) solo para bloques particulares (por ejemplo, con un tamaño mayor que un tamaño especificado, o solo para bloques cuadrados, etc.) y no soportar el modo de fusión para estos bloques. O se puede eliminar el modo de SALTO/DIRECTO y todos los candidatos (incluidos los parámetros que representan una combinación de parámetros para los bloques vecinos espaciales/temporales) se añaden al modo de fusión como candidatos. Esta opción se ha descrito en las Figs. 1 a 5. El conjunto de candidatos aumentado solo se podría usar para bloques particulares (con un tamaño mayor que un tamaño mínimo dado, o bloques cuadrados, etc.), donde para otros bloques se usa un conjunto de candidatos reducido. O como una variante adicional, el modo de fusión se usa con un conjunto de candidatos reducido (por ejemplo, solo el vecino superior e izquierdo) y se usan candidatos adicionales (por ejemplo, el vecino superior izquierdo, el bloque cúbico, etc.) para el modo de SALTO/DIRECTO. También en tales configuraciones, los modos de SALTO/DIRECTO solo se pueden permitir para bloques particulares (con un tamaño mayor que un tamaño mínimo dado, o bloques cuadrados, etc.), mientras que el modo de fusión se permite para un conjunto más grande de bloques.

4. Transmisión de información de fusión

- Para la realización preferida y, en particular, para las realizaciones de las Figuras 1 a 5, se puede aplicar lo siguiente. Imaginemos que solo se consideran candidatos los dos bloques que contienen la muestra vecina izquierda y superior de la muestra superior izquierda de los bloques actuales. Si el conjunto de bloques candidatos finales (después de la eliminación de candidatos como se describió anteriormente) no está vacío, se señala una bandera llamada bandera de fusión, que especifica si el bloque actual se fusiona con cualquiera de los bloques candidatos. Si `merge_flag` es igual a 0 (para "false"), este bloque no se fusiona con uno de sus bloques candidatos y todos los parámetros de codificación se transmiten normalmente. Si `merge_flag` es igual a 1 (para "true"), se aplica lo siguiente. Si el conjunto de bloques candidatos contiene uno y solo un bloque, este bloque candidato se usa para fusión. De lo contrario, el conjunto de bloques candidatos contiene exactamente dos bloques. Si los parámetros de predicción de estos dos bloques son idénticos, estos parámetros de predicción se usan para el bloque actual. De lo contrario (los dos bloques tienen parámetros de predicción diferentes), se señala una bandera llamada `merge_left_flag`. Si `merge_left_flag` es igual a 1 (para «true»), el bloque que contiene la posición de muestra vecina izquierda de la posición de muestra superior izquierda del bloque actual se selecciona del conjunto de bloques candidatos. Si `merge_left_flag` es igual a 0 (para «false»), se selecciona el otro bloque (es decir, el vecino superior) del conjunto de bloques candidatos. Los parámetros de predicción del bloque seleccionado se usan para el bloque actual. En otra realización, se transmite un elemento de sintaxis combinado que señale el procedimiento de fusión. En otra realización, `merge_left_flag` se transmite independientemente de si los dos bloques candidatos tienen los mismos parámetros de predicción.

Cabe señalar que el elemento de sintaxis `merge_left_flag` también podría denominarse `merge_index` ya que su función es indexar el elegido entre los candidatos no eliminados.

- En otra realización preferida, se pueden incluir más de dos bloques en el conjunto de bloques candidatos. La información de fusión (es decir, si un bloque se fusiona y, en caso afirmativo, con qué bloque candidato se fusiona) se señala mediante uno o más elementos de sintaxis. En este caso, el conjunto de palabras de código depende del número de candidatos en el conjunto de candidatos finales y se selecciona de la misma manera en el codificador y el decodificador. En una realización, la información de fusión se transmite usando un elemento de sintaxis. En otra realización, un elemento de sintaxis especifica si el bloque se fusiona con cualquiera de los bloques candidatos (véase la bandera `merge_flag` descrita anteriormente). Esta bandera solo se transmite si el conjunto de bloques candidatos no está vacío. El segundo elemento de sintaxis señala cuál de los bloques candidatos se emplea para la fusión; solo se transmite si el primer elemento de sintaxis señala que el bloque actual se fusiona con uno de los bloques candidatos. En una realización preferida de la invención, el segundo elemento de sintaxis solo se transmite si el conjunto de bloques candidatos contiene más de un bloque candidato y/o si cualquiera de los bloques candidatos tiene parámetros de predicción diferentes de cualquier otro de los bloques candidatos. La sintaxis puede depender de cuántos bloques candidatos se den y/o de cómo se asocian parámetros de predicción diferentes con los bloques candidatos.

- Es posible añadir un conjunto de candidatos para la fusión de bloques como se hizo para el modo DIRECTO.

- Como se describe en otras realizaciones preferidas, el segundo elemento de sintaxis `merge_index` solo se puede transmitir si la lista de candidatos contiene más de un candidato. Esto requiere derivar la lista antes de analizar sintácticamente `merge_index`, impidiendo llevar a cabo estos dos procedimientos en paralelo. Para permitir un mayor rendimiento de análisis sintáctico y hacer que el procedimiento de análisis sintáctico sea más robusto con respecto a los errores de transmisión, es posible eliminar esta dependencia mediante el uso de una palabra de código fija para cada valor de índice y un número fijo de candidatos. Si este número no puede ser alcanzado por una selección de candidatos, es posible derivar candidatos auxiliares para completar la lista. Estos candidatos adicionales pueden incluir los llamados candidatos combinados, que se construyen a partir de parámetros de movimiento de candidatos

posiblemente diferentes ya en la lista, y vectores de movimiento cero.

En otra realización preferida, la sintaxis para señalar cuál de los bloques del conjunto de candidatos puede adaptarse simultáneamente en el codificador y el decodificador. Si, por ejemplo, se dan 3 opciones de bloques para la fusión, esas tres opciones solo están presentes en la sintaxis y se consideran para la codificación de entropía. Las probabilidades para todas las demás opciones se consideran 0 y el códec de entropía se ajusta simultáneamente en el codificador y el decodificador.

Los parámetros de predicción que se infieren como consecuencia del procedimiento de fusión pueden representar el conjunto completo de los parámetros de predicción que están asociados con un bloque o pueden representar un subconjunto de estos parámetros de predicción (por ejemplo, los parámetros de predicción para una hipótesis de un bloque para el cual se usa predicción de múltiples hipótesis).

En una realización preferida, los elementos de sintaxis relacionados con la información de fusión son codificados por entropía usando modelado de contexto. Los elementos de sintaxis pueden consistir en `merge_flag` and `merge_left_flag` descritos anteriormente.

En una realización preferida, se usa uno de cada tres modelos de contexto para codificar el `merge_flag`. El modelo de contexto usado `merge_flag_ctx` se deriva de la siguiente manera. Si el conjunto de bloques candidatos contiene dos elementos, el valor de `merge_flag_ctx` es igual a la suma de los valores de `merge_flag` de los dos bloques candidatos. Si el conjunto de bloques candidatos contiene un elemento, el valor de `merge_flag_ctx` es igual a dos veces el valor de `merge_flag` de este un bloque candidato.

En una realización preferida, `merge_left_flag` se codifica usando un único modelo de probabilidad.

Se puede usar codificación de diferentes modelos de contexto para `merge_idx` (`merge_left_flag`).

En otras realizaciones, se podrían usar diferentes modelos de contexto. Los elementos de sintaxis no binarios pueden asignarse a una secuencia de símbolos binarios (bins). Los modelos de contexto para algunos elementos de sintaxis o bins de elementos de sintaxis pueden derivar en función de elementos de sintaxis ya transmitidos de bloques vecinos o el número de bloques candidatos u otras medidas, mientras que otros elementos de sintaxis o bins de elementos de sintaxis pueden codificarse con un modelo de contexto fijo.

