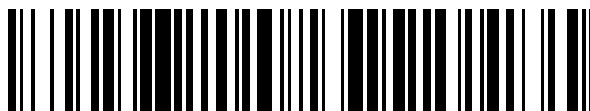


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 587 710**

51 Int. Cl.:

H04N 19/597 (2014.01)
H04N 19/30 (2014.01)
H04N 19/51 (2014.01)
H04N 19/31 (2014.01)
H04N 19/33 (2014.01)
H04N 19/70 (2014.01)
H04N 19/583 (2014.01)
H04N 19/105 (2014.01)
H04N 19/172 (2014.01)
H04N 19/573 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **04.04.2014** **PCT/SE2014/050421**
- 87 Fecha y número de publicación internacional: **16.10.2014** **WO14168561**
- 96 Fecha de presentación y número de la solicitud europea: **04.04.2014** **E 14720272 (5)**
- 97 Fecha y número de publicación de la concesión europea: **08.06.2016** **EP 2984841**

54 Título: **Construcción de listas de imágenes de referencia entre capas**

30 Prioridad:

12.04.2013 US 201361811376 P

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

26.10.2016

73 Titular/es:

TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)
(100.0%)
164 83 Stockholm, SE

72 Inventor/es:

RUSERT, THOMAS y
NORKIN, ANDREY

74 Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

ES 2 587 710 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

Construcción de listas de imágenes de referencia entre capas

Campo técnico

- 5 La invención se refiere a un método de construir al menos una lista de imágenes de referencia para la predicción entre capas de una imagen actual, un dispositivo para construir al menos una lista de imágenes de referencia para la predicción entre capas de una imagen actual, un programa correspondiente de ordenador y un producto correspondiente de programa de ordenador.

Antecedentes

- 10 La norma principal Codificación de Vídeo de Alta Eficiencia (HEVC) ha sido finalizada recientemente por la Unión Internacional de Telecomunicación (ITU) (ITU-T Rec. H. 265) y por el Grupo de Expertos de Imágenes en Movimiento (MPEG) (ISO/IEC 23008-2/MPEG-H Part 2). Se encuentran en desarrollo extensiones a las capas a la norma HEVC, por ejemplo, la extensión Multi Toma (MV-HEVC), la extensión 3D (3D-HEVC) y la extensión Escalable (SHVC). En el futuro se pueden especificar extensiones adicionales o combinaciones de las extensiones existentes.
- 15 La norma HEVC y sus extensiones hacen extensivo el uso de herramientas predictivas de codificación. Desde la perspectiva del descodificador, los datos del píxel se reconstruyen utilizando datos del píxel previamente descodificados para la predicción. En particular, para la predicción entre imágenes, imágenes previamente descodificadas, llamadas imágenes de referencia, se usan para la predicción en el proceso de reconstrucción de una imagen actual.
- 20 De acuerdo con la especificación HEVC, cada imagen se subdivide en uno o múltiples fragmentos y cada fragmento puede contener múltiples bloques (más específicamente, unidades de codificación en forma de bloques y unidades de predicción). Las imágenes que están disponibles como referencias de predicción para descodificar un fragmento actual se colocan en las llamadas listas de imágenes de referencia. De acuerdo con la especificación HEVC, existen diferentes tipos de fragmentos. Para los "fragmentos P", como mucho se puede utilizar una imagen de referencia para la predicción de un bloque actual. De acuerdo con ello, los fragmentos P tienen una lista de imágenes de referencia, llamada "list0". Para los "fragmentos B" se pueden utilizar como mucho dos imágenes de referencia para la predicción de un bloque actual, referida también como "predicción de bit". De acuerdo con ello, los fragmentos B tienen dos listas de imágenes de referencia, referidas como "list0" y "list1".
- 25 La imagen de referencia utilizada para reconstruir un bloque en particular puede ser señalizada por medio de los llamados índices de imagen de referencia. Un índice de imagen de referencia es un índice dentro de una lista de imágenes de referencia, tal como list0 o list1. Los índices de imágenes de referencia se codifican junto con otros datos en la corriente de bits HEVC como parte de los datos codificados del fragmento. La longitud de una palabra de código utilizada para enviar un índice de imágenes de referencia depende del propio valor del índice, en particular si se utiliza Codificación de Longitud Variable (VLC). Típicamente, los índices pequeños de imágenes de referencia requieren palabras de código más cortas. Por consiguiente, cuanto más adelante se coloque una cierta imagen de referencia al frente de una lista de imágenes de referencia, menos bits se requerirán para indicar su uso. De acuerdo con ello, con objeto de lograr un alto rendimiento en la compresión, una estrategia típica es colocar imágenes de referencia que se usen frecuentemente para la predicción al frente de una lista de imágenes de referencia.
- 30 Típicamente, las listas de imágenes de referencia se construyen en un proceso de dos etapas, (1) construcción de la lista inicial de imágenes de referencia seguida por (2) modificación de la lista de imágenes de referencia. La etapa 1 está predefinida por medio de la especificación del descodificador y da lugar a una lista inicial de imágenes de referencia. La etapa 2 implica la señalización de comandos de modificación de la lista de imágenes de referencia en los encabezamientos de los fragmentos y dan lugar a la lista final de imágenes de referencia aplicando los comandos de modificación de la lista de imágenes de referencia en la lista inicial de imágenes de referencia. Una vez enviados los comandos de modificación de la lista de imágenes de referencia se necesita la transmisión de bits adicionales, por lo que es deseable que la lista inicial de imágenes de referencia se diseñe cuidadosamente, de modo que las imágenes de referencia usadas con frecuencia se puedan indicar con pocos bits, proporcionando un alto rendimiento de compresión.
- 35 El documento US 2013/0077677 A1 describe las técnicas relativas a la construcción de listas de imágenes de referencia para la codificación de vídeo de capa única. La lista de imágenes de referencia se puede construir a partir de subconjuntos de imágenes de referencia de un conjunto de imágenes de referencia. Los subconjuntos de imágenes de referencia se pueden solicitar de una manera en particular para formar las listas de imágenes de referencia.
- 40 En "AHG21: Construction and modification of predefined reference picture sets and reference picture lists", documento de entrada JCTVC-G 548 al Equipo de Unión Colaborativa sobre Codificación de Video (JCT-VC) de ITU-T SG 16 WP3 y ISO/IEC JTC1/SC29/WG11, 7ª Reunión: Ginebra, CH, 21-30 de Noviembre, 2011 de Viktor
- 45
- 50
- 55

Wahadaniah, Chong Soon Lim y Sue Mon Thet Naing, se describe un número de modificaciones en el diseño del conjunto de imágenes de referencia y señalización de la lista de imágenes de referencia en HEVC.

Mientras que la especificación principal HEVC sólo utiliza temporalmente las imágenes cercanas para la predicción entre imágenes, es decir, imágenes dentro de la misma capa temporal, es probable que las extensiones HEVC multi capa, tales como extensiones escalables y 3D, vayan a utilizar imágenes de otras capas, por ejemplo, capas de escalabilidad y/o capas de tomas, como imágenes de referencia. En el borrador actual SHVC, MV-HEVC y 3D-HEVC, las especificaciones utilizan métodos ad hoc para la construcción de la lista de imágenes de referencia. Por consiguiente, con objeto de mejorar un poco la eficacia, existe la necesidad de métodos más eficaces para la construcción de las listas de imágenes de referencia para las extensiones HEVC multi capa utilizando imágenes de referencia a través de las capas.

En el borrador SHVC (JCTVC-L1008) y en las especificaciones MV-HEVC (JCT3V-C1004), una capa identifica, es decir, está asociada a un conjunto de imágenes que corresponden a, por ejemplo, una resolución espacial o calidad (para SHVC), a una toma de cámara (para MV-HEVC) o a una toma en profundidad (para 3D-HEVC). Cada capa tiene un índice *i* y se identifica por medio de un identificador de capa *layer_id* (véase más adelante el elemento de sintaxis *layer_id_in_nuh[i]*). El índice de capa *i* es típicamente un indicador del orden de la decodificación. Por ello, para cada unidad de acceso (por ejemplo, tiempo de muestreo o instante), hasta una imagen por cada capa (toma, resolución de la imagen, etcétera,) se decodifica según el orden del índice de capa *i*.

Adicionalmente, se asocia un conjunto de identificadores de la escalabilidad con cada capa (véase más adelante el elemento de sintaxis *dimension_id[i][j]*). Ejemplos para identificadores de la escalabilidad son "ViewId" (que identifica una cierta toma de cámara), "DepthFlag" (que identifica si una capa lleva datos de la profundidad o no), "DependencyId" (que indica las dependencias de la decodificación en el caso de, por ejemplo, escalabilidad espacial), "QualityId" (que indica una calidad del video) y otros.

En SHVC y MV-HEVC, los parámetros relativos a las representaciones de vídeo de alto nivel se señalizan en extensiones del llamado Conjunto de Parámetros de Vídeo (VPS). La sintaxis de la extensión VPS, y alguna semántica relativa, se describe más adelante. Específicamente, las dependencias de capa se señalizan utilizando el elemento de sintaxis "direct_dependency_flag", basándose en el cual se derivan para cada capa *i* las estructuras variables *RefLayerId[i][j]* y *NumDirectRefLayers[i]*, como se describe más adelante.

vps_extension() {	Descriptor
while(!byte_aligned() {	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0; NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
vps_nuh_layer_id_present_flag	u(1)
for(i = 1; i <= vps_max_layers_minus1; i++) {	

if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(lsdidx = 1; lsdidx <= vps_num_layer_sets_minus1; lsdidx ++) {	
vps_profile_present_flag [lsdidx]	u(1)
if(!vps_profile_present_flag[lsdidx])	
profile_layer_set_ref_minus1[lsdidx]	ue (v)
profile_tier_level(vps_profile_present_flag[lsdidx], vps_max_sub_layers_minus1)	
}	
num_output_layer_sets	ue (v)
for(i = 0; i < num_output_layer_sets; i++) {	
output_layer_set_idx[i]	ue (v)
lsidx = output_layer_set_idx[i]	
for(j = 0 ; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[lsdidx][j])	
output_layer_flag[lsdidx][j]	u(1)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
}	

layer_id_in_nuh[i] especifica el valor del elemento de sintaxis nuh_layer_id en las unidades de Capa de Codificación de Vídeo (VCL) y Capa de Abstracción de la Red (NAL) para la capa i-ésima. Para i en un intervalo de 0 a vps_max_layers_minus1, inclusive, cuando no está presente, el valor de layer_id_in_nuh[i] se infiere que es igual a i. Cuando i es mayor que 0, layer_id_in_nuh[i] será mayor que layer_id_in_nuh[i-1]. Para i en un intervalo de 0 a vps_max_layers_minus1, inclusive, la variable LayerIdInVps[layer_id_in_nuh[i]] se fija igual a i.

dimension_id[i][j] especifica el identificador del tipo de dimensión de la escalabilidad presente j-ésima de la capa i-ésima. Cuando no está presente, se infiere que el valor de[i][j] es igual a 0. El número de bits utilizados para la representación de dimension_id[i][j] es dimension_id_len_minus1[j] + 1. Cuando splitting_flag es igual a 1, un requisito de la conformidad de la corriente de bits es que dimension_id[i][j] deberá ser igual a ((layer_id_in_nuh[i] & ((1 << dimBitOffset[j] + 1) - 1)) >> dimBitOffset[j]).

La variable ScalabilityId[i] [smldx] que especifica el identificador del tipo de dimensión de la escalabilidad de smldx-ésimo de la capa i-ésima y la variable ViewId[layer_in_id_nuh[i]] que especifica el identificador de la toma de la capa i-ésima se derivan como sigue:

```

for ( i = 0; i <= vps_max_layers_minus1; i++ ) {
    for( smldx= 0; j =0; smldx< 16; smldx ++ )
        if( ( i != 0 ) && scalability_mask[ smldx ] )
            ScalabilityId[ i ] [ smldx ] = dimension_id [ i ] [ j++ ]
        else
            ScalabilityId [ i ] [ smldx ] = 0
    ViewId [ layer_id_in_nuh[ i ] ] = ScalabilityId[ i ] [ 0 ]
}

```

direct_dependency_flag[i][j] igual a 0 especifica que la capa con índice j no es una capa de referencia directa para la capa con índice i. direct_dependency_flag[i][j] igual a 1 especifica que la capa con índice j puede ser una capa de

referencia directa para la capa con índice i. Cuando `direct_dependency_flag[i][j]` no se encuentra presente para i y j en el intervalo de 0 a `vps_max_layers_minus1`, se infiere que es igual a 0.