5. Funcionamiento del codificador

La inclusión del concepto de fusión proporciona un codificador con una mayor libertad para la creación de un flujo de bits, ya que la estrategia de fusión aumenta significativamente el número de posibilidades para seleccionar una partición para las matrices de muestras de una imagen, por supuesto, a una mayor tasa de señalización. Algunos o todos los patrones adicionales que se pueden representar mediante una combinación de subparticionamiento y fusión (por ejemplo, los patrones de la Fig. 7, cuando se soporta el patrón de partición de la Fig. 6) se pueden probar adicionalmente (usando los tamaños de bloque correspondientes para estimación de movimiento y decisión de modo) y el mejor de los patrones proporcionados mediante partición puro (Fig. 6) y mediante partición y fusión (Fig. 7) se puede seleccionar en función de una medida de distorsión de velocidad particular. Además, para cada bloque se puede probar si una fusión con cualquiera de los conjuntos de candidatos ya codificados produce una disminución de una medida de distorsión de velocidad particular y a continuación se establecen las banderas de fusión correspondientes durante el procedimiento de codificación.

En otra realización preferida, el codificador podría determinar primero la mejor subdivisión de las matrices de muestras (como en los esquemas de codificación de última generación). Y luego podría verificar para cada conjunto de muestras, si una fusión con otro conjunto de muestras u otro grupo de conjuntos de muestras reduce una medida de costo de distorsión de velocidad particular. En este caso, los parámetros de predicción asociados con el grupo fusionado de conjuntos de muestras pueden reestimarse (por ejemplo, realizando una nueva búsqueda de movimiento) o los parámetros de predicción que ya se han determinado para el conjunto de muestras actual y el conjunto de muestras candidato (o grupo de conjuntos de muestras) para la fusión podrían evaluarse para el grupo considerado de conjuntos de muestras.

En otra realización preferida, podría evaluarse una medida particular de coste de distorsión de velocidad particular para grupos de candidatos adicionales de conjuntos de muestras. Como ejemplo particular, cuando se prueban los diversos patrones de partición posibles (véase la Fig. 6, por ejemplo), parte o la totalidad del patrón que se puede representar mediante una combinación de partición y fusión (véase la Fig. 7, por ejemplo) se puede probar adicionalmente. Es decir, para todos los patrones se lleva a cabo un procedimiento específico de estimación de movimiento y decisión de modo y se selecciona el patrón que produce la medida de distorsión de velocidad más pequeña. Este procedimiento también se puede combinar con el procedimiento de baja complejidad descrito anteriormente, de modo que para los bloques resultantes se prueba adicionalmente si una fusión con bloques ya

codificados (por ejemplo, fuera de los patrones de la Fig. 6 y la Fig. 7) produce una disminución en una medida de distorsión de velocidad.

5 En otra realización preferida, el codificador prueba los diferentes patrones que se pueden representar mediante partición y fusión en un orden de prioridad y prueba tantos patrones como sea posible mediante un requisito de tiempo real dado. El orden de prioridad también puede modificarse en función de los bloques ya codificados y los patrones de partición elegidos.

10 Una forma de transferir las realizaciones descritas anteriormente a una sintaxis específica se explica a continuación con respecto a las siguientes figuras. En particular, las Figs. 9-11 muestran diferentes partes de una sintaxis que aprovecha las realizaciones descritas anteriormente. En particular, de acuerdo con la realización descrita a continuación, la imagen 20 se divide en primer lugar en bloques de árbol de codificación cuyo contenido de imagen se codifica usando la sintaxis `coding_tree` mostrada en la Fig. 9. Tal como se muestra en este documento, para `entropy_coding_mode_flag=1`, que se refiere, por ejemplo, a la codificación aritmética binaria adaptativa al contexto u
15 otro modo de codificación de entropía específico, la subdivisión de árbol cuádruple del bloque de árbol de codificación actual se señala dentro de la porción de sintaxis `coding_tree` mediante las banderas llamadas `split_coding_unit_flag` en la marca 400. Como se muestra en la Fig. 9, de acuerdo con la realización descrita a continuación, el bloque de raíz arbórea se subdivide de acuerdo con lo señalado por `split_coding_unit_flag` en un primer orden transversal en profundidad como se muestra en la Fig. 9a. Cada vez que se alcanza un nodo de hoja, lo mismo representa un conjunto
20 de codificación que se codifica de inmediato usando la función de sintaxis `coding_unit`. Esto se puede ver en la Fig. 9 cuando se observa la cláusula `if` en 402 que verifica si la `split_coding_unit_flag` actual está configurada o no. En caso afirmativo, se llama recursivamente a la función `coding_tree`, lo que lleva a una transmisión/extracción adicional de una `split_coding_unit_flag` adicional en el codificador y decodificador, respectivamente. Si no es así, es decir, si la `split_coding_unit_flag=0`, el sub-bloque actual del bloque de raíz arbórea 200 de la Fig. 5a es un bloque de hoja y, con
25 el fin de codificar este conjunto de codificación, se llama a la función `coding_unit` de la Fig. 10 en 404.

En la realización descrita actualmente, se usa la opción mencionada anteriormente de acuerdo con la cual la fusión es simplemente utilizable para imágenes para las cuales está disponible el modo de predicción inter. Es decir, los sectores/imágenes intracodificados no usan la fusión de todos modos. Esto es visible en la Fig. 10, donde la bandera
30 `merge_flag` se transmite en 406 simplemente en caso de que un tipo de sector no sea igual al tipo de sector intraimagen. La fusión se refiere, de acuerdo con la presente realización, simplemente a los parámetros de predicción relacionados con la predicción inter. De acuerdo con la presente realización, la `merge_flag` se señala para todo el conjunto de codificación 40 y también señala al decodificador un determinado modo de partición para el conjunto de codificación actual, concretamente el modo de no partición. Por consiguiente, la función `prediction_unit` se denomina
35 408 con la indicación de que el conjunto de codificación actual es un conjunto de predicción. Sin embargo, esta no es la única posibilidad de activar la opción de fusión. Más bien, si la `merge_flag` relacionada con todo el conjunto de codificación no se establece en 406, el tipo de predicción del conjunto de codificación del sector no intraimagen se señala en 410 mediante el elemento de sintaxis `pred_type` con, dependiendo de ello, la función de llamada `prediction_unit` para cualquier partición del conjunto de codificación actual en, por ejemplo, 412 en caso de que el conjunto de codificación actual no se particione adicionalmente. En la Fig. 10, simplemente se muestran cuatro
40 opciones de partición diferentes, pero las otras opciones de partición mostradas en la Fig. 6 también pueden estar disponibles. Otra posibilidad sería que la opción de partición `PART_NxN` no esté disponible, sino las otras. La asociación entre los nombres para los modos de partición usados en la Fig. 10 a las opciones de partición mostradas en la Fig. 6 se indica en la Fig. 6 por los respectivos subíndices debajo de las opciones de partición individuales. La función `prediction_unit` se llama para cada partición, tal como las particiones 50 y 60 en el orden de codificación mencionado anteriormente. La función `prediction_unit` comienza verificando la `merge_flag` en 414. Si se establece la `merge_flag`, en 416 sigue inevitablemente un `merge_index`. La verificación en la etapa 414, es para verificar si se ha establecido o no la `merge_flag` relacionada con todo el conjunto de codificación como se señaló en 406. Si no es así, se señala de nuevo una `merge_flag` en 418, y si se establece este último, sigue un `merge_index` en 420 que indica
45 el candidato a fusión para la partición actual. Nuevamente, se señala `merge_flag` para la partición actual en 418 simplemente en caso de que el modo de predicción actual del conjunto de codificación actual sea un modo de predicción inter (véase 422).

50 Como es visible en la Fig. 11, la transmisión de los parámetros de predicción en uso para el conjunto de predicción actual en 424 se realiza, de acuerdo con la presente realización, simplemente en caso de que la fusión no se use para el presente conjunto de predicción.

Aunque la descripción anterior de la realización de las Figs. 9-11 ya describe la mayoría de la funcionalidad y semántica, a continuación se presenta alguna información adicional.

60 `merge_flag[x0][y0]` especifica si los parámetros de predicción inter para el conjunto de predicción actual (véanse 50 y 60 en las figuras) se infieren de una partición interpretada vecina. Los índices de matriz `x0`, `y0` especifican la ubicación (`x0`, `y0`) de la muestra de luma superior izquierda del bloque de predicción considerado (véanse 50 y 60 en las figuras) con respecto a la muestra de luma superior izquierda de la imagen (véase 20 en las figuras).

merge_idx[x0][y0] especifica el índice de candidato a fusión de la lista de candidatos a fusión donde x0, y0 especifican la ubicación (x0, y0) de la muestra de luma superior izquierda del bloque de predicción considerado con respecto a la muestra de luma superior izquierda de la imagen.