Las variables `NumDirectRefLayers[i]` y `RefLayerId[i][j]` se derivan como sigue:

```
for( i = 1; i <= vps_max_layers_minus1; i++ )
    for( j = 0, NumDirectRefLayers[ i ] = 0; j < i; j++ )
        if( direct_dependency_flag[ i ] [ j ] == 1 )
            RefLayerId [ i ] [ NumDirectRefLayers [ i ]++ ] =
                layer_id_in_nuh[ j ]
```

- 5 Basándose en `RefLayerId[i][j]` y `NumDirectRefLayers[i]`, se construye el llamado conjunto de imágenes de referencia entre capas, como se describe más adelante.

La salida del proceso de decodificación para un conjunto de imágenes de referencia entre capas es una lista actualizada de imágenes entre capas `RefPicSetInterlayer`.

La lista `RefPicSetInterlayer` se vacía en primer lugar y se deriva entonces como sigue:

```
for( i = 0; i < NumDirectRefLayers[ LayerIdInVps[ nuh_layer_id ] ]; i++ ) {
    RefPicSetInterLayer[ i ] = the picture with picture order count
        equal to PicOrderCnt and nuh_layer_id equal to
        RefLayerId[ LayerIdInVps[ nuh_layer_id ] ] [ i ]
    RefPicSetInterLayer[ i ] is marked as "used for long-term reference"
10 }
```

La salida del proceso de marcaje para finalizar la decodificación de una imagen codificada con `nuh_layer_id` mayor que 0 es un marcaje actualizado potencialmente como "usado para referencia cercana" para algunas imágenes decodificadas.

Se aplica lo siguiente:

```
for ( i = 0; i < NumDirectRefLayers[ LayerIdInVps[ nuh_layer_id ] ]; i++ )
15 RefPicSetInterLayer[ i ] is marked as "used for short-term reference"
```

Las imágenes de referencia temporal y las imágenes de referencia entre capas se combinan en dos listas temporales de imágenes de referencia, `RefPicListTemp0` y `RefPicListTemp1`, como se describe más adelante. Finalmente se aplican los comandos de modificación de las listas potenciales de imágenes de referencia y se obtienen las listas finales de imágenes de referencia `RefPicList0` y `RefPicList1`, como se describe más adelante.

- 20 El proceso de decodificación para la construcción de las listas de imágenes de referencia se invoca al comienzo del proceso de decodificación para cada fragmento P o B.

Las imágenes de referencia se direccionan por medio de los índices de referencia como se especifica en la subcláusula 8.5.3.3.2 de la especificación base HEVC (JCTVC-L1003). Un índice de referencia es un índice dentro de una lista de imágenes de referencia. Cuando se decodifica un fragmento P, existe una única lista de imágenes de referencia `RefPicList0`. Cuando se decodifica un fragmento B, existe una segunda lista independiente de imágenes de referencia `RefPicList1` adicional a la `RefPicList0`.

Al comienzo del proceso de decodificación para cada fragmento, las listas de imágenes de referencia `RefPicList0`, y las listas `RefPicList1` para los fragmentos B, se derivan como sigue.

La variable `NumRpsCurrTempList0` se fija igual a

- 30 `Max(num_ref_idx_l0_active_minus1 + 1, NumPocTotalCurr)` y la lista `RefPicListTemp0` se construye como sigue:

```

rIdx = 0
while( rIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0;
        rIdx++, i++ )
        RefPicListTemp0 [ rIdx ] = RefPicSetStCurrBefore [ i ]
    for ( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList0;
        rIdx++, i++ )
        RefPicListTemp0 [ rIdx ] = RefPicSetStCurrAfter[ i ]
    for ( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList0;
        rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetLtCurr[ i ]
    for( i = 0; i < NumDirectRefLayers[ LayerIdInVps[ nuh_layer_id ] ];
        rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetInterLayer[ i ]
}

```

La lista RefPicList0 se construye como sigue:

```

for( rIdx = 0; rIdx <= num_ref_idx_l0_active_minus1; rIdx++ )
    RefPicList0[ rIdx ] = ref_pic_list_modification_flag_l0 ?
        RefPicListTemp0 [ list_entry_l0[ rIdx ] ] : RefPicListTemp0[ rIdx ]

```

Cuando el fragmento es un fragmento B, la variable NumRpsCurrTempList1 se fija igual a:

- 5 Max(num_ref_idx_l1_active_minus1 + 1, NumPocTotalCurr) y la lista RefPicListTemp1 se construye como sigue:

```

rIdx = 0
while( rIdx < NumRpsCurrTempList1 ) {
    for ( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1;
        rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList1;
        rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList1;
        rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetLtCurr[ i ]
    for( i = 0; i < NumDirectRefLayers [ LayerIdInVps[ nuh_layer_id ] ];
        rIdx++, i++ )
        RefPicListTemp1 [ rIdx ] = RefPicSetInterLayer [ i ]
}

```

Cuando el fragmento es un fragmento B, la lista RefPicList1 se construye como sigue:

```

for ( rIdx = 0; rIdx <= num_ref_idx_l1_active_minus1; rIdx++ )
    RefPicList1 [ rIdx ] = ref_pic_list_modification_flag_l1 ?
        RefPicListTemp1 [ list_entry_l1[ rIdx ] ] : RefPicListTemp1 [ rIdx ]

```

- 10 En el procedimiento de inicialización de la lista de imágenes de referencia resumido anteriormente en este documento, las imágenes de referencia entre capas se adjuntan a la lista de imágenes de referencia de acuerdo con el orden de las capas de referencia en la estructura ordenada RefPicSetInterLayer[]. El orden de las capas de referencia en RefPicSetInterLayer[] se fija de acuerdo con el índice de capa i de las capas de referencia, de pequeños a grandes valores de i. Por ello, las imágenes de referencia entre capas en ambas listas iniciales de imágenes de referencia se insertan siempre según el orden creciente del índice de capa i. Este orden no tiene en
- 15 consideración potenciales similitudes o disimilitudes de capas diferentes, y no es por consiguiente lo mejor en términos de eficacia de compresión o de eficacia de velocidad de bits.

Resumen

Es un objetivo de la invención proporcionar una alternativa mejorada a las técnicas reseñadas anteriormente y a la técnica anterior.

Más específicamente, un objetivo de la invención es proporcionar una obstrucción mejorada de listas de imágenes de referencia entre capas y en particular listas iniciales de imágenes de referencia.

Estos y otros objetivos de la invención se logran por medio de diferentes aspectos de la misma, como se definen por medio de las reivindicaciones independientes. Las realizaciones de la invención se caracterizan por medio de las reivindicaciones dependientes.

De acuerdo con un primer aspecto de la invención, se proporciona un método de construir al menos una lista de imágenes de referencia para la predicción entre capas de una imagen actual. El método comprende insertar imágenes de referencia en un primer conjunto de imágenes de referencia o en un segundo conjunto de imágenes de referencia. Las imágenes de referencia se insertan basándose en los respectivos valores de un identificador de la escalabilidad asociado con las imágenes de referencia y de un valor del identificador de la escalabilidad asociado con la imagen actual. El método comprende además el primer conjunto de imágenes de referencia y el segundo conjunto de imágenes de referencia dentro de al menos una lista de imágenes de referencia.

De acuerdo con un segundo aspecto de invención, se proporciona un programa de ordenador. El programa de ordenador comprende instrucciones ejecutables por ordenador para hacer que un dispositivo realice el método de acuerdo con el primer aspecto de la invención cuando se ejecutan las instrucciones ejecutables por ordenador en una unidad de procesamiento incluida en el dispositivo.

De acuerdo con un tercer aspecto de la invención, se proporciona un producto de programa de ordenador. El producto de programa de ordenador comprende un medio de almacenamiento interpretable por ordenador. El medio de almacenamiento interpretable por ordenador tiene un programa de ordenador de acuerdo con el segundo aspecto de la invención incorporada en ese documento.

De acuerdo con un cuarto aspecto de la invención, se proporciona un dispositivo para construir al menos una lista de imágenes de referencia para la predicción entre capas de una imagen actual. El dispositivo comprende medios configurados para insertar imágenes de referencia dentro de un primer conjunto de imágenes de referencia y dentro de un segundo conjunto de imágenes de referencia. Los medios están configurados para insertar las imágenes de referencia basándose en los respectivos valores de un identificador de la escalabilidad asociado con las imágenes de referencia y en un valor del identificador de la escalabilidad asociado con la imagen actual. Los medios están configurados además para insertar el primer conjunto de imágenes de referencia y el segundo conjunto de imágenes de referencia en al menos una lista de imágenes de referencia.

La invención se basa en entender que la construcción de listas de imágenes de referencia se puede mejorar tomando indicaciones de las similitudes entre las capas de referencia y la capa actual dentro de la cuenta. Por ello, se logra una compresión de vídeo multi capa más eficaz. Tales indicaciones de las similitudes las proporcionan los identificadores de la escalabilidad. Con este fin, se insertan las imágenes de referencia entre capas de acuerdo con los respectivos valores de sus identificadores de la escalabilidad cuando se construyen las listas iniciales de imágenes de referencia.

La construcción de las listas de imágenes de referencia propuesta en este documento se puede realizar utilizando un proceso predefinido, por ejemplo, clasificar las imágenes de referencia de acuerdo con los identificadores de la escalabilidad, tanto en un codificador de vídeo como en un decodificador de vídeo. Alternativamente, se pueden construir las listas de imágenes de referencia en el codificador de vídeo y señalizárselas a continuación al decodificador de vídeo, por ejemplo, información adicional, metadatos, por medio de comandos de modificación de la lista de imágenes de referencia o similares y aplicados a continuación al decodificador de vídeo.

De acuerdo con una realización de la invención, las imágenes de referencia se insertan en el primer conjunto de imágenes de referencia o en el segundo conjunto de imágenes de referencia basándose en una diferencia de su valor respectivo del identificador de la escalabilidad y el valor del identificador de la escalabilidad asociado con la imagen actual.

De acuerdo con una realización de la invención, las imágenes de referencia que tienen valores respectivos del identificador de la escalabilidad que son menores que el valor del identificador de la escalabilidad asociado a la imagen actual se insertan dentro del primer conjunto de imágenes de referencia y las imágenes de referencia que tienen valores respectivos del identificador de la escalabilidad que son mayores que el valor del identificador de la escalabilidad asociado con la imagen actual se insertan dentro del segundo conjunto de imágenes de referencia.

De acuerdo con una realización de la invención, las imágenes de referencia comprendidas en el primer conjunto de imágenes de referencia y en el segundo conjunto de imágenes de referencia se ordenan de acuerdo con sus respectivos valores del identificador de la escalabilidad.

De acuerdo con una realización de invención, las imágenes de referencia se ordenan según el valor decreciente del identificador de la escalabilidad.

De acuerdo con una realización de invención, las imágenes de referencia se ordenan según el valor creciente del identificador de la escalabilidad.

De acuerdo con una realización de la invención, las imágenes de referencia se ordenan según la diferencia absoluta decreciente de sus valores respectivos del identificador de la escalabilidad y del valor del identificador de la escalabilidad asociado con la imagen actual.

5 De acuerdo con una realización de la invención, la imagen de referencia se ordena según la diferencia absoluta creciente de sus respectivos valores del identificador de la escalabilidad y del valor del identificador de la escalabilidad asociado con la imagen actual.

10 De acuerdo con una realización de invención, se construyen dos listas de imágenes de referencia, el primer conjunto de imágenes de referencia se inserta dentro de la primera lista de imágenes de referencia al frente del segundo conjunto de imágenes de referencia y el segundo conjunto de imágenes de referencia se inserta dentro de la segunda lista de imágenes de referencia al frente del primer conjunto de imágenes de referencia.

15 De acuerdo con una realización de la invención, el primer conjunto de imágenes de referencia se inserta dentro de la primera lista de imágenes de referencia después de un primer conjunto de imágenes de referencia cercanas y antes de un segundo conjunto de imágenes de referencia cercanas y al final de la segunda lista de imágenes de referencia. Adicionalmente, se inserta el segundo conjunto de imágenes de referencia dentro de la segunda lista de imágenes de referencia después del segundo conjunto de imágenes de referencia cercanas y antes del primer conjunto de imágenes de referencia cercanas y al final de la primera lista de imágenes de referencia.

20 De acuerdo con una realización de la invención, las imágenes de referencia se insertan dentro del primer conjunto de imágenes de referencia y dentro del segundo conjunto de imágenes de referencia basándose en sus respectivos valores de al menos dos identificadores de la escalabilidad, un primer y un segundo identificadores de la escalabilidad, asociados con las imágenes de referencia y los valores respectivos de los identificadores de la escalabilidad asociados con la imagen actual.