5

Aunque no se indica específicamente en la descripción anterior de las Figs. 9-11, los candidatos a fusión o la lista de candidatos a fusión se determinan en esta realización usando de manera ejemplar no solo parámetros de codificación o parámetros de conjuntos de predicción/particiones espacialmente vecinos, sino que también se forma una lista de candidatos mediante el uso de parámetros de predicción de particiones temporalmente vecinas de imágenes temporalmente vecinas y previamente codificadas. Además, se usan combinaciones de parámetros de predicción de conjuntos de predicción/particiones espacial y/o temporalmente vecinos y se incluyen en la lista de candidatos a fusión. Naturalmente, se puede usar simplemente un subconjunto de los mismos. En particular, la Fig. 12 muestra una posibilidad de determinar vecinos espaciales, es decir, particiones o conjuntos de predicción espacialmente vecinos. La Fig. 12 muestra de manera ejemplar un conjunto de predicción o partición 60 y los píxeles B₀ a B₂ y A₀ y A₁ que se ubican directamente adyacentes al borde 500 de la partición 60, concretamente B₂ que es diagonalmente adyacente al píxel superior izquierdo de la partición 60, B₁ que se ubica verticalmente por encima y adyacente al píxel superior derecho de la partición 60, B₀ que se ubica diagonalmente al píxel superior derecho de la partición 60, A₁ que se ubica horizontalmente a la izquierda de y adyacente al píxel inferior izquierdo de la partición 60, y A₀ que se ubica diagonalmente al píxel inferior izquierdo de la partición 60. Una partición que incluye al menos uno de los píxeles B₀ a B₂ y A₀ y A₁ forma un vecino espacial y los parámetros de predicción del mismo forman un candidato a fusión.

20

Con el fin de realizar la eliminación anteriormente mencionada de los candidatos que conduciría a otro modo de partición que también habría estado disponible, se podrían usar las siguientes funciones:

25

En particular, el candidato N, es decir, los parámetros de codificación/predicción que se derivan del conjunto de predicción/partición que cubre el píxel N=(B₀, B₁, B₂, A₀, A₁), es decir, la posición (xN, yN), se elimina de la lista de candidatos si cualquiera de las siguientes condiciones es verdadera (consúltese la Fig. 6 para el modo de partición PartMode y el índice de partición correspondiente PartIdx que indexa la partición respectiva dentro del conjunto de codificación):

30

PartMode del conjunto de predicción actual es PART_2NxN y PartIdx es igual a 1 y los conjuntos de predicción que cubren la ubicación de luma (xP, yP - 1) (PartIdx = 0) y la ubicación de luma (xN, yN) (Cand. N) tienen parámetros de movimiento idénticos:

35

$$mvLX[xP, yP - 1] == mvLX[xN, yN]$$

$$refIdxLX[xP, yP - 1] == refIdxLX[xN, yN]$$

$$predFlagLX[xP, yP - 1] == predFlagLX[xN, yN]$$

40

PartMode del conjunto de predicción actual es PART_Nx2N y PartIdx es igual a 1 y los conjuntos de predicción que cubren la ubicación de luma (xP - 1, yP) (PartIdx = 0) y la ubicación de luma (xN, yN) (Cand. N) tienen parámetros de movimiento idénticos:

45

$$mvLX[xP - 1, yP] == mvLX[xN, yN]$$

$$refIdxLX[xP - 1, yP] == refIdxLX[xN, yN]$$

$$predFlagLX[xP - 1, yP] == predFlagLX[xN, yN]$$

50

PartMode del conjunto de predicción actual es PART_NxN y PartIdx es igual a 3 y los conjuntos de predicción que cubren la ubicación de luma (xP - 1, yP) (PartIdx = 2) y la ubicación de luma (xP - 1, yP - 1) (PartIdx = 0) tienen parámetros de movimiento idénticos:

55

$$mvLX[xP - 1, yP] == mvLX[xP - 1, yP - 1]$$

$$refIdxLX[xP - 1, yP] == refIdxLX[xP - 1, yP - 1]$$

$$predFlagLX[xP - 1, yP] == predFlagLX[xP - 1, yP - 1]$$

60

y los conjuntos de predicción que cubren la ubicación de luma (xP, yP - 1) (PartIdx = 1) y la ubicación de luma (xN, yN) (Cand. N) tienen parámetros de movimiento idénticos:

$$mvLX[xP, yP - 1] == mvLX[xN, yN]$$

$$refIdxLX[xP, yP - 1] == refIdxLX[xN, yN]$$

$$5 \quad predFlagLX[xP, yP - 1] == predFlagLX[xN, yN]$$

.- PartMode del conjunto de predicción actual es PART_NxN y PartIdx es igual a 3 y los conjuntos de predicción que cubren la ubicación de luma (xP, yP - 1) (PartIdx = 1) y la ubicación de luma (xP - 1, yP - 1) (PartIdx = 0) tienen parámetros de movimiento idénticos:

$$10 \quad mvLX[xP, yP - 1] == mvLX[xP - 1, yP - 1]$$

$$refIdxLX[xP, yP - 1] == refIdxLX[xP - 1, yP - 1]$$

$$15 \quad predFlagLX[xP, yP - 1] == predFlagLX[xP - 1, yP - 1]$$

y los conjuntos de predicción que cubren la ubicación de luma (xP - 1, yP) (PartIdx = 2) y la ubicación de luma (xN, yN) (Cand. N) tienen parámetros de movimiento idénticos:

$$20 \quad mvLX[xP - 1, yP] == mvLX[xN, yN]$$

$$refIdxLX[xP - 1, yP] == refIdxLX[xN, yN]$$

25 En este sentido, obsérvese que la posición o ubicación (xP, yP) denota el píxel más alto de la partición/conjunto de predicción actual. Es decir, de acuerdo con el primer elemento, se verifican todos los candidatos a parámetros de codificación que se han derivado adoptando directamente los parámetros de codificación respectivos de conjuntos de predicción vecinos, concretamente el conjunto de predicción N. Sin embargo, los otros candidatos a parámetros de codificación adicionales pueden verificarse de la misma manera en cuanto a si son iguales a los parámetros de codificación del conjunto de predicción respectivo que surge con lo que daría como resultado la obtención de otro patrón de partición también soportado por la sintaxis. De acuerdo con las realizaciones recién descritas, la igualdad de los parámetros de codificación comprende una verificación de la igualdad del vector de movimiento, es decir, mvLX, el índice de referencia, es decir, refIdxLX, y la bandera de predicción predFlagLX que indica que los parámetros, es decir, el vector de movimiento y el índice de referencia, asociados con la lista de referencia X, donde X es 0 o 1, se usan en la predicción inter.

35 Obsérvese que la posibilidad recién descrita para la eliminación de candidatos a parámetros de codificación de conjuntos de predicción/particiones vecinos también sería aplicable en caso de soportar modos de partición asimétricos mostrados en la mitad derecha de la Fig. 6. En ese caso, el modo PART_2NxN podría representar todos los modos de subdivisión horizontal y PART_Nx2N podría corresponder a todos los modos de subdivisión vertical. Además, el modo PART_2NxN podría excluirse de los modos de partición o patrones de partición soportados y, en ese caso, simplemente se tendrían que realizar las dos primeras verificaciones de eliminación.

45 Con respecto a la realización de las Figs. 9-12, también debería observarse que es posible excluir las particiones intrapredichas de la lista de candidatos, es decir, sus parámetros de codificación, naturalmente, no se incluyen en la lista de candidatos.

Además, se observa que se podrían usar tres contextos para la merge_flag y el merge_index.