25 De acuerdo con una realización de la invención, cada uno de los identificadores de la escalabilidad es cualquiera de un identificador de toma que indique una posición relativa espacial de la captura de la cámara, un identificador de dependencia que indique un orden de descodificación y un indicador de calidad que indique un orden de descodificación de capa, un identificador temporal que indique la información temporal de la escalabilidad o un indicador de profundidad que indique si una capa contiene información de textura o de profundidad.

De acuerdo con una realización de la invención, el método comprende un método de codificar una señal de vídeo.

De acuerdo con una realización de la invención, el método comprende un método de descodificar una señal de vídeo.

30 De acuerdo con una realización de invención, el dispositivo es un codificador de señal de vídeo.

De acuerdo con una realización de la invención, el dispositivo es un descodificador de señal de video.

De acuerdo con una realización de invención, el dispositivo comprende un terminal móvil.

35 Incluso aunque las ventajas de la invención se han descrito en algunos casos con referencia a las realizaciones de un cierto aspecto de la invención, se aplica lo que razonablemente corresponda a realizaciones de otros aspectos de la invención.

Objetivos adicionales, características y ventajas de la invención serán evidentes al estudiar la siguiente descripción detallada, los dibujos y las reivindicaciones adjuntas. Los expertos en la técnica observarán que las diferentes características de la invención se pueden combinar para crear realizaciones diferentes a las descritas a continuación.

40 **Breve descripción de los dibujos**

Las características anteriores, así como los objetivos adicionales, y las ventajas de la invención, se entenderán mejor por medio de la siguiente descripción detallada ilustrativa y no limitadora de las realizaciones de la invención, con referencia a los dibujos adjuntos, en los cuales:

45 La figura 1 ilustra la construcción de un conjunto de imágenes de referencia de acuerdo con una realización de la invención.

La figura 2 muestra un método de construir listas de imágenes de referencia, de acuerdo con una realización de la invención.

La figura 3 muestra un codificador de vídeo, de acuerdo con una realización de la invención.

La figura 4 muestra un descodificador de vídeo, de acuerdo con una realización de la invención.

50 La figura 5 muestra un codificador o descodificador de vídeo, de acuerdo con otra realización de la invención.

La figura 6 muestra un terminal móvil, de acuerdo con una realización de la invención.

Todas las figuras son esquemáticas, no necesariamente a escala, y generalmente sólo muestran partes que son necesarias con objeto de aclarar la invención, en las que otras partes se pueden omitir o son meramente sugeridas.

5 Descripción detallada

Se describirá a continuación la invención de modo completo con referencia a los dibujos que se acompañan, en los cuales se muestran ciertas realizaciones de la invención. La invención puede, sin embargo, realizarse de muchas formas diferentes y no se debe interpretar como que se limita a las realizaciones expuestas en este documento. Más aún, estas realizaciones se proporcionan a modo de ejemplo de modo que esta descripción será más rigurosa y completa, y expresa totalmente el alcance de la invención a los expertos en la materia.

En aras de la claridad, para cada realización las diferencias con respecto a la sintaxis y a la semántica de las especificaciones preliminares actuales, como se describen en la sección Antecedentes, se identifican como sigue.

Construcción de la lista de imágenes de referencia basándose en ViewId

Esta realización se refiere a aquellos casos en los que la capa actual y las capas de referencia difieren en sus respectivos valores de ViewId. El mecanismo propuesto consta de dos etapas:

- Etapa 1: Derivación de dos listas de imágenes de referencia, RefPicSetInterLayerBefore y RefPicSetInterLayerAfter.

Cada imagen de referencia entre tomas dada en la lista RefLayer Id[] se inserta bien en RefPicSetInterLayerBefore o en RefPicSetInterLayerAfter, dependiendo de su valor de ViewId, ViewId_ref, y del valor de ViewId en la imagen actual, ViewId_curr. Si ViewId_ref < ViewId_curr, la imagen se inserta en RefPicSetInterLayerBefore. De lo contrario, se inserta en RefPicSetInterLayerAfter.

Esto se ilustra en 100 en la figura 1, que muestra cinco imágenes, una imagen actual (Curr) y cuatro imágenes de referencia (A, B, C y D) con los respectivos valores de ViewId que se van incrementando de izquierda a derecha en la figura 1. De acuerdo con una realización de la invención, las imágenes de referencia A y B, ilustradas en la figura 1 que tienen los respectivos ViewIds menores que el ViewId de la imagen actual Curr, se insertan dentro de un primer conjunto 101. De acuerdo con ello, las imágenes de referencia C y D, ilustradas en la figura 1 que tienen los respectivos ViewIds mayores que el ViewId de la imagen actual Curr, se insertan dentro de un segundo conjunto 102.

Opcionalmente, las imágenes en RefPicSetInterLayerBefore (por ejemplo, conjunto 101) se clasifican de acuerdo con el valor decreciente de su ViewId, es decir, el orden de las imágenes de referencia en RefPicSetInterLayerBefore es tal que la diferencia absoluta $\text{abs}(\text{ViewId_ref} - \text{ViewId_curr})$ va aumentando. Adicionalmente, las imágenes en RefPicSetInterLayerAfter (por ejemplo, lista 102) se clasifican por el valor creciente de su ViewId, es decir, el orden de las imágenes de referencia en RefPicSetInterLayerAfter es tal que la diferencia absoluta $\text{abs}(\text{ViewId_ref} - \text{ViewId_curr})$ aumenta.

- Etapa 2: Inserción de las imágenes de referencia entre tomas dentro de las listas de imágenes de referencia RefPicListTemp0 y RefPicListTemp1.

RefPicListTemp0 y RefPicListTemp1 son las listas de imágenes de referencia antes de aplicar los comandos de modificación a la lista de imágenes de referencia. Para RefPicListTemp0, las imágenes de RefPicSetInterLayerBefore se insertan de acuerdo con su orden en RefPicSetInterLayerBefore. Inmediatamente, las imágenes de RefPicSetInterLayerAfter se insertan de acuerdo con su orden en RefPicSetInterLayerAfter. Para RefPicListTemp1, las imágenes de RefPicSetInterLayerAfter se insertan de acuerdo con su orden en RefPicSetInterLayerAfter. Inmediatamente, las imágenes de RefPicSetInterLayerBefore se insertan de acuerdo con su orden en RefPicSetInterLayerBefore.

La salida del proceso de decodificación para los conjuntos de imágenes de referencia entre capas son dos listas actualizadas de las imágenes entre capas, RefPicSetInterLayerBefore y RefPicSetInterLayerAfter, así como las variables NumInterLayerBefore y NumInterLayerAfter.

Las listas RefPicSetInterLayerBefore y RefPicSetInterLayerAfter primero se vacían y a continuación se derivan como sigue:

```

for( i = 0, NumInterLayerBefore = 0, NumInterLayerAfter = 0; i <
  NumDirectRefLayers[LayerIdInVps[ nuh_layer_id ] ]; i++ )
  if( ViewId[ RefLayerId[ LayerIdInVps[ nuh_layer_id ] ][ i ] ] <
    ViewId [ nuh_layer_id ] ) {
    for ( j = NumInterLayerBefore; j > 0 &&
      ViewId [ RefLayerId [ LayerIdInVps [ nuh_layer_id ] ]
        [ i ] ] > ViewId [ the value of nuh_layer_id of
          RefPicSetLayerBefore [ j - 1 ] ]; j-- )
      RefPicSetInterLayerBefore [ j ] =
        RefPicSetInterLayerBefore [ j - 1 ]
    RefPicSetInterLayerBefore [ j ] = the picture with picture order
      count equal to PicOrderCnt and nuh_layer_id equal to
        RefLayerId [ LayerIdInVps [ nuh_layer_id ] ] [ i ]
    NumInterLayerBefore++
    RefPicSetInterBefore [ j ] is marked as "used for long-term
      reference"
  } else {
    for( j = NumInterLayerAfter; j > 0 && ViewId [ RefLayerId [
      LayerIdInVps [ nuh_layer_id ] ] [ i ] ] <
      ViewId [ the value of nuh_layer_id of
        RefPicSetLayerAfter [ j - 1 ] ]; j-- )
      RefPicSetInterLayerAfter [ j ] =
        RefPicSetInterLayerAfter [ j - 1 ]
    RefPicSetInterLayerAfter [ j ] = the picture with picture order
      count equal to PicOrderCnt and nuh_layer_id equal to
        RefLayerId [ LayerIdInVps [ nuh_layer_id ] ] [ i ]
    NumInterLayerAfter++
    RefPicSetInterAfter [ j ] is marked as "used for long-term
      reference"
  }
}

```

La salida del procedimiento de marcaje para finalizar la descodificación de un imagen codificada con nuh_layer_id mayor que 0 es un marcaje potencialmente actualizado como "utilizado para referencia cercana" para algunas imágenes descodificadas. Se aplica lo siguiente:

5

```

for( i = 0; i < NumInterLayerBefore; i++ )
  RefPicSetInterLayerBefore [ i ] is marked as "used for short-term
    reference"
  for ( i = 0; i < NumInterLayerAfter; i++ )
    RefPicSetInterLayerAfter [ i ] is marked as "used for short-term
      reference"

```

El proceso de descodificación para la construcción de las listas de imágenes de referencia se invoca al comienzo del proceso de descodificación para cada fragmento P o B.

10

Al comienzo del proceso de descodificación para cada fragmento, la lista de imágenes de referencia RefPicList0, y para los fragmentos B la lista RefPicList1, se derivan como sigue.

La variable NumRpsCurrTempList0 se fija igual a Max(num_ref_idx_l0_active_minus1 + 1, NumPocTotalCurr) y la lista RefPicListTemp0 se construye como sigue:

```

rIdx = 0
while( rIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0;
        rIdx++, i++ )
        RefPicListTemp0 [ rIdx ] = RefPicSetStCurrBefore [ i ]
    for ( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList0;
        rIdx++, i++ )
        RefPicListTemp0 [ rIdx ] = RefPicSetStCurrAfter [ i ]
    for ( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList0;
        rIdx++, i++ )
        RefPicListTemp0 [ rIdx ] = RefPicSetLtCurr [ i ]
    for ( i = 0; i < NumInterLayerBefore; rIdx++, i++ )
        RefPicListTemp0 [ rIdx ] = RefPicSetInterLayerBefore [ i ]
    for ( i = 0; i < NumInterLayerAfter; rIdx++, i++ )
        RefPicListTemp0 [ rIdx ] = RefPicSetInterLayerAfter [ i ]
}

```

La lista RefPicList0 se construye como sigue:

```

for ( rIdx = 0; rIdx <= num_ref_idx_l0_active_minus1; rIdx++)
    RefPicList0 [ rIdx ] = ref_pic_list_modification_flag_l0 ?
        RefPicListTemp0 [ list_entry_l0 [ rIdx ] ] : RefPicListTemp0 [ rIdx ]

```

- 5 Cuando el fragmento es un fragmento B, la variable NumRpsCurrTempList1 se fija igual a Max(num_ref_idx_l1_active_minus1 + 1, NumPocTotalCurr) y la lista RefPicListTemp1 se construye como sigue:

```

rIdx = 0
while ( rIdx < NumRpsCurrTempList1 ) {
    for ( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1;
        rIdx++, i++ )
        RefPicListTemp1 [ rIdx ] = RefPicSetStCurrAfter [ i ]

    for ( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList1;
        rIdx++, i++ )
        RefPicListTemp1 [ rIdx ] = RefPicSetStCurrBefore [ i ]
    for ( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList1;
        rIdx++, i++ )
        RefPicListTemp1 [ rIdx ] = RefPicSetLtCurr [ i ]
    for ( i = 0; i < NumInterLayerAfter; rIdx++, i++ )
        RefPicListTemp1 [ rIdx ] = RefPicSetInterLayerAfter [ i ]
    for ( i = 0; i < NumInterLayerBefore; rIdx++, i++ )
        RefPicListTemp1 [ rIdx ] = RefPicSetInterLayerBefore [ i ]
}

```

Cuando el fragmento es un fragmento B, la lista RefPicList1 se construye como sigue:

```

for ( rIdx = 0; rIdx <= num_ref_idx_l1_active_minus1; rIdx++)
    RefPicList1 [ rIdx ] = ref_pic_list_modification_flag_l1 ?
        RefPicListTemp1 [ list_entry_l1 [ rIdx ] ] : RefPicListTemp1 [ rIdx ]

```

- 10 De acuerdo con la sintaxis de la extensión VPS actual referida en la sección Antecedentes, ViewId se señala sólo para aquellas capas con un índice de capa mayor de 0. La capa base (índice de capa 0) tiene siempre todas las dimensiones de la escalabilidad igual a 0, es decir, se infiere que ViewId es igual a 0. También, para el índice de capa mayor de 0, el valor de ViewId es siempre no negativo.

- 15 De acuerdo con el proceso propuesto descrito anteriormente, ViewId se interpreta como una posición espacial de la cámara asociada con una toma. Si la toma base tiene ViewId igual a 0, entonces se restringiría a ser sólo la toma “más a la izquierda” o “más a la derecha”. Con objeto de superar esta restricción, se puede introducir una señalización más flexible y se describen a continuación diversas alternativas.

- Alternativa A: Enviar dimension_id para el índice de capa igual a 0

vps_extension() {	Descriptor
while(!byte_aligned() {	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0, NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
vps_nuh_layer_id_present_flag	u(1)
for(i = 0; i <= vps_max_layers_minus1; i++) {	
if(vps_nuh_layer_id_present_flag && i > 0)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id [i] [j]	u(v)
}	
...	

- Alternativa B: Alternativa A con la presencia opcional de dimension_id para la capa base.

Los elementos sintácticos de dimension_id pueden estar presentes en la capa base, opcionalmente obstaculizada por un indicador. Podría existir un indicador que indique la presencia de dimension_id para la capa base para todos los tipos de escalabilidad.

Alternativamente, para cada tipo de escalabilidad podría existir un indicador separado que indique si dimension_id para la capa base está presente para ese tipo de escalabilidad.

- Alternativa C: interpretar ViewId como un entero con signo

En este caso, ViewId para la capa base se deriva todavía para que sea igual a 0. Sin embargo, ViewId para las capas más altas se deriva de tal manera que un bit de su valor se interpreta como un signo, por ejemplo, el bit menos significativo o bit más significativo, y los restantes bits se interpretan como magnitud (menos 1). Esto se ilustra en el siguiente pseudo código, en el que el bit menos significativo se interpreta como signo.

La variable ScalabilityId[i][smlIdx], que especifica el identificador del tipo de dimensión de la escalabilidad smlIdx-ésimo de la capa i-ésima, y la variable ViewId[layer_id_in_nuh[i]], que especifica el identificador de toma de la capa i-ésima, se derivan como sigue:

```

for ( i = 0; i <= vps_max_layers_minus1; i++ ) {
    for( smIdx= 0, j =0; smIdx< 16; smIdx ++ )
        if( ( i != 0 ) && scalability_mask [ smIdx ] )
            ScalabilityId [ i ] [ smIdx ] = dimension_id [ i ] [ j++ ]
        else
            ScalabilityId [ i ] [ smIdx ] = 0
    ViewId[ layer_id_in_nuh[ i ] ] = ( i == 0 ) ? 0 :
        ( ScalabilityId [ i ] [ 0 ] & 1 ? -1 : 1 ) * ( ( ScalabilityId [ i ] [ 0 ]
            » 1 ) + 1 )
}

```

Construcción de la lista de imágenes de referencia basándose en DependencyId

Esta realización se refiere a aquellos casos en los que la capa actual y las capas de referencia difieren en los valores de DependencyId. El mecanismo propuesto consta de dos etapas

- Etapa 1: Derivación de la lista RefPicSetInterLayer.

- 5 Cada imagen de referencia entre capas dada en la lista RefLayerId[] se inserta en RefPicSetInterLayer. Las imágenes en RefPicSetInterLayer se clasifican por el valor decreciente de su respectiva DependencyId. Por ello, con DependencyId_ref siendo el valor de DependencyId de la imagen de referencia, y siendo DependencyId-curr el valor de DependencyId de la imagen actual, el orden de las imágenes de referencia en RefPicSetInterLayer es tal que la diferencia absoluta $\text{abs}(\text{DependencyId_ref} - \text{DependencyId_curr})$ va aumentando.

- 10 - Etapa 2: Inserción de las imágenes de referencia entre tomas dentro de las listas de imágenes de referencia RefPicListTemp0 y RefPicListTemp1.

La etapa 2 no cambia al compararla con la especificación provisional referida en la sección Antecedentes.

La salida del proceso de decodificación para los conjuntos de imágenes de referencia entre capas es una lista actualizada de imágenes entre capas, RefPicSetInterLayer. La lista RefPicSetInterLayer se vacía primero y a continuación se deriva como sigue:

- 15
- ```

for (i = 0; i < NumDirectRefLayers [LayerIdInVps [nuh_layer_id]]; i++)
 for (j = i; j > 0 && DependencyId[RefLayerId[
 LayerIdInVps [nuh_layer_id]] [i]] >
 DependencyId [the value of nuh_layer_id of
 RefPicSetLayer [j - 1]]; j--)
 RefPicSetInterLayer [j] = RefPicSetInterLayer [j - 1]
 RefPicSetInterLayer [j] = the picture with picture order count equal
 to PicOrderCnt and nuh_layer_id equal to
 RefLayerId [LayerIdInVps [nuh_layer_id]] [i]
 RefPicSetInterLayer [j] is marked as "used for long-term reference"

```

De otra manera, la derivación se hace de la misma forma como se describió en la sección Antecedentes.

**Construcción de la lista de imágenes de referencia basándose en una combinación de DependencyId y QualityId**

- 20 Esta realización se refiere a aquellos casos en que la capa actual y las capas de referencia difieren en sus respectivos valores de DependencyId, y adicionalmente pueden diferir en sus respectivos valores de QualityId. La solución propuesta consta de dos etapas. - Etapa 1: Derivación de la lista RefPicSetInterLayer.

- 25 Cada imagen de referencia entre capas dada en la lista RefLayerId[] se inserta en RefPicSetInterLayer. Las imágenes en RefPicSetInterLayer se clasifican en primer lugar por medio del valor decreciente de su DependencyId, y en segundo lugar se clasifican por medio del valor decreciente de su QualityId. Es decir, las imágenes se seleccionan por medio de DependencyId, y si DependencyId es idéntico para unas imágenes en la lista, entonces aquellas imágenes se clasifican por medio del valor decreciente de QualityId. - Etapa 2: Inserción de las imágenes de referencia entre tomas dentro de las listas de imágenes de referencia RefPicListTemp0 y RefPicListTemp1.

La etapa 2 no cambia comparándola con la especificación provisional referida a ella en la sección Antecedentes.

- 30 **Construcción de la lista de imágenes de referencia basándose en una combinación de DependencyId y ViewId**

Esta realización se refiere a aquellos casos en los que la capa actual y las capas de referencia difieren en sus respectivos valores de DependencyId y/o ViewId. Se pueden combinar otros identificadores de la escalabilidad de modo similar. La solución propuesta consta de dos etapas.

- 35 - Etapa 1: Derivación de las listas RefPicSetInterDependency, RefPicSetInterViewBefore, RefPicSetInterViewAfter y RefPicSetInterDependencyView.

Cada imagen de referencia entre capas dada en la lista RefLayerId[] se inserta en RefPicSetInterDependency, en RefPicSetInterViewBefore, en RefPicSetInterViewAfter o en RefPicSetInterDependencyView.

- 40 Las imágenes de referencia que difieren de la imagen actual en sus respectivos valores de DependencyId, pero no difieren en su respectivo valor de ViewId, se insertan en RefPicSetInterDependency.

Las imágenes de referencia que difieren de la imagen actual en sus respectivos valores de ViewId, pero no difieren en su respectivo valor de DependencyId, se insertan en RefPicSetInterViewBefore o en RefPicSetInterViewAfter, dependiendo de su valor de ViewId comparado con el valor de ViewId de la imagen actual, similar a lo que se ha descrito anteriormente en este documento.

- 5 Las imágenes de referencia que difieren de la imagen actual en sus respectivos valores de ViewId y DependencyId se insertan en RefPicSetInterDependencyView. Estas imágenes se pueden repartir alternativamente en dos listas, dependiendo de su valor de ViewId comparado con el valor de ViewId de la imagen actual, similar a lo que se ha descrito anteriormente en este documento.

- 10 Las imágenes de RefPicSetInterDependency se clasifican según el valor decreciente de su DependencyId, similar a lo que se ha descrito anteriormente en este documento.

Las imágenes en RefPicSetInterViewBefore se clasifican según el valor decreciente de su ViewId, similar a lo que se ha descrito anteriormente en este documento.

Las imágenes en RefPicSetInterViewAfter se clasifican según el valor decreciente de su ViewId, similar a lo que se ha descrito anteriormente en este documento.

- 15 Las imágenes en RefPicSetInterDependencyView se clasifican mediante una combinación de sus valores de DependencyId y ViewId, DependencyId-ref y ViewId\_ref, comparados con los valores de DependencyId y ViewId de la imagen actual, DependencyId-curr y ViewId\_curr. Por ejemplo, la combinación de valores mencionada anteriormente puede ser una suma de las diferencias absolutas ( $\text{abs}(\text{ViewId\_ref} - \text{ViewId\_curr}) + \text{abs}(\text{DependencyId\_ref} - \text{DependencyId\_curr})$ ), o una suma de los cuadrados de las diferencias  $(\text{ViewId\_ref} - \text{ViewId\_curr})^2 + (\text{DependencyId\_ref} - \text{DependencyId\_curr})^2$ , o combinaciones similares.

- 20 - Etapa 2: Inserción de imágenes de referencia entre tomas dentro de las listas de imágenes de referencia RefPicListTemp0 y RefPicListTemp1.

- 25 RefPicSetInterDependency, RefPicSetInterViewBefore, RefPicSetInterViewAfter y RefPicSetInterDependency-View se insertan separadamente en las listas de imágenes de referencia, por ejemplo, según el orden RefPicSetInterViewBefore, RefPicSetInterViewAfter, RefPicSetInterDependency y RefPicSetInterDependency-View.

### **Construcción de una lista de imágenes de referencia basándose en una combinación de DependencyId y POC o ViewId y POC**

- 30 Esta realización se refiere a aquellos casos en los que la capa actual y las capas de referencia difieren en sus respectivos valores de DependencyId, y pueden adicionalmente diferir en su respectivo valor de Cuenta del Orden de la Imagen (POC). De manera similar, las capas pueden diferir en los respectivos valores de, por ejemplo, ViewId, o algún otro identificador de la escalabilidad y pueden diferir adicionalmente en su valor de POC. Este es, por ejemplo, el caso en el que la predicción entre capas se utiliza pero no se restringe a la predicción entre capas del mismo POC o unidad de acceso. Es decir, se puede utilizar la predicción de imágenes en otras capas que no estén alineadas temporalmente con la imagen actual. Dado que la predicción en este caso es tanto a través de las capas
- 35 (predicción entre capas) como a través del tiempo (predicción temporal), por ejemplo, el abarcar dos dimensiones más que una, se refiere a veces como predicción "diagonal".

Con objeto de realizar la predicción diagonal en un esquema de codificación de vídeo, se requiere una indicación y/o un acuerdo entre codificador y decodificador como para qué casos temporales, por ejemplo, un conjunto de valores de POC, de la capa de referencia se usen para la predicción. A continuación, se proponen diversas alternativas.

- 40 Para cada capa de referencia, una lista de valores de POC o de diferencias de POC, es decir, diferencias entre la POC de una imagen de referencia y la POC de una imagen actual, para las imágenes de referencia que se tienen que usar de la capa de referencia se indica, por ejemplo, en un Conjunto de Parámetros Secuenciales (SPS) o un encabezamiento de fragmentos y codificado en un flujo de datos. El decodificador puede entonces utilizar los valores indicados para construir la lista de imágenes de referencia.

- 45 Alternativamente, un número N de imágenes de referencia, indicado por los valores de POC, utilizado para la predicción de la capa de referencia se indica, por ejemplo, en un SPS o en un encabezamiento de fragmentos y codificado en un flujo de datos. Basándose en el número, el decodificador identifica la lista de los valores de POC o de la diferencias de POC, con respecto al POC de la imagen actual, utilizados para la predicción, y utiliza los valores para construir la lista de imágenes de referencia.

- 50 La identificación de una lista de los valores de POC o de la diferencia de POC puede, por ejemplo, incluir añadir las N imágenes previamente decodificadas en la lista de imágenes de referencia. La identificación de una lista de valores de POC o de diferencia de POC puede incluir alternativamente utilizar los mismos valores de la POC o las mismas diferencias de la POC que se están utilizando para la predicción temporal en la toma de referencia o en la toma básica. Además, se puede utilizar un subconjunto de los valores, por ejemplo, los N primeros valores de POC o
- 55 de diferencia de POC que se usan para la predicción temporal en la toma de referencia o en la toma básica.