50 Si bien se han descrito algunos aspectos en el contexto de un aparato, es obvio que estos aspectos también representan una descripción del método correspondiente, en el cual un bloque o dispositivo corresponde a una etapa del método o a una característica de una etapa del método. De manera análoga, los aspectos descritos en el contexto de una etapa del método también representan una descripción de un bloque o elemento correspondiente o de una característica de un aparato correspondiente. Algunos o todas las etapas del método pueden ser ejecutados por medio de (o utilizando) un aparato de hardware, como por ejemplo, un microprocesador, un ordenador programable o un circuito electrónico. En algunas realizaciones, alguno o varios de los pasos metodológicos más importantes pueden ser ejecutados por tal aparato.

60 Dependiendo de ciertos requisitos de implementación, las realizaciones de la invención se pueden implementar en hardware o en software. La implementación se puede realizar empleando un medio de almacenamiento digital, por ejemplo un disco blando, un DVD, un Blu-Ray, un CD, una ROM, una PROM, una EPROM, una EEPROM o una memoria FLASH, que tiene almacenadas en la misma señales control legibles electrónicamente, que cooperan (o tienen capacidad para cooperar) con un sistema de informático programable de tal manera que se ejecute el método respectivo. Por lo tanto, el medio de almacenamiento digital puede ser legible por ordenador.

Algunas realizaciones de acuerdo con la invención comprenden un transportador no transitorio de datos que comprende señales de control legibles electrónicamente, con capacidad para cooperar con un sistema informático programable de tal manera que se ejecute uno de los métodos descritos en el presente documento.

5

En general, las realizaciones de la presente invención pueden ser implementadas en forma de producto de programa informático con un código de programa, donde el código de programa cumple la función de ejecutar uno de los métodos al ejecutarse el programa informático en un ordenador. El código de programa puede ser almacenado, por ejemplo, en un portador legible por una máquina.

10

Otras realizaciones comprenden el programa informático para ejecutar uno de los métodos aquí descritos, almacenado en un portador legible por una máquina.

15

En otras palabras, una realización del método inventivo es, por lo tanto, un programa informático que tiene un código de programa para realizar uno de los métodos descritos en el presente documento, cuando el programa informático se ejecuta en un ordenador.

20

Una realización adicional de los métodos inventivos es, por lo tanto, un portador de datos (o un medio de almacenamiento digital, o un medio legible por ordenador) que comprende, grabado en el mismo, el programa informático para realizar uno de los métodos descritos en el presente documento. El soporte de datos, el medio de almacenamiento digital o el medio grabado son típicamente tangibles y/o no transitorios.

25

Otra realización del método inventivo es, por lo tanto, un flujo de datos o una secuencia de señales que representan el programa informático para realizar uno de los métodos descritos en la presente. El flujo de datos o la secuencia de señales pueden estar configurados, por ejemplo, para ser transferida a través de una conexión de comunicación de datos, por ejemplo a través de Internet.

30

Otra realización comprende un medio de procesamiento, por ejemplo un ordenador o un dispositivo lógico programable, configurado o adaptado para realizar uno de los métodos descritos en el presente documento.

35

Otra realización de acuerdo con la invención comprende un aparato o un sistema configurado para transferir (por ejemplo, electrónicamente u ópticamente) un programa informático para realizar uno de los métodos descritos en el presente documento a un receptor. El receptor puede ser, por ejemplo, un ordenador, un dispositivo móvil, un dispositivo de memoria o similar. El aparato o sistema puede, por ejemplo, comprender un servidor de archivos para transferir el programa informático al receptor.

40

En algunas realizaciones, se puede utilizar un dispositivo lógico programable (por ejemplo, una matriz de puertas programable en campo) para realizar algunas o todas las funcionalidades de los métodos descritos en el presente documento. En algunas realizaciones, una matriz de puertas programables en campo puede cooperar con un microprocesador para realizar uno de los métodos descritos en el presente documento. Por lo general, los métodos son ejecutados preferentemente por cualquier aparato de hardware.

45

Las realizaciones anteriormente descritas son meramente ilustrativas de los principios de la presente invención. Se entiende que modificaciones y variaciones de las disposiciones y los detalles aquí descritos resultarán evidentes para los expertos en la técnica. El alcance de la protección está definido por las reivindicaciones adjuntas.

50

REIVINDICACIONES

1. Decodificador (80) configurado para decodificar un flujo de bits (30) que señala uno de los patrones de partición soportados para un bloque actual (40) de una imagen (20), estando el decodificador configurado para
5
si el señalado de los patrones de partición soportados especifica una subdivisión del bloque actual (40) en dos o más particiones de bloque (50, 60),
10
para cada una de las particiones de bloque excepto una primera partición de bloque de las particiones de bloque del bloque actual en un orden de codificación (70),
determinar, para la partición de bloque respectiva (60) del bloque actual, un conjunto de candidatos a parámetros de codificación, donde
15
al menos algunos candidatos a parámetros de codificación se adoptan a partir de los parámetros de codificación de simplemente una partición de bloque previamente decodificada de modo que al menos algunos candidatos a parámetros de codificación sean iguales a los parámetros de codificación de dicha simplemente una partición de bloque previamente decodificada,
20
con realizar la determinación de tal manera que
candidatos a parámetros de codificación iguales a los parámetros de codificación asociados con cualquiera de las particiones de bloques, que, al fusionarse con la respectiva partición de bloques (60), darían como resultado uno de los patrones de partición soportados,
25
se excluyen del conjunto de candidatos a parámetros de codificación para la partición de bloque respectiva (60),
30
caracterizado por que
la determinación del conjunto de candidatos a parámetros de codificación de la partición de bloque respectiva (60) del bloque actual comprende además derivar al menos algunos candidatos a parámetros de codificación adicionales
35
de una combinación de parámetros de codificación de más de una partición de bloque previamente decodificada, o
de - por modificación - parámetros de codificación de una partición previamente decodificada.
40
2. Decodificador de acuerdo con la reivindicación 1, en el que el decodificador (80) está configurado para establecer, si un número de los candidatos a parámetros de codificación no excluidos es distinto de cero, los parámetros de codificación asociados con la partición de bloque respectiva (60) dependiendo de uno de los candidatos a parámetros de codificación no excluidos.
45
3. Decodificador de acuerdo con la reivindicación 1, en el que el decodificador (80) está configurado para establecer, si un número de los candidatos a parámetros de codificación no excluidos es distinto de cero, los parámetros de codificación asociados con la partición de bloque respectiva iguales a uno de los candidatos a parámetros de codificación no excluidos.
50
4. Decodificador de acuerdo con cualquiera de las reivindicaciones 2 a 3, en el que el decodificador está configurado para admitir modos de predicción intra e inter para el bloque actual y para realizar
la fusión y
55
la determinación del conjunto de candidatos de parámetros de codificación de la manera en que, del conjunto de candidatos de parámetros de codificación para la partición de bloque respectiva (60), los candidatos de parámetros de codificación que igualan los parámetros de codificación asociados con cualquiera de las particiones de bloque, que, al fusionarse con la partición de bloque respectiva (60), darían como resultado uno de los patrones de partición admitidos, se excluyen del conjunto de candidatos de parámetros de codificación para la partición de bloque respectiva (60)
60
simplemente en el caso de que el bloque actual (40) se codifique en modo de inter predicción.