La solución propuesta comprende dos etapas.

- Etapa 1: Derivación de la lista RefPicSetInterLayer.

Las imágenes de referencia se insertan en RefPicSetInterLayer y se clasifican mediante una combinación de sus valores de DependencyId y POC, DependencyId\_ref POC\_ref, comparados con los valores de DependencyId y POC de la imagen actual, DependencyId\_curr y POC\_curr. La combinación mencionada anteriormente de valores puede ser una suma de las diferencias absolutas ( $\text{abs}(\text{POC\_ref} - \text{POC\_curr}) + \text{abs}(\text{DependencyId\_ref} - \text{DependencyId\_curr})$ ), una suma de los cuadrados de las diferencias  $(\text{ViewId\_ref} - \text{ViewId\_curr})^2 + (\text{DependencyId\_ref} - \text{DependencyId\_curr})^2$ , o combinaciones similares.

Alternativamente, las imágenes en RefPicSetInterLayer se pueden clasificar en primer lugar aumentando  $\text{abs}(\text{POC\_ref} - \text{POC\_curr})$  y en segundo lugar aumentando  $\text{abs}(\text{DependencyId\_ref} - \text{DependencyId\_curr})$ , o en primer lugar aumentando  $\text{abs}(\text{DependencyId\_ref} - \text{DependencyId\_curr})$  y en segundo lugar aumentando  $\text{abs}(\text{POC\_ref} - \text{POC\_curr})$ . Aquí, clasificar en primer lugar una lista por "X" y clasificar en segundo lugar por "Y" significa que la lista se clasifica por "X", y si las imágenes tienen el mismo valor de "X", estas imágenes se clasifican por "Y".

Alternativamente, las imágenes se puede dividir en tres listas, RefPicSetInterLayerPOCBefore, RefPicSetInterLayerPOCCurr y RefPicSetInterLayerPOCAfter, de modo que RefPicSetInterLayerPOCBefore contiene imágenes que tienen una POC menor que la POC de la imagen actual, RefPicSetInterLayerPOCCurr contiene imágenes que tienen una POC idéntica a la POC de la imagen actual y RefPicSetInterLayerPOCAfter contiene imágenes que tienen una POC mayor que la POC de la imagen actual. Cada una de las tres listas se puede clasificar aumentando la diferencia de la POC comparada con la POC de las imágenes de referencia.

- Etapa 2: Inserción de las imágenes de referencia entre tomas dentro de las listas de imágenes de referencia RefPicListTemp0 y RefPicListTemp1.

RefPicSetInterLayer se inserta en las listas de imágenes de referencia. RefPicSetInterLayerPOCBefore, RefPicSetInterLayerPOCCurr y RefPicSetInterLayerPOCAfter, se insertan en las listas de imágenes de referencia, por ejemplo, según el orden RefPicSetInterLayerPOCBefore, RefPicSetInterLayerPOCCurr, RefPicSetInterLayerPOCAfter para RefPicListTemp0 y según el orden RefPicSetInterLayerPOCAfter, RefPicSetInterLayerPOCCurr, RefPicSetInterLayerPOCBefore para RefPicListTemp1.

### Construcción de la lista de imágenes de referencia desde el lado del codificador

En esta realización el codificador calcula un orden de las listas de imágenes de referencia de acuerdo con una de las realizaciones previas y envía el orden resultante, por ejemplo, como listas de imágenes de referencia, explícitamente al decodificador. La señalización se puede hacer de acuerdo con los comandos de modificación de la lista de imágenes de referencia. Sin embargo, los comandos de modificación de la lista de imágenes de referencia pueden ser algo ineficaces ya que se envían en el encabezamiento del fragmento ya que pueden tener que ser enviados a todas las imágenes de referencia, no sólo a las imágenes de referencia entre tomas.

Por ello, se puede utilizar una señalización alternativa, por ejemplo, en un Conjunto de Parámetros Secuenciales (SPS) o en un Conjunto de Parámetros de la Imagen (PPS), como se indica a continuación.

```
num_ref_layers_10 {coded as ue(v)}
num_ref_layers_11 {coded as ue(v)}
if(num_ref_layers_10 > 0 || num_ref_layers_11 > 0) {
 log2_max_num_ref_layers {coded as ue(v)}
 if(log2_max_num_ref_layers > 0) {
 for(i=0; i < num_ref_layers_10; i++)
 ref_layer_idx_10[j] {coded as u(v) with
 v=log2_max_num_ref_layers, or v = 6}
 for(i=0; i < num_ref_layers_11; i++)
 ref_layer_idx_11[j] {coded as u(v) with
 v=log2_max_num_ref_layers, or v = 6}
 }
}
```

Aquí ref\_layer\_idx\_10 y ref\_layer\_idx\_11 pueden especificar índices dentro de una lista de capas de referencia RefLayers[] como se especificó en la sección Antecedentes. Alternativamente

```

max_num_ref_layers (coded as ue(v))
if (max_num_ref_layers > 0) {
 num_ref_layers_l0_minus1 (coded as ue(v))
 num_ref_layers_l1_minus1 (coded as ue(v))
 if(max_num_ref_layers > 1) {
 for (i=0; i <= num_ref_layers_l0_minus1; i++)
 ref_layer_idx_l0[j] (coded as u(v) with
 v=log2_max_num_ref_layers, or v = 6)
 for(i=0; i <= num_ref_layers_l1_minus1; i++)
 ref_layer_idx_l1[j] (coded as u(v) with
 v=log2_max_num_ref_layers, or v = 6)
 }
}

```

Aquí `ref_layer_idx_l0` y `ref_layer_idx_l1` pueden especificar índices dentro de una lista de capas de referencia `RefLayers[]` como se especificó en la sección Antecedentes.

### Construcción de una lista de imágenes de referencia basándose en la distancia entre tomas

- 5 Esta realización se refiere a aquellos casos en los que la capa actual y las capas de referencia pertenecen a diferentes posiciones de la cámara. Las posiciones relativas de la cámara se pueden determinar a partir de los parámetros enviados al descodificador, tales como la distancia de la línea de la base o un parámetro de escala utilizado en la norma provisional 3D-HEVC actual, los cuales se calculan usualmente basándose en el parámetro de la línea de la base de la cámara. La solución propuesta comprende dos etapas.-Etapa 1: Derivación de dos listas
- 10 `RefPicSetInterLayerLeft` y `RefPicSetInterLayerRight`.

Cada imagen de referencia entre tomas comprendida en la lista `RefLayerId[]` se inserta en `RefPicSetInterLayerLeft` o en `RefPicSetInterLayerRight`, dependiendo de su respectivo valor del parámetro de la línea de la base, o de un parámetro derivado de la línea de la base. Si el parámetro de la escala correspondiente a la imagen de referencia y a la imagen actual es menor que 0, la imagen de referencia se inserta en la lista `RefPicSetInterLayerLeft`, y si es al

15 contrario la imagen de referencia se inserta en la lista `RefPicSetInterLayerRight`. Las imágenes en `RefPicSetInterLayerLeft` se clasifican por orden decreciente de los parámetros de la escala entre la cámara actual (toma) y las cámaras que corresponden a las imágenes de referencia. Las imágenes en `RefPicSetInterLayerRight` se clasifican según el orden de valores crecientes de los parámetros de la escala entre las imágenes de referencia y la posición actual de la cámara (imagen).-Etapa 2: Inserción de las imágenes de referencia entre tomas en las listas de

20 imágenes de referencia `RefPicListTemp0` y `RefPicListTemp1`.

`RefPicListTemp0` y `RefPicListTemp1` son las listas de imágenes de referencia antes de aplicar los comandos de modificación de la lista de imágenes de referencia. Las imágenes de `RefPicSetInterLayerBefore` se pueden insertar separadamente de las imágenes de `RefPicSetInterLayerAfter` (por ejemplo, en diferentes posiciones en la lista).

- 25 La salida del proceso de descodificación para los conjuntos de imágenes de referencia entre capas consiste en dos listas actualizadas de imágenes entre capas, `RefPicSetInterLayerLeft` y `RefPicSetInterLayerRight`, así como las variables `NumInterLayerLeft` y `NumInterLayerRight`.

Las listas `RefPicSetInterLayerLeft` y `RefPicSetInterLayerRight` se vacían primero y a continuación se derivan como sigue:

```

for(i = 0, NumInterLayerLeft = 0, NumInterLayerRight = 0; i <
 NumDirectRefLayers[LayerIdInVps[nuh_layer_id]]; i++)
 if(scale [ViewId[[LayerIdInVps[nuh_layer_id]][i]] ,
 ViewId[nuh_layer_id]] < 0) {
 for(j = NumInterLayerLeft; j > 0 && scale [ViewId[
 RefLayerId[LayerIdInVps[nuh_layer_id]][i]]] [
 ViewId[the value of nuh_layer_id of
 RefPicSetLayerLeft[j - 1]] < 0]; j--)
 RefPicSetInterLayerLeft[j] =
 RefPicSetInterLayerLeft[j - 1]
 RefPicSetInterLayerLeft[j] = the picture with picture order
 count equal to PicOrderCnt and nuh_layer_id equal to
 RefLayerId[LayerIdInVps[nuh_layer_id]][i]
 NumInterLayerLeft++
 RefPicSetInterLayerLeft[j] is marked as "used for long-term
 reference"
 } else {

```



```

for(j = NumInterLayerRight; j > 0 && scale[ViewId[
 RefLayerId[LayerIdInVps[nuh_layer_id]][i]] ,
 ViewId[the value of nuh_layer_id of
 RefPicSetLayerRight[j - 1]] } > 0 ; j--)
 RefPicSetInterLayerRight[j] =
 RefPicSetInterLayerRight[j - 1]
 RefPicSetInterLayerRight[j] = the picture with picture
 order count equal to PicOrderCnt and nuh_layer_id equal
 to RefLayerId[LayerIdInVps[nuh_layer_id]][i]
 NumInterLayerRight++
 RefPicSetInterRight[j] is marked as "used for long-term
 reference"
}

```

La salida del procedimiento de marcaje para finalizar la descodificación de una imagen codificada con nuh\_layer\_id mayor de cero es un marcaje actualizado potencialmente como "utilizado para referencia cercana" para algunas imágenes descodificadas. Se aplica lo siguiente

```

for(i = 0; i < NumInterLayerLeft; i++)
 RefPicSetInterLayerLeft[i] is marked as "used for short-term
 reference"
for(i = 0; i < NumInterLayerRight; i++)
 RefPicSetInterLayerRight[i] is marked as "used for short-term
 reference"

```

5

El siguiente procedimiento de descodificación para la construcción de listas de imágenes de referencia se invoca al principio del procedimiento de descodificación para cada fragmento P o B.

Cuando se descodifica un fragmento P, existe una única lista de imágenes de referencia RefPicList0. Cuando se descodifica un fragmento B, existe una segunda lista de imágenes de referencia independiente RefPicList1 adicional a la RefPicList0. Al comienzo del procedimiento de descodificación para cada fragmento, la lista de imágenes de referencia RefPicList0, y RefPicList1 para los fragmentos B, se derivan como sigue.

10

La variable NumRpsCurrTempList0 es igual a Max(num\_ref\_idx\_l0\_active\_minus1 + 1, NumPocTotalCurr) y la lista RefPicListTemp0 se construye como sigue:

```

rIdx = 0
while(rIdx < NumRpsCurrTempList0) {
 for(i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0;
 rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetStCurrBefore[i]
 for(i = 0; i < NumInterLayerLeft; rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetInterLayerLeft[i]
 for(i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList0;
 rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetStCurrAfter[i]
 for(i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList0;
 rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetLtCurr[i]
 for(i = 0; i < NumInterLayerRight; rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetInterLayerRight[i]
}

```

15 La lista RefPicList0 se construye como sigue:

```

for(rIdx = 0; rIdx <= num_ref_idx_l0_active_minus1; rIdx++)
 RefPicList0[rIdx] = ref_pic_list_modification_flag_l0 ?
 RefPicListTemp0[list_entry_l0[rIdx]] : RefPicListTemp0[rIdx]

```

Cuando el fragmento es un fragmento B, la variable NumRpsCurrTempList1 se fija igual a Max(num\_ref\_idx\_l1\_active\_minus1 + 1, NumPocTotalCurr) y la lista RefPicListTemp1 se construye como sigue:

```

rIdx = 0
while(rIdx < IdumRpsCurrTemplList1) {
 for(i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTemplList1;

 rIdx++, i++)
 RefPicListTempl[rIdx] = RefPicSetStCurrAfter[i]
 for(i = 0; i < NumInterLayerRight; rIdx++, i++)
 RefPicListTempl[rIdx] = RefPicSetInterLayerRight[i]
 for(i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTemplList1;

 rIdx++, i++)
 RefPicListTempl[rIdx] = RefPicSetStCurrBefore[i]
 for(i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTemplList1;

 rIdx++, i++)
 RefPicListTempl[rIdx] = RefPicSetLtCurr[i]
 for(i = 0; i < NumInterLayerLeft; rIdx++, i++)
 RefPicListTempl[rIdx] = RefPicSetInterLayerLeft[i]
}

```

En cuando el fragmento es n fragmento B, la lista RefPicList1 se construye como sigue:

```

for(rIdx = 0; rIdx <= num_ref_idx_ll_active_minus1; rIdx++)
 RefPicList1[rIdx] = ref_pic_list_modification_flag_ll ?
 RefPicListTempl[list_entry_ll[rIdx]] : RefPicListTempl[rIdx]

```

- 5 La variable ScalabilityId[i] [smIdx] que especifica el identificador del tipo de dimensión de la escalabilidad smIdx-ésima de la capa i-ésima y la variable ViewId[layer\_in\_id\_nuh[i] que especifica el identificador de la toma de la capa i-ésima se derivan como sigue:

```

for (i = 0; i <= vps_max_layers_minus1; i++) {
 for(smIdx= 0; j =0; smIdx< 16; smIdx ++)
 if((i != 0) && scalability_mask[smIdx])
 ScalabilityId[i][smIdx] = dimension_id[i][j++]
 else
 ScalabilityId[i][smIdx] = 0
 ViewId[layer_id_in_nuh[i]] = (i == 0) ? 0 :
 (ScalabilityId[i][0] & 1 ? -1 : 1) *
 ((ScalabilityId[i][0] >> 1) + 1)
}

```

## 10 Construcción de la lista de imágenes de referencia basándose en ViewId y creación de una lista de imágenes de referencia

Esta realización se refiere a aquellos casos en los que la capa actual y las capas de referencia difieren en sus respectivos valores de ViewId. La solución propuesta consta de dos etapas.- Etapa 1: Derivación de dos listas RefPicSetInterLayerLeft y RefPicSetInterLayerRight.

- 15 Cada imagen de referencia entre tomas dada en la lista RefLayerId[] se inserta en RefPicSetInterLayerLeft o en RefPicSetInterLayerRight, dependiendo de su valor de ViewId, View\_ref y el valor de ViewId de la imagen actual, ViewId\_curr. Si View\_ref < ViewId\_curr, se inserta en RefPicSetInterLayerLeft, si no, se inserta en RefPicSetInterLayerRight.

- 20 Opcionalmente, las imágenes en RefPicSetInterLayerLeft se clasifican por el valor decreciente de su ViewId, es decir, el orden de las imágenes de referencia en RefPicSetInterLayerLeft es tal que la diferencia absoluta abs(View\_ref - ViewId\_curr) aumenta. Adicionalmente, las imágenes en RefPicSetInterLayerRight se clasifican por el valor creciente de su ViewId, es decir, el orden de las imágenes de referencia en RefPicSetInterLayerRight es tal que la diferencia absoluta abs(View\_ref - ViewId\_curr) aumenta. – Etapa 2: Inserción de las imágenes de referencia entre tomas en las listas de imágenes de referencia RefPicListTemp0 y RefPicListTemp1.

- 25 RefPicListTemp0 y RefPicListTemp1 son las listas de imágenes de referencia antes de aplicar los comandos de modificación de la lista de imágenes de referencia. Para RefPicListTemp0, las imágenes de RefPicSetInterLayerLeft se insertan de acuerdo con su orden en RefPicSetInterLayerLeft. A continuación, las imágenes de RefPicSetInterLayerRight se insertan de acuerdo con su orden en RefPicSetInterLayerRight. Para RefPicListTemp1, las imágenes de RefPicSetInterLayerRight se insertan de acuerdo con su orden en RefPicSetInterLayerRight.

A continuación, las imágenes de RefPicSetInterLayerLeft se insertan de acuerdo con su orden en RefPicSetInterLayerLeft.

Se propone la siguiente sintaxis de la extensión SPS:

|                                                |            |
|------------------------------------------------|------------|
| sps_extension( ) {                             | Descriptor |
| <b>inter_view_mv_vert_constraint_flag</b>      | u(1)       |
| sps_extension( ) {                             | Descriptor |
| <b>inter_view_ref_picture_list_right_first</b> | u(1)       |
| }                                              |            |

5 Aquí, inter\_view\_ref\_picture\_list\_right\_first\_flag igual a 1 fija la variable interViewRefPictureListRightFirst igual a 1. Si no, la variable interViewRefPictureListRightFirst es igual a 0. La variable interViewRefPictureListRightFirst permuta las posiciones en las cuales las listas RefPicSetInterLayerLeft y RefPicSetInterLayerRight se insertan en NumRpsCurrTempList0 y en NumRpsCurrTempList1.

10 Alternativamente, el indicador inter\_view\_ref\_picture\_list\_right\_first\_flag puede ser señalizado en un encabezamiento de fragmento o en un PPS. Adicionalmente, se puede señalar un indicador con objeto de especificar si inter\_view\_ref\_picture\_list\_right\_first\_flag está señalizado a nivel de fragmento o a nivel de SPS.

Se propone la siguiente sintaxis de extensión SPS alternativa:

|                                                     |            |
|-----------------------------------------------------|------------|
| sps_extension( ) {                                  | Descriptor |
| <b>inter_view_mv_vert_constraint_flag</b>           | u(1)       |
| <b>inter_view_ref_picture_list_right_first_flag</b> | u(1)       |
| }                                                   |            |

15 Correspondientemente, se propone la siguiente sintaxis de la porción del fragmento:

|                                                                           |            |
|---------------------------------------------------------------------------|------------|
|                                                                           | Descriptor |
| <b>inter_view_ref_picture_list_right_first_flag_in_slice_header</b>       | u(1)       |
| <b>if (inter_view_ref_picture_list_right_first_flag_in_slice_header )</b> | u(1)       |
| <b>inter_view_ref_picture_list_right_first_slice_flag</b>                 |            |

Si el indicador inter\_view\_ref\_picture\_list\_right\_first\_flag está señalizado a nivel de fragmento, el valor del indicador en un encabezamiento del fragmento sobrescribe el valor del indicador en un SPS.

20 La salida del proceso de decodificación para los conjuntos de imágenes de referencia entre capas son dos listas actualizadas de imágenes entre capas, RefPicSetInterLayerLeft y RefPicSetInterLayerRight, y las variables NumInterLayerBefore y NumInterLayerAfter.

Las listas RefPicSetInterLayerLeft y RefPicSetInterLayerRight se vacían primero y a continuación se derivan como sigue:

```

for(i = 0, NumInterLayerLeft = 0, NumInterLayerAfter = 0; i <
 NumDirectRefLayers[LayerIdInVps[nuh_layer_id]]; i++)
 if(ViewId[RefLayerId[LayerIdInVps[nuh_layer_id]][i]] <
 ViewId[nuh_layer_id]) {
 for(j = NumInterLayerLeft; j > 0 && ViewId[RefLayerId[
 LayerIdInVps[nuh_layer_id]][i]] > >
 ViewId[the value of nuh_layer_id of
 RefPicSetLayerLeft[j - 1]]; j--)
 RefPicSetInterLayerLeft[j] =
 RefPicSetInterLayerLeft[j - 1]
 RefPicSetInterLayerLeft[j] = the picture with picture order
 count equal to PicOrderCnt and nuh_layer_id equal to
 RefLayerId[LayerIdInVps[nuh_layer_id]][i]
 NumInterLayerLeft++
 RefPicSetInterLeft[j] is marked as "used for long-term
 reference"
 } else {
 for(j = NumInterLayerRight; j > 0 && ViewId[RefLayerId[
 LayerIdInVps[nuh_layer_id]][i]] < <
 ViewId[the value of nuh_layer_id of

 RefPicSetLayerAfter[j - 1]]; j--)
 RefPicSetInterLayerRight [j] =
 RefPicSetInterLayerRight[j - 1]
 RefPicSetInterLayerRight[j] = the picture with picture order
 count equal to PicOrderCnt and nuh_layer_id equal to
 RefLayerId[LayerIdInVps[nuh_layer_id]][i]
 NumInterLayerRight++
 RefPicSetInterRight [j] is marked as "used for long-term
 reference"
 }
}

```

La salida del procedimiento de marcaje para finalizar la descodificación de una imagen codificada con nuh\_layer\_id mayor que 0 es un marcaje actualizado potencialmente como "utilizado para referencia cercana" para algunas imágenes descodificadas. Se aplica lo siguiente:

5

```

for(i = 0; i < NumInterLayerBefore; i++)
 RefPicSetInterLayerLeft[i] is marked as "used for short-term
 reference"
 for(i = 0; i < NumInterLayerAfter; i++)
 RefPicSetInterLayerRight[i] is marked as "used for short-term
 reference"

```

El siguiente proceso de descodificación para la construcción de las listas de imágenes de referencia se invoca al principio del proceso de descodificación para cada fragmento P o B.

10

Cuando se descodifica un fragmento P, existe una única lista de imágenes de referencia RefPicList0. Cuando se descodifica un fragmento B, existe una segunda lista independiente de imágenes de referencia RefPicList1 en adición a RefPicList0.

Al comienzo del proceso de descodificación para cada fragmento, la lista de imágenes de referencia RefPicList0, y RefPicList1 para los fragmentos B, se derivan como sigue.

15

La variable NumRpsCurrTempList0 se fija igual a  $\text{Max}(\text{num\_ref\_idx\_l0\_active\_minus1} + 1, \text{NumPocTotalCurr})$  y la lista RefPicListTemp0 se construye como sigue:

```

rIdx = 0
while(rIdx < NumRpsCurrTempList0) {
 for(i = 0; i < NumPocStCurrBefore && rIdx < IdumRpsCurrTempList0;
 rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetStCurrBefore[i]
 if (!interViewRefPictureListRightFirst)
 for(i = 0; i < NumInterLayerLeft; rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetInterLayerLeft[i]
 else
 for(i = 0; i < NumInterLayerRight; rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetInterLayerRight[i]
 for(i = 0; i < IdumPocStCurrAfter && rIdx < IdumRpsCurrTempList0;
 rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetStCurrAfter[i]
 for(i = 0; i < IdumPocLtCurr && rIdx < NumRpsCurrTempList0;
 rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetLtCurr[i]
 if (!interViewRefPictureListRightFirst)
 for(i = 0; i < NumInterLayerRight; rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetInterLayerRight[i]
 else
 for(i = 0; i < NumInterLayerLeft; rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetInterLayerLeft[i]
}

```

La lista RefPicList0 se construye como sigue:

```

for(rIdx = 0; rIdx <= num_ref_idx_l0_active_minus1; rIdx++)
 RefPicList0[rIdx] = ref_pic_list_modification_flag_l0 ?
 RefPicListTemp0[list_entry_l0[rIdx]] : RefPicListTemp0[rIdx]

```

- 5 Cuando el fragmento es un fragmento B, la variable NumRpsCurrTempList1 se fija igual a Max(num\_ref\_idx\_l1\_active\_minus1 + 1, NumPocTotalCurr) y la lista RefPicListTemp1 se construye como sigue:

```

rIdx = 0
while(rIdx < NumRpsCurrTempList1) {
 for(i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1;
 rIdx++, i++)
 RefPicListTemp1[rIdx] = RefPicSetStCurrAfter[i]
 if (!interViewRefPictureListRightFirst)
 for(i = 0; i < NumInterLayerRight; rIdx++, i++)
 RefPicListTemp1[rIdx] = RefPicSetInterLayerRight[i]
 else
 for(i = 0; i < NumInterLayerLeft; rIdx++, i++)
 RefPicListTemp1[rIdx] = RefPicSetInterLayerLeft[i]
 for(i = 0; i < IdumPocStCurrBefore && rIdx < NumRpsCurrTempList1;
 rIdx++, i++)
 RefPicListTemp1[rIdx] = RefPicSetStCurrBefore[i]
 for(i = 0; i < IdumPocLtCurr && rIdx < NumRpsCurrTempList1;
 rIdx++, i++)
 RefPicListTemp1[rIdx] = RefPicSetLtCurr[i]
 if (!interViewRefPictureListRightFirst)
 for(i = 0; i < NumInterLayerLeft; rIdx++, i++)
 RefPicListTemp1[rIdx] = RefPicSetInterLayerLeft[i]
 else
 for(i = 0; i < NumInterLayerRight; rIdx++, i++)
 RefPicListTemp1[rIdx] = RefPicSetInterLayerRight[i]
}

```

Cuando el fragmento es un fragmento B, la lista RefPicList1 se construye como sigue:

```
for(rIdx = 0; rIdx <= num_ref_idx_l1_active_minus1; rIdx++)
 RefPicList1[rIdx] = ref_pic_list_modification_flag_l1 ?
 RefPicListTemp1[list_entry_l1[rIdx]] : RefPicListTemp0[rIdx]
```

Alternativamente, se pueden colocar imágenes de la lista RefPicSetLtCurr en la última posición cuando se construyen las listas RefPicListTemp0 y RefPicListTemp1.