5. Decodificador de acuerdo con cualquiera de las reivindicaciones 2 a 4, en el que los parámetros de codificación son parámetros de predicción y el decodificador está configurado para usar los parámetros de predicción de la partición de bloque respectiva (60) con el fin de derivar una señal de predicción para la partición de bloque respectiva (60).
- 5 6. Decodificador de acuerdo con cualquiera de las reivindicaciones 1 a 5, en donde el decodificador está configurado para, dependiendo de un número de candidatos a parámetros de codificación no excluidos para una partición de bloque respectiva, simplemente esperar que el flujo de bits (30) comprenda un elemento de sintaxis que especifique cuál de los candidatos a parámetros de codificación no excluidos se emplea para fusión, si el número de candidatos a parámetros de codificación no excluidos para la partición de bloque respectiva es mayor que 1.
- 10 7. Decodificador de acuerdo con cualquiera de las reivindicaciones 1 a 6, en el que el decodificador (80) está configurado para determinar el conjunto de candidatos a parámetros de codificación para la partición de bloque respectiva en función, al menos parcialmente, de los parámetros de codificación asociados con particiones de bloque previamente decodificadas vecinas a la partición de bloque respectiva, y que se encuentran fuera y dentro del bloque actual, respectivamente.
- 15 8. Decodificador de acuerdo con cualquiera de las reivindicaciones 1 a 7, en el que el decodificador (80) está configurado para determinar el conjunto de candidatos a parámetros de codificación para la partición de bloque respectiva de un conjunto inicial de bloques previamente decodificados, excluyendo los que se codifican en un modo de predicción intra.
- 20 9. Decodificador de acuerdo con cualquiera de las reivindicaciones 1 a 8, en donde el decodificador (80) está configurado para subdividir la imagen (20) en bloques de codificación de acuerdo con la información de subdivisión contenida en el flujo de bits (30), donde los bloques de codificación incluyen el bloque actual.
- 25 10. Decodificador de acuerdo con la reivindicación 9, en donde decodificador (80) está configurado para subdividir adicionalmente el bloque actual (40) en uno o más bloques de transformación de acuerdo con la información de subdivisión adicional contenida en el flujo de bits y para derivar una señal residual del bloque actual (40) del flujo de bits (30) en conjuntos de los bloques de transformación.
- 30 11. Decodificador de acuerdo con cualquiera de las reivindicaciones 1 a 10, en el que la imagen tiene asociada con la misma un mapa de profundidad como información adicional.
- 35 12. Codificador (10) configurado para codificar una imagen (20) en un flujo de bits (30), el codificador está configurado para
40 señalizar dentro de un flujo de bits (30) uno de los patrones de partición admitidos para un bloque actual (40);
y
si el señalado de los patrones de partición soportados especifica una subdivisión del bloque actual (40) en dos o más particiones de bloque (50, 60),
45 para cada una de las particiones de bloque excepto una primera partición de bloque de las particiones de bloque del bloque actual en un orden de codificación (70),
determinar, para la partición de bloque respectiva (60) del bloque actual, un conjunto de candidatos a parámetros de codificación, donde
50 al menos algunos candidatos a parámetros de codificación se adoptan a partir de los parámetros de codificación de simplemente una partición de bloque previamente codificada de modo que al menos algunos candidatos a parámetros de codificación sean iguales a los parámetros de codificación de dicha simplemente una partición de bloque previamente codificada,
55 con realizar la determinación de tal manera que
candidatos de parámetros de codificación que igualan los parámetros de codificación asociados con cualquiera de las particiones de bloque, lo que, al fusionarse con la partición de bloque respectiva, daría como resultado uno de los patrones de partición admitidos,
60 se excluyen del conjunto de candidatos a parámetros de codificación para la partición de bloque respectiva (60),

- caracterizado por que
- la determinación del conjunto de candidatos a parámetros de codificación de la partición de bloque respectiva (60) del bloque actual comprende además derivar al menos algunos candidatos a parámetros de codificación adicionales
- de una combinación de parámetros de codificación de más de una partición de bloque previamente codificada, o
- de - por modificación - parámetros de codificación de una partición previamente codificada.
13. Codificador de acuerdo con la reivindicación 12, en el que la imagen tiene asociado con la misma un mapa de profundidad como información adicional.
14. Método para decodificar un flujo de bits (30) que señala uno de los patrones de partición soportados para un bloque actual (40) de una imagen (20), comprendiendo el procedimiento
- si el señalado de los patrones de partición soportados especifica una subdivisión del bloque actual (40) en dos o más particiones de bloque (50, 60),
- para cada una de las particiones de bloque excepto una primera partición de bloque de las particiones de bloque del bloque actual en un orden de codificación (70),
- determinar, para la partición de bloque respectiva (60) del bloque actual, un conjunto de candidatos a parámetros de codificación, donde
- al menos algunos candidatos a parámetros de codificación se adoptan a partir de los parámetros de codificación de simplemente una partición de bloque previamente decodificada de modo que al menos algunos candidatos a parámetros de codificación sean iguales a los parámetros de codificación de dicha simplemente una partición de bloque previamente decodificada,
- con realizar la determinación de tal manera que
- candidatos a parámetros de codificación iguales a los parámetros de codificación asociados con cualquiera de las particiones de bloques, que, al fusionarse con la respectiva partición de bloques (60), darían como resultado uno de los patrones de partición soportados,
- se excluyen del conjunto de candidatos a parámetros de codificación para la partición de bloque respectiva (60),
- caracterizado por que
- la determinación del conjunto de candidatos a parámetros de codificación de la partición de bloque respectiva (60) del bloque actual comprende además derivar al menos algunos candidatos a parámetros de codificación adicionales
- de una combinación de parámetros de codificación de más de una partición de bloque previamente decodificada, o
- de - por modificación - parámetros de codificación de una partición previamente decodificada.
15. Método para codificar una imagen (20) en un flujo de bits (30), el método comprende señalar dentro de un flujo de bits (30) uno de los patrones de partición admitidos para un bloque actual (40); y
- si el señalado de los patrones de partición soportados especifica una subdivisión del bloque actual (40) en dos o más particiones de bloque (50, 60),
- para cada una de las particiones de bloque excepto una primera partición de bloque de las particiones de bloque del bloque actual en un orden de codificación (70),
- determinar, para la partición de bloque respectiva (60) del bloque actual, un conjunto de candidatos a parámetros de codificación, donde
- al menos algunos candidatos a parámetros de codificación se adoptan a partir de los parámetros de

- codificación de simplemente una partición de bloque previamente codificada de modo que al menos algunos candidatos a parámetros de codificación sean iguales a los parámetros de codificación de dicha simplemente una partición de bloque previamente codificada,
- 5 con realizar la determinación de tal manera que
- candidatos a parámetros de codificación iguales a los parámetros de codificación asociados con cualquiera de las particiones de bloques, que, al fusionarse con la respectiva partición de bloques (60), darían como resultado uno de los patrones de partición soportados,
- 10 se excluyen del conjunto de candidatos a parámetros de codificación para la partición de bloque respectiva (60),
- caracterizado por que
- 15 la determinación del conjunto de candidatos a parámetros de codificación de la partición de bloque respectiva (60) del bloque actual comprende además derivar al menos algunos candidatos a parámetros de codificación adicionales
- 20 de una combinación de parámetros de codificación de más de una partición de bloque previamente decodificada, o
- de - por modificación - parámetros de codificación de una partición previamente codificada.
- 25 16. Programa informático que tiene un código de programa para realizar, cuando se ejecuta en un ordenador, un procedimiento de acuerdo con la reivindicación 14 o 15.