## 5 Posicionamiento de los conjuntos de imágenes de referencia en la construcción de la lista de imágenes de referencia.

Esta realización se refiere a aquellos casos en los que la capa actual y las capas de referencia difieren en sus respectivos valores de ViewId. La solución propuesta consta de dos etapas. - Etapa 1: Derivación de dos listas RefPicSetInterLayerLeft y RefPicSetInterLayerRight o, alternativamente, RefPicSetInterLayerBefore y RefPicSetInterLayerAfter.

Cada imagen de referencia entre tomas dada en la lista RefLayerId[] se inserta en RefPicSetInterLayerLeft o en RefPicSetInterLayerRight, dependiendo de su valor de ViewId, ViewId\_ref, y del valor de ViewId de la imagen actual, ViewId\_curr. Si ViewId\_ref < ViewId\_curr, se inserta en RefPicSetInterLayerLeft. Si no, se inserta en RefPicSetInterLayerRight. Las imágenes en RefPicSetInterLayerLeft se clasifican por el valor decreciente de su ViewId, es decir, el orden de las imágenes de referencia en RefPicSetInterLayerLeft es tal que la diferencia absoluta abs(ViewId\_ref - ViewId\_curr) aumenta. Las imágenes en RefPicSetInterLayerRight se clasifican por el valor creciente de su ViewId, es decir, el orden de las imágenes de referencia en RefPicSetInterLayerRight es tal que la diferencia absoluta abs(ViewId\_ref - ViewId\_curr) aumenta. - Etapa 2: Inserción de las imágenes de referencia entre tomas en las listas de imágenes de referencia RefPicListTemp0 y RefPicListTemp1.

RefPicListTemp0 y RefPicListTemp1 son las listas de imágenes de referencia antes de aplicar los comandos de modificación de la lista de imágenes de referencia. Para RefPicListTemp0, las imágenes de RefPicSetInterLayerLeft se insertan de acuerdo con su orden en RefPicSetInterLayerLeft. A continuación, las imágenes de RefPicSetInterLayerRight se insertan de acuerdo con su orden en RefPicSetInterLayerRight. Para RefPicListTemp1, las imágenes de RefPicSetInterLayerRight se insertan de acuerdo con su orden en RefPicSetInterLayerRight. A continuación, las imágenes de RefPicSetInterLayerLeft se insertan de acuerdo con su orden en RefPicSetInterLayerLeft.

Se propone la siguiente sintaxis de la extensión SPS:

| sps_extension( ) {                       | Descriptor |
|------------------------------------------|------------|
| <b>inter_layers_pos_flag</b>             | u(1)       |
| if (inter_layers_pos_flag) {             |            |
| <b>ref_pic_set_inter_layer_left_pos</b>  | u (v)      |
| <b>ref_pic_set_inter_layer_right_pos</b> | u (v)      |
| }                                        |            |
| }                                        |            |

Alternativamente, la información de la posición se envía al encabezamiento del fragmento, utilizando la siguiente sintaxis:

| slice_segment_header( ) {                | Descriptor |
|------------------------------------------|------------|
| <b>inter_layers_pos_flag</b>             | u(1)       |
| if (inter_layers_pos_flag) {             |            |
| <b>ref_pic_set_inter_layer_left_pos</b>  | u (v)      |
| <b>ref_pic_set_inter_layer_right_pos</b> | u (v)      |
| }                                        |            |
| }                                        |            |

El indicador `ref_pic_set_inter_layer_left_pos` especifica la posición para insertar el conjunto izquierdo de imágenes de referencia entre capas en `RefPicListTemp0` y `RefPicListTemp1`. El indicador `ref_pic_set_inter_layer_rigth_pos` especifica la posición para insertar el conjunto derecho de imágenes de referencia entre capas en `RefPicListTemp0` y `RefPicListTemp1`. Los indicadores especifican la posición en términos de los conjuntos de imágenes de referencia.

5 Alternativamente, los indicadores pueden especificar la posición en términos de las imágenes de referencia.

La salida del proceso de decodificación para los conjuntos de imágenes de referencia entre capas son dos listas actualizadas de imágenes entre capas, `RefPicSetInterLayerLeft` y `RefPicSetInterLayerRight`, y las variables `NumInterLayerLeft` y `NumInterLayerRight`. Este proceso es similar a los procesos correspondientes descritos para otras realizaciones.

10 La salida del siguiente procedimiento de marcaje para finalizar la decodificación de una imagen codificada con `nuh_layer_id` mayor que 0 es un marcaje potencialmente actualizado como "utilizado para referencia cercana" para algunas imágenes descodificadas. Se aplica lo siguiente:

```
for(i = 0; i < NumInterLayerLeft; i++)
 RefPicSetInterLayerLeft[i] is marked as "used for short-term
 reference"
for(i = 0; i < NumInterLayerRight; i++)
 RefPicSetInterLayerRight[i] is marked as "used for short-term
 reference"
```

15 El proceso de decodificación para la construcción de las listas de imágenes de referencia se invoca al principio del proceso de decodificación para cada fragmento P o B.

Cuando se descodifica un fragmento P, existe una única lista de imágenes de referencia `RefPicList0`. Cuando se descodifica un fragmento B, existe una segunda lista independiente de imágenes de referencia `RefPicList1` en adición a `RefPicList0`.

20 Al principio del proceso de decodificación para cada fragmento, la lista de imágenes de referencia `RefPicList0`, y `RefPicList1` para los fragmentos B, se derivan como sigue.

La variable `NumRpsCurrTempList0` se fija igual a  $\text{Max}(\text{num\_ref\_idx\_l0\_active\_minus1} + 1, \text{NumPocTotalCurr})$  y la lista `RefPicListTemp0` se construye como sigue:

```
insertInterLayerLeftSet (){
 for(i = 0; i < NumInterLayerLeft; rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetInterLayerLeft[i]
 }
 insertInterLayerRightSet (){
```

```

for(i = 0; i < NumInterLayerRight; rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetInterLayerRight[i]
}
rIdx = 0
refPicSetPos = 1
while(rIdx < NumRpsCurrTempList0) {
if (refPicSetInterLayerLeftPos == refPicSetPos) {
 insertInterLayerLeftSet ();
refPicSetPos = refPicSetPos +1 ;
}
if (refPicSetInterLayerRightPos == refPicSetPos) {
 insertInterLayerRightSet ();
 refPicSetPos = refPicSetPos +1 ;
}
for(i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0;
 rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetStCurrBefore[i]
refPicSetPos = refPicSetPos +1 ;
if (refPicSetInterLayerLeftPos == refPicSetPos) {
 insertInterLayerLeftSet ();
 refPicSetPos = refPicSetPos +1 ;
}
if (refPicSetInterLayerRightPos == refPicSetPos) {
 insertInterLayerRightSet ();
 refPicSetPos = refPicSetPos +1 ;
}

for(i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList0;
 rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetStCurrAfter[i]
refPicSetPos = refPicSetPos +1 ;

if (refPicSetInterLayerLeftPos == refPicSetPos) {
 insertInterLayerLeftSet ();
 refPicSetPos = refPicSetPos +1 ;
}
if (refPicSetInterLayerRightPos == refPicSetPos) {
 insertInterLayerRightSet ();
 refPicSetPos = refPicSetPos +1 ;
}
for(i = 0; i < NumPocLtCurr && rIdx < IdumRpsCurrTempList0;
 rIdx++, i++)
 RefPicListTemp0[rIdx] = RefPicSetLtCurr[i]
refPicSetPos = refPicSetPos +1 ;
if (refPicSetInterLayerLeftPos == refPicSetPos) {
 insertInterLayerLeftSet ();
 refPicSetPos = refPicSetPos +1 ;
}
if (refPicSetInterLayerRightPos == refPicSetPos) {
 insertInterLayerRightSet ();
 refPicSetPos = refPicSetPos +1 ;
}
}
}

```

La lista RefPicList0 se construye como sigue:



```

for(rIdx = 0; rIdx <= num_ref_idx_l0_active_minus1; rIdx++)
 RefPicList0[rIdx] = ref_pic_list_modification_flag_l0 ?
 RefPicListTemp0[list_entry_l0[rIdx]] : RefPicListTemp0[rIdx]

```

Cuando el fragmento es un fragmento B, la variable NumRpsCurrTempList1 se fija igual a  $\text{Max}(\text{num\_ref\_idx\_l1\_active\_minus1} + 1, \text{NumPocTotalCurr})$  y la lista RefPicListTemp1 se construye como sigue:

```

rIdx = 0

while(rIdx < IdumRpsCurrTempList1) {
 if (refPicSetInterLayerLeftPos == refPicSetPos) {
 insertInterLayerLeftSet ();
 refPicSetPos = refPicSetPos + 1 ;
 }
 if (refPicSetInterLayerRightPos == refPicSetPos) {
 insertInterLayerRightSet ();
 refPicSetPos = refPicSetPos + 1 ;
 }
 for(i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1;
 rIdx++, i++)
 RefPicListTemp1[rIdx] = RefPicSetStCurrAfter[i]
 refPicSetPos = refPicSetPos + 1 ;
 if (refPicSetInterLayerLeftPos == refPicSetPos) {
 insertInterLayerLeftSet ();
 refPicSetPos = refPicSetPos + 1 ;
 }
 if (refPicSetInterLayerRightPos == refPicSetPos) {
 insertInterLayerRightSet ();
 refPicSetPos = refPicSetPos + 1 ;
 }

 for(i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList1;
 rIdx++, i++)
 RefPicListTemp1[rIdx] = RefPicSetStCurrBefore[i]
 refPicSetPos = refPicSetPos + 1 ;

 if (refPicSetInterLayerRightPos == refPicSetPos) {
 insertInterLayerRightSet ();
 refPicSetPos = refPicSetPos + 1 ;
 }
 for(i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList1;
 rIdx++, i++)
 RefPicListTemp1[rIdx] = RefPicSetLtCurr[i]
 refPicSetPos = refPicSetPos + 1 ;
 if (refPicSetInterLayerLeftPos == refPicSetPos) {
 insertInterLayerLeftSet ();
 refPicSetPos = refPicSetPos + 1 ;
 }
 if (refPicSetInterLayerRightPos == refPicSetPos) {
 insertInterLayerRightSet ();
 refPicSetPos = refPicSetPos + 1 ;
 }
}

```

Cuando el fragmento es un fragmento B, la lista RefPicList1 se construye como sigue:

```

for(rIdx = 0; rIdx <= num_ref_idx_l1_active_minus1; rIdx++)
 RefPicList1[rIdx] = ref_pic_list_modification_flag_l1 ?
 RefPicListTemp1[list_entry_l1[rIdx]] : RefPicListTemp1[rIdx]

```

Alternativamente, las imágenes de la lista RefPicSetLtCurr se pueden colocar en la última posición cuando se construyen las listas RefPicListTemp0 y RefPicListTemp1.

La invención se clarifica adicionalmente por medio de los siguientes ejemplos no limitativos.