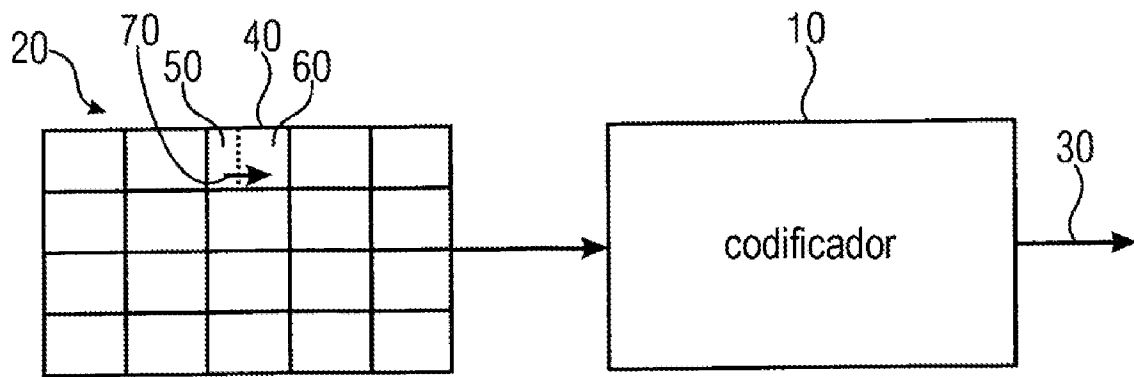


FIGURA 1

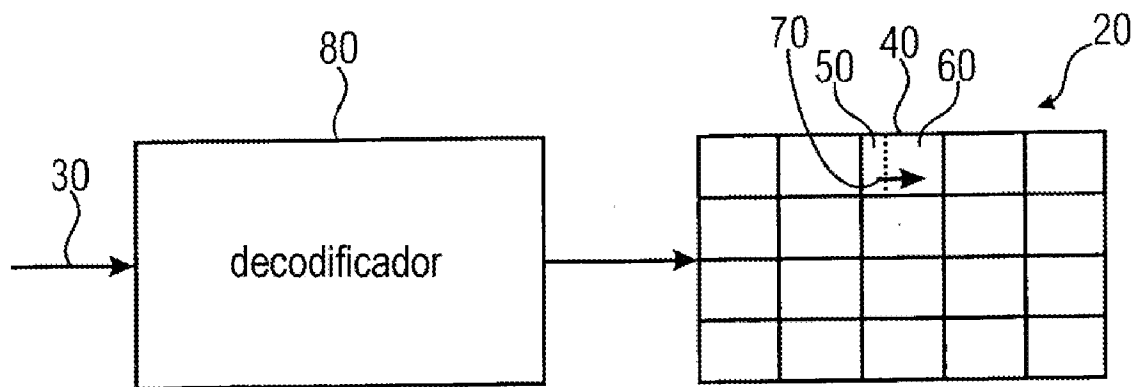


FIGURA 2

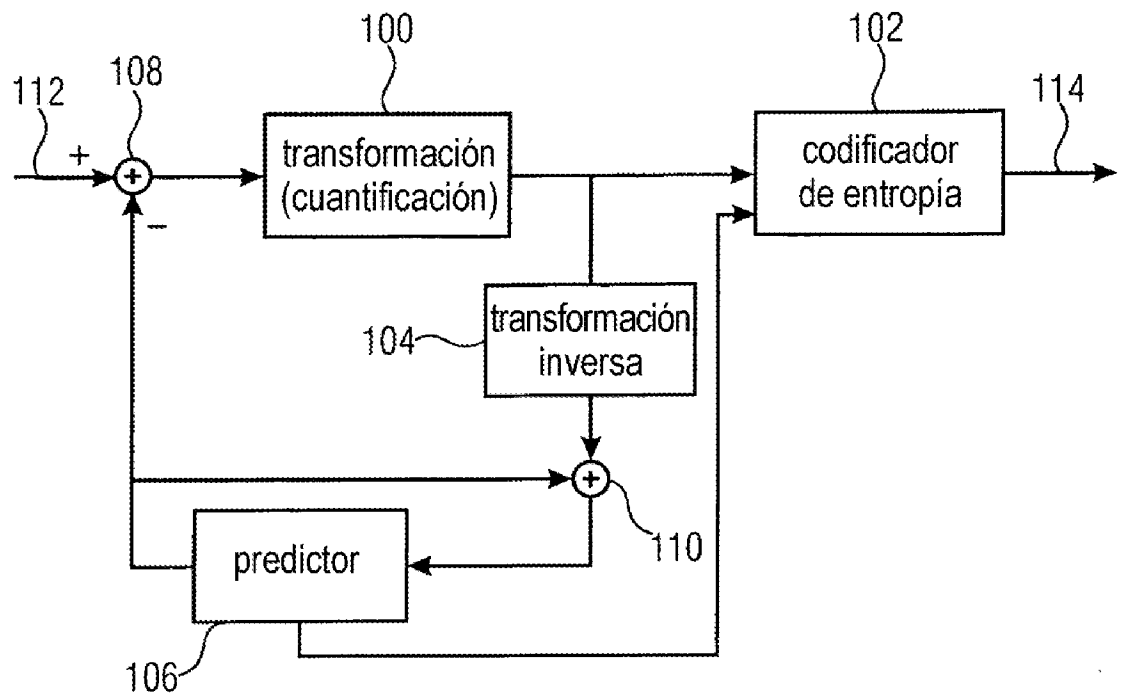


FIGURA 3

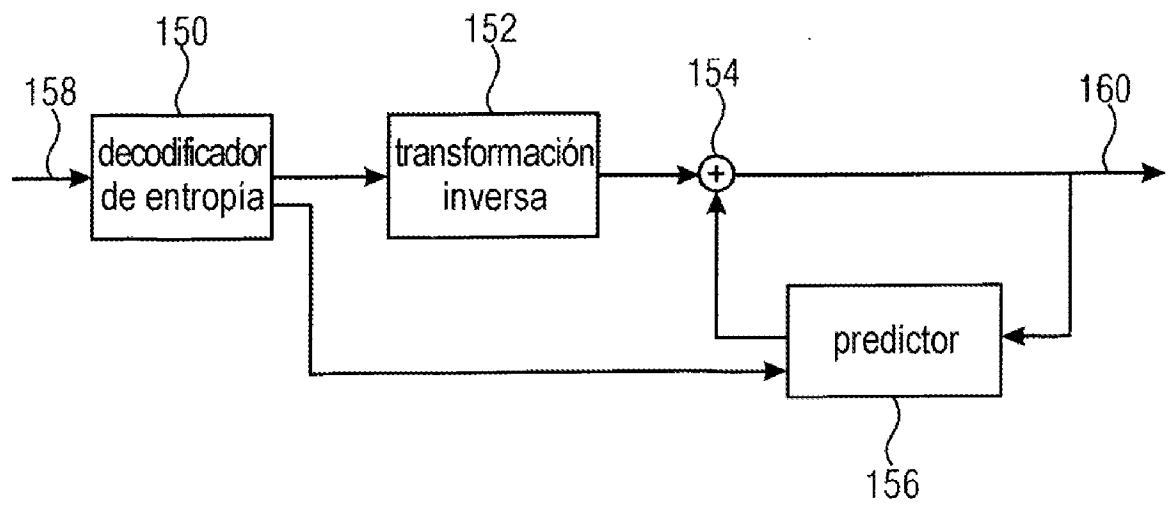


FIGURA 4

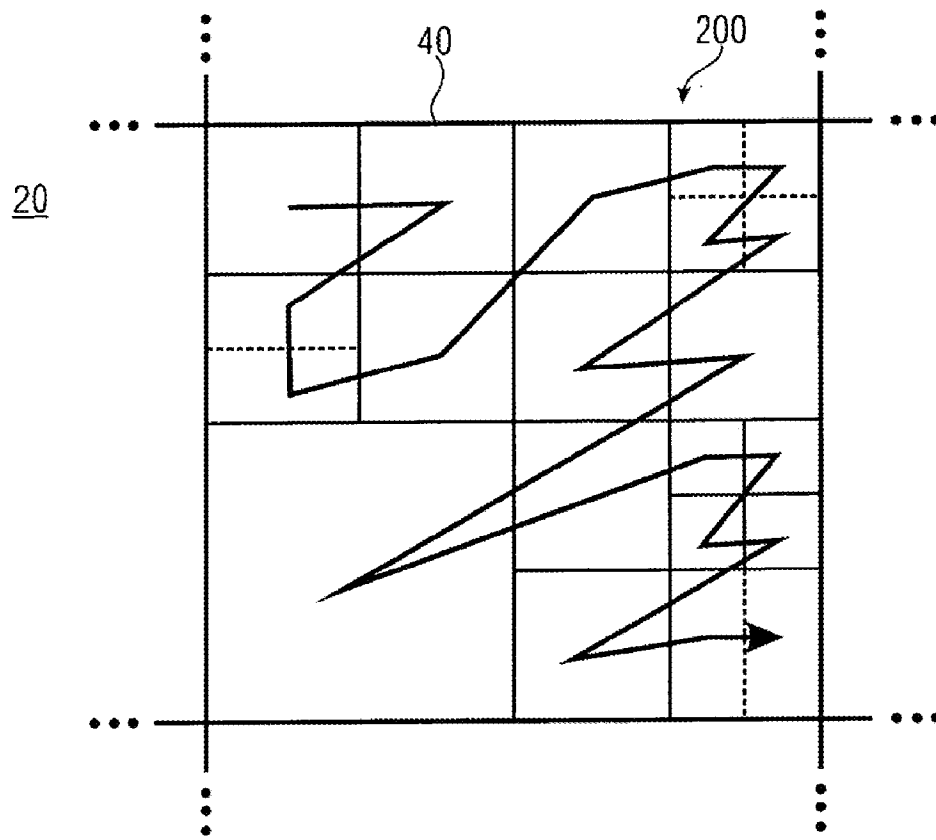


FIGURA 5A

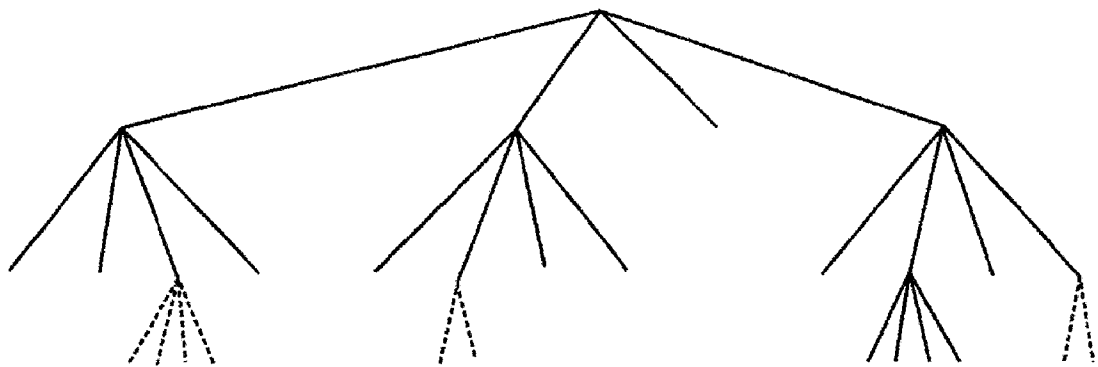


FIGURA 5B

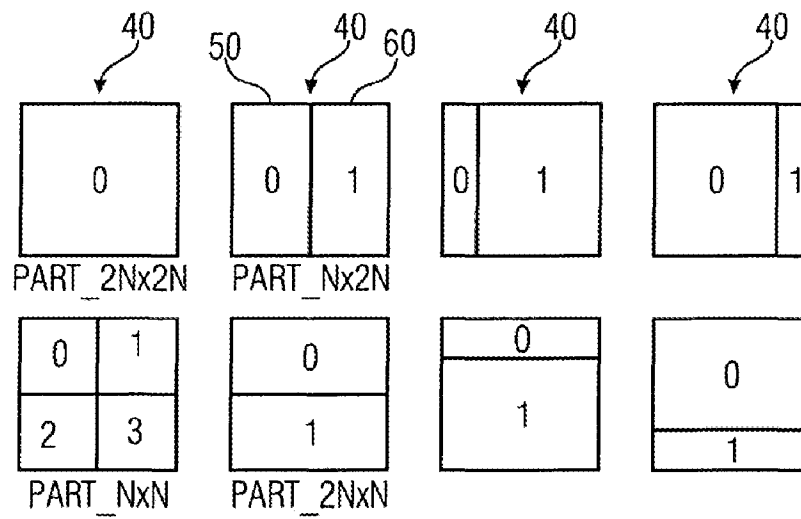


FIGURA 6

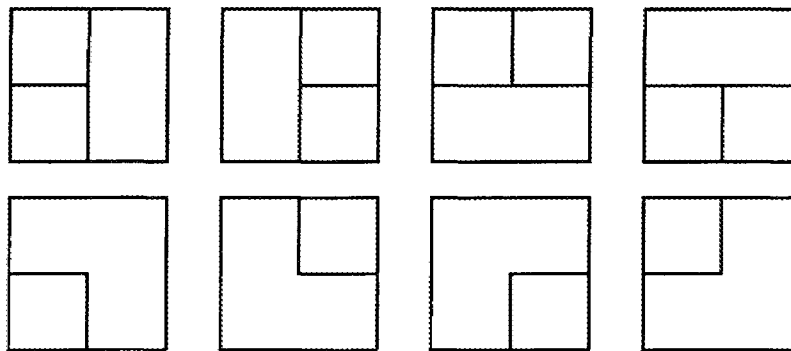


FIGURA 7

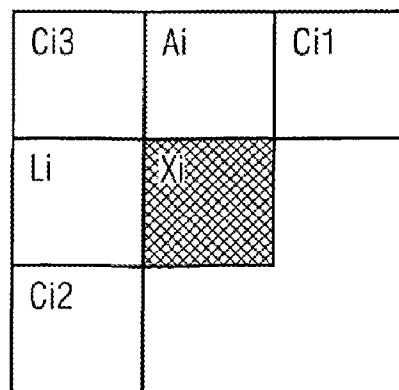


FIGURA 8

coding_tree(x0, y0, log2CUSize) {	Descriptor
<pre> if(x0 + (1 << log2CUSize) <= PicWidthInSamplesL && y0 + (1 << log2CUSize) <= PicHeightInSamplesL && cuAddress(x0, y0) >= SliceAddress) { if(!entropy_coding_mode_flag && slice_type != I) cu_split_pred_part_mode[x0][y0] </pre>	ce(v)
<pre> 400 else if(log2CUSize > Log2MinCUSize) split_coding_unit_flag[x0][y0] </pre>	u(1) ae(v)
<pre> } if(adaptive_loop_filter_flag && alf_cu_control_flag) { cuDepth = Log2MaxCUSize - log2CUSize if(cuDepth <= alf_cu_control_max_depth) if(cuDepth == alf_cu_control_max_depth split_coding_unit_flag[x0][y0] == 0) AlfCuFlagIdx++ 402 } if(split_coding_unit_flag[x0][y0]) { if(cu_qp_delta_enabled_flag && log2CUSize == log2MinCUDQPSize) IsCuQpDeltaCoded 0 x1 = x0 + ((1 << log2CUSize) >> 1) y1 = y0 + ((1 << log2CUSize) >> 1) if(cuAddress(x1, y0) > SliceAddress) moreDataFlag = coding_tree(x0, y0, log2CUSize - 1) if(cuAddress(x0, y1) > SliceAddress && moreDataFlag && x1 < PicWidthInSamplesL) moreDataFlag = coding_tree(x1, y0, log2CUSize - 1) if(cuAddress(x1, y1) > SliceAddress && moreDataFlag && y1 < PicHeightInSamplesL) moreDataFlag = coding_tree(x0, y1, log2CUSize - 1) if(moreDataFlag && x1 < PicWidthInSamplesL && y1 < PicHeightInSamplesL) moreDataFlag = coding_tree(x1, y1, log2CUSize - 1) } else { if(adaptive_loop_filter_flag && alf_cu_control_flag) AlfCuFlag[x0][y0] = alf_cu_flag[AlfCuFlagIdx] 404 coding_unit(x0, y0, log2CUSize) if(!entropy_coding_mode_flag) moreDataFlag = more_rbsp_data() else { if(granularity_block_boundary(x0, y0, log2CUSize)) { end_of_slice_flag </pre>	ae(v)
<pre> moreDataFlag = !end_of_slice_flag } else moreDataFlag = 1 } } return moreDataFlag } </pre>	

FIGURA 9

coding_unit(x0, y0, log2CUSize) {	Descriptor
if(entropy_coding_mode_flag && slice_type != I) → merge_flag[x0][y0]	u(1) ae(v)
406 if(merge_flag[x0][y0]) → prediction_unit(x0, y0, log2CUSize, log2CUSize, 0, 0) 408 else { if(!entropy_coding_mode_flag) { if(slice_type == I && log2CUSize == Log2MinCUSize) intra_part_mode	u(1)
} else if(slice_type != I log2CUSize == Log2MinCUSize) → pred_type	u(1) ae(v)
410 x1 = x0 + ((1 << log2CUSize) >> 1) y1 = y0 + ((1 << log2CUSize) >> 1) if(PartMode == PART_2Nx2N) { → prediction_unit(x0, y0, log2CUSize, log2CUSize, 0) 412 } else if(PartMode == PART_2NxN) { prediction_unit(x0, y0, log2CUSize, log2CUSize - 1, 0) prediction_unit(x0, y1, log2CUSize, log2CUSize - 1, 1) } else if(partmode == PART_Nx2N) { prediction_unit(x0, y0, log2CUSize - 1, log2CUSize, 0) prediction_unit(x1, y0, log2CUSize - 1, log2CUSize, 1) } else { /* PART_NxN */ prediction_unit(x0, y0, log2CUSize - 1, log2CUSize - 1, 0) prediction_unit(x1, y0, log2CUSize - 1, log2CUSize - 1, 1) prediction_unit(x0, y1, log2CUSize - 1, log2CUSize - 1, 2) prediction_unit(x1, y1, log2CUSize - 1, log2CUSize - 1, 3) } : } }	

FIGURA 10

	prediction_unit(x0, y0, log2PUWidth, log2PUHeight, PartIdx) {	Descriptor
	if(merge_flag[x0][y0]) {	
414	merge_idx[x0][y0]	ue(v) ae(v)
	} else if(PredMode == MODE_INTRA) {	
416	if(PartMode == PART_2Nx2N && log2PUWidth >= Log2IPCMCSize)	
	pcm_flag	u(1) ae(v)
	if(pcm_flag) {	
	while (!byte_aligned())	
	pcm_alignment_zero_bit	u(v)
	for(i = 0; i < 1 << (log2CUSize << 1); i++)	
	pcm_sample luma[i]	u(v)
	for(i = 0; i < (1 << (log2CUSize << 1)) >> 1; i++)	
	pcm_sample chroma[i]	u(v)
	} else {	
	prev_intra_luma_pred_flag[x0][y0]	u(1) ae(v)
	if(prev_intra_luma_pred_flag[x0][y0])	
	if(NumMPMCand < 1)	
	mpm_idx[x0][y0]	u(1) ae(v)
	Else	
	rem_intra_luma_pred_mode[x0][y0]	ce(v) ae(v)
	if(IntraPredMode[x0][y0] == 2)	
	planar_flag_luma[x0][y0]	u(1) ae(v)
	intra_chroma_pred_mode[x0][y0]	ue(v) ae(v)
	SignaledAsChromaDC =	
	(chroma_pred_from_luma-enabled_flag ?	
	intra_chroma_pred_mode[x0][y0] == 3 :	
	intra_chroma_pred_mode[x0][y0] == 2)	
	if(IntraPredMode[x0][y0] != 2 &&	
	IntraPredMode[x0][y0] != 34 && SignaledAsChromaDC)	
422	planar_flag_chroma[x0][y0]	u(1) ae(v)
	} else { /* MODE_INTER */	
	if(entropy_coding_mode_flag PartMode != PART_2Nx2N)	
418	merge_flag[x0][y0]	u(1) ae(v)
	if(merge_flag[x0][y0]) {	
420	merge_idx[x0][y0]	ue(v) ae(v)
	} else {	
	if(slice_type == B) {	
	if(!entropy_coding_mode_flag) {	
	combined_inter_pred_ref_idx	ue(v)
	if(combined_inter_pred_ref_idx == MaxPredRef)	
	inter_pred_flag[x0][y0]	ue(v)
	⋮	
	⋮	
	⋮	
	} else	

FIGURA 11A

FIGURA 11

FIGURA 11A
FIGURA 11B

	inter_pred_flag[x0][y0]	ue(v) ae(v)
	} if(inter_pred_flag[x0][y0] == Pred_LC) { if(num_ref_idx_lc_active_minus1 > 0) { if(entropy_coding_mode_flag) { if(combined_inter_pred_ref_idx == MaxPredRef) ref_idx_lc_minus4[x0][y0]	ue(v)
	} else ref_idx_lc[x0][y0]	ae(v)
	} mvd_lc[x0][y0][0]	se(v) ae(v)
424	mvd_lc[x0][y0][1]	se(v) ae(v)
	mvp_idx_lc[x0][y0]	ue(v) ae(v)
	} else { /* Pred_L0 or Pred_BI */ if(num_ref_idx_10_active_minus1 > 0) { if(entropy_coding_mode_flag) { if(combined_inter_pred_ref_idx == MaxPredRef) ref_idx_10_minusX[x0][y0]	ue(v)
	} else ref_idx_10_minusX[x0][y0]	ue(v) ae(v)
	} mvd_10[x0][y0][0]	se(v) ae(v)
	mvd_10[x0][y0][1]	se(v) ae(v)
	mvp_idx_10[x0][y0]	ue(v) ae(v)
	} if(inter_pred_flag[x0][y0] == Pred_BI) { if(num_ref_idx_11_active_minus1 > 0) { if(entropy_coding_mode_flag) { if(combined_inter_pred_ref_idx == MaxPredRef) ref_idx_11_minusX[x0][y0]	ue(v)
	} else ref_idx_11[x0][y0]	ue(v) ae(v)
	} mvd_11[x0][y0][0]	se(v) ae(v)
	mvd_11[x0][y0][1]	se(v) ae(v)
	mvp_idx_11[x0][y0]	ue(v) ae(v)
	} } }	

FIGURA 11 B

FIGURA 11

FIGURA 11 A

FIGURA 11 B

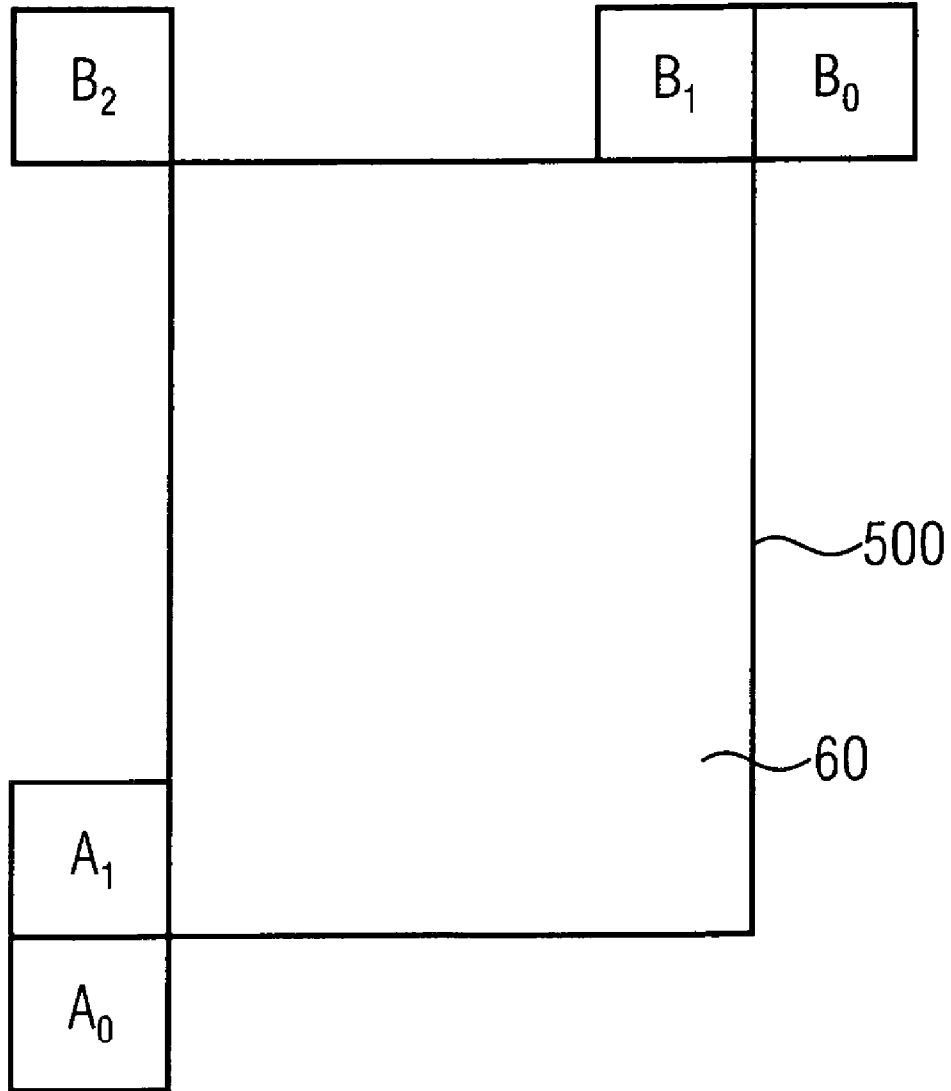


FIGURA 12