- 5     - Ejemplo 1: El identificador de la escalabilidad DependencyId puede corresponder a una resolución espacial, en la que un mayor valor de DependencyId corresponde a una mayor resolución espacial. Por ejemplo, si la capa actual tiene DependencyId=3 y las imágenes de referencia tienen DependencyId=0, DependencyId=1 y DependencyId=2, entonces la capa con DependencyId=2 es probablemente la más similar a la capa actual y se puede predecir que probablemente funcionará mejor. Además, la capa con DependencyId=1 es probablemente la segunda más similar a la capa actual y la capa con DependencyId=0 es probablemente la menos parecida a la capa actual.
- 10    Por consiguiente, las imágenes de referencia entre capas se insertan según el orden DependencyId=2, DependencyId=1, DependencyId=0, es decir, en el orden de la diferencia respectiva de su valor de DependencyId y el valor de DependencyId de la imagen actual.
- 15     - Ejemplo 2: El identificador de la escalabilidad QualityId puede corresponder a un nivel de calidad (por ejemplo, la magnitud de la etapa cuantificadora) en el que un mayor valor de QualityId corresponde a una mayor fidelidad de la señal. Por ejemplo, si la capa actual tiene QualityId=3 y las imágenes de referencia tienen QualityId=0, QualityId=1 y QualityId=2, entonces la capa con QualityId=2 es probablemente la más similar a la capa actual y se pueda predecir que probablemente funcionará mejor. Además, la capa con QualityId=1 es probablemente la segunda más similar a la capa actual y la capa con QualityId=0 es probablemente la menos parecida a la capa actual.
- 20    Por consiguiente, las imágenes de referencia entre capas se insertan según el orden QualityId=2, QualityId=1, QualityId=0, es decir, en el orden de la diferencia respectiva de su valor de QualityId y el valor de QualityId de la imagen actual.
- 25     - Ejemplo 3: El identificador de la escalabilidad DependencyId puede corresponder a una resolución espacial y el identificador de la escalabilidad QualityId puede corresponder a un nivel de calidad (ejemplo, magnitud de la etapa cuantificadora), en el que un mayor valor de DependencyId corresponde a una mayor resolución espacial y un valor mayor de QualityId corresponde a una mayor fidelidad de la señal. Por ejemplo, si la capa actual tiene DependencyId=1 y QualityId=0 y las capas de referencia tienen respectivamente DependencyId=0/QualityId=0, DependencyId=0/QualityId=1 y DependencyId=0/QualityId=2, entonces la capa con DependencyId=0/QualityId=2 es probablemente la más parecida a la capa actual y las imágenes de referencia entre capas se insertan según el orden DependencyId=0/QualityId=2, DependencyId=0/QualityId=1, DependencyId=0/QualityId=0, es decir, empezando con el mayor valor de QualityId y a continuación en orden decreciente de su valor.
- 30     - Ejemplo 4: El identificador de la escalabilidad ViewId puede corresponder a un identificador de toma, en el que el identificador de toma indica una posición espacial de la cámara. Las tomas con similares posiciones de cámara son probablemente parecidas. Por ello, las imágenes de referencia entre capas se pueden insertar de acuerdo con la diferencia de su valor de ViewId y el valor de ViewId de la imagen actual. En un caso particular, cuando el fragmento actual es un fragmento B que tiene dos listas de imágenes de referencia, list0 puede contener imágenes de referencia que tienen ViewId menor que el ViewId de la imagen actual, y list1 puede contener imágenes de referencia que tienen ViewId mayor que el ViewId de la imagen actual (o viceversa) y las listas se pueden ordenar de acuerdo con su distancia, es decir, diferencia de los identificadores de capa, de la toma actual. Por consiguiente, si la imagen actual tiene ViewId=2, y las imágenes de referencia tienen ViewId=0, ViewId=1, ViewId=3, ViewId=4, entonces list0 puede contener dos imágenes de referencia según el orden ViewId=1, ViewId=0 y list1 puede contener dos imágenes de referencia según el orden ViewId=3, ViewId=4. Si se pueden insertar más imágenes de referencia, list0 puede contener ViewId=1, ViewId=0, ViewId=3, ViewId=4, y list1 puede contener ViewId=3, ViewId=4 ViewId=1, ViewId=0.
- 35     De acuerdo con una realización de la invención, en la que la selección y ordenamiento se realizan basándose en el valor de un identificador de toma ViewId, se define que el valor de ViewId sea igual a 0 para la capa con layer\_Id igual a 0, y el valor de ViewId de una capa con layer\_Id mayor que 0 se señala según un formato binario, en el cual un bit representa un signo y uno o más bits representan una magnitud.
- 40     De acuerdo con una realización de invención, el valor de ViewId para la capa con layer\_Id igual a 0 se señala explícitamente.
- 45     - Ejemplo 5: Los parámetros relativos a unas posiciones de la cámara se utilizan con objeto de clasificar las imágenes de referencia en la lista. Por ejemplo, un parámetro de escala como se define en la especificación provisional 3D-HEVC, se puede utilizar para este propósito. Alternativamente, se pueden utilizar otros parámetros derivados de las posiciones relativas de las cámaras, por ejemplo, un parámetro de la línea de base, para clasificar las imágenes de referencia en la lista. Se crea entonces la lista en orden creciente o decreciente de la distancia de las posiciones de la cámara, que corresponde a una diferencia de la imagen de referencia de la imagen actual en términos de identificadores de capas, es decir, una diferencia entre identificadores de capas.
- 50     - Ejemplo 5: Los parámetros relativos a unas posiciones de la cámara se utilizan con objeto de clasificar las imágenes de referencia en la lista. Por ejemplo, un parámetro de escala como se define en la especificación provisional 3D-HEVC, se puede utilizar para este propósito. Alternativamente, se pueden utilizar otros parámetros derivados de las posiciones relativas de las cámaras, por ejemplo, un parámetro de la línea de base, para clasificar las imágenes de referencia en la lista. Se crea entonces la lista en orden creciente o decreciente de la distancia de las posiciones de la cámara, que corresponde a una diferencia de la imagen de referencia de la imagen actual en términos de identificadores de capas, es decir, una diferencia entre identificadores de capas.
- 55     - Ejemplo 5: Los parámetros relativos a unas posiciones de la cámara se utilizan con objeto de clasificar las imágenes de referencia en la lista. Por ejemplo, un parámetro de escala como se define en la especificación provisional 3D-HEVC, se puede utilizar para este propósito. Alternativamente, se pueden utilizar otros parámetros derivados de las posiciones relativas de las cámaras, por ejemplo, un parámetro de la línea de base, para clasificar las imágenes de referencia en la lista. Se crea entonces la lista en orden creciente o decreciente de la distancia de las posiciones de la cámara, que corresponde a una diferencia de la imagen de referencia de la imagen actual en términos de identificadores de capas, es decir, una diferencia entre identificadores de capas.

- Ejemplo 6: Los parámetros derivados de las posiciones relativas de las cámaras se pueden también utilizar con objeto de crear dos listas de referencia, por ejemplo, llamadas "lista de referencia entre tomas antes" y "lista de referencia entre tomas después", o, alternativamente, llamadas "lista de imágenes de referencia entre tomas izquierda" y "lista de imágenes de referencia entre tomas derecha". Estas listas de imágenes de referencia se ordenan según el orden creciente de distancia entre las posiciones de la cámara correspondientes a las imágenes de referencia y la posición de la cámara correspondiente a la imagen actual. En este caso, sólo una lista puede incluir las imágenes que corresponden a las posiciones de la cámara que están localizadas a un lado de la posición de la cámara correspondiente a la imagen. En correspondencia, la segunda lista puede incluir las imágenes localizadas al otro lado de la posición de la cámara correspondiente a la imagen actual.

- Ejemplo 7: Los parámetros derivados de las posiciones relativas de las cámaras se pueden también utilizar con objeto de crear dos listas de referencia, por ejemplo, llamadas "lista de referencia entre tomas antes" y "lista de referencia entre tomas después", o, alternativamente, llamadas "lista de imágenes de referencia entre tomas izquierda" y "lista de imágenes de referencia entre tomas derecha". Estas dos listas se pueden utilizar separadamente cuando se crea la lista final de imágenes de referencia. Por ejemplo, las posiciones para insertar la lista de imágenes de referencia derecha y la lista de imágenes de referencia izquierda en una lista final de imágenes de referencia se puede especificar por defecto o puede ser señalizada en el flujo de datos.

La lista final de imágenes de referencia puede incluir además, por ejemplo, una o más listas de imágenes de referencia cercanas y una o más listas de imágenes de referencia lejanas, y las posiciones para insertar la lista de imágenes de referencia derecha y la lista de imágenes de referencia izquierda en la lista final de imágenes de referencia puede ser relativa a las posiciones en las que se insertan una o más listas de las imágenes de referencia cercanas y una o más listas de las imágenes de referencia lejanas.

- Ejemplo 8: Se puede señalar un indicador con objeto de "permutar" la "lista de imágenes de referencia entre tomas izquierda" y la "lista de imágenes de referencia entre tomas derecha". Aquí, permutar quiere decir que las imágenes de referencia que se incluyeron en la "lista de imágenes de referencia entre tomas izquierda" se retirarán de la "lista de imágenes de referencia entre tomas izquierda" y se incluirán en la "lista de imágenes de referencia entre tomas derecha", si el valor del indicador se ha fijado en "verdadero" (o 1). De la misma manera, las imágenes de la "lista de imágenes de referencia entre tomas derecha" se retirarán de la "lista de imágenes de referencia entre tomas derecha" y se incluirán en la "lista de imágenes de referencia entre tomas izquierda". El indicador se puede señalar en el SPS, en el PPS, en un encabezamiento del fragmento (o encabezamiento de la porción del fragmento), o en otro parámetro fijado que sea parte del flujo de datos.

Alternativamente, se puede utilizar un criterio implícito con objeto de determinar si las imágenes de la "lista de imágenes de referencia entre tomas derecha" se deben permutar con las imágenes en la "lista de referencia entre tomas izquierda".

- Ejemplo 9: En los ejemplos 4, 6 y 7, las posiciones exactas para los conjuntos resultantes de imágenes de referencia en la lista preliminar resultante de imágenes de referencia se puede señalar explícitamente con objeto de mejorar la eficiencia de la codificación. El presente ejemplo se puede considerar como una extensión del Ejemplo 8 con la diferencia de que la posición exacta para el primer conjunto de imágenes de referencia y para el segundo conjunto de imágenes de referencia se envía al descodificador. La posición se puede señalar en forma de la posición de la imagen de referencia en la lista así como de la posición del conjunto de imágenes de referencia en términos de los conjuntos de imágenes de referencia.

Con referencia a la figura 2, se describe una realización 200 del método de construir al menos una lista de imágenes de referencia para la predicción entre capas de una imagen actual. El método 200 comprende insertar imágenes de referencia 210 en un primer conjunto de imágenes de referencia o en un segundo conjunto de imágenes de referencia, basándose en los respectivos valores de un identificador de la escalabilidad asociado con las imágenes de referencia y en un valor del identificador de la escalabilidad asociado con la imagen actual, e insertar 220 el primer conjunto de imágenes de referencia y el segundo conjunto de imágenes de referencia en al menos una lista de imágenes de referencia. El método 200 puede además comprender ordenar 230 las imágenes de referencia comprendidas en el primer conjunto de imágenes de referencia y en el segundo conjunto de imágenes de referencia de acuerdo con sus respectivos valores del identificador de la escalabilidad. Opcionalmente, el método 200 puede comprender además las etapas descritas anteriormente en este documento.

En la figura 3, se ilustra una realización 300 de un codificador. El codificador 300 comprende al menos una sección de entrada 301, una sección de salida 302, un procesador 303 y una memoria 304.

El codificador 300 está dispuesto para recibir un vídeo, es decir, una secuencia de imágenes, a través de la sección de entrada 301 y un procesador 303 está configurado para realizar los procedimientos descritos anteriormente en este documento. Adicionalmente, el procesador 303 puede estar configurado para realizar otros procedimientos relativos a la codificación de vídeo que son conocidos en la técnica. La sección de salida 302 puede estar dispuesta para proporcionar el flujo de datos para el tratamiento o transporte adicional sobre una red de comunicaciones.

La funcionalidad del procesador se puede realizar por medio de un programa de ordenador 305, es decir, software, almacenado en la memoria 304. El programa de ordenador 305 consta de instrucciones ejecutables por ordenador que están adaptadas, cuando se ejecutan en el procesador 303, para ejecutar los procedimientos descritos anteriormente en este documento.

- 5 Se puede proporcionar una realización del codificador 300 como parte 601 de un terminal móvil 600 (ilustrado en la figura 6), una tableta, un ordenador o similares.

En la figura 4, se ilustra una realización 400 de un descodificador. El descodificador 400 comprende al menos una sección entrada 401, una sección de salida 402, un procesador 403 y una memoria 404.

- 10 El descodificador 400 está dispuesto para recibir un flujo de datos a través de la sección de entrada 401 y el procesador 403 está configurado para realizar los procedimientos descritos anteriormente en este documento. Adicionalmente, el procesador 403 puede estar configurado para realizar otros procedimientos relativos a la codificación de vídeo como es conocido en la técnica. La sección de salida 402 puede estar dispuesta para proporcionar el vídeo para un tratamiento adicional, por ejemplo, presentación en pantalla. La funcionalidad del procesador 403 se puede realizar por medio de un programa de ordenador 405, es decir, software, almacenado en la memoria 404. El programa de ordenador 405 comprende instrucciones ejecutables por ordenador que están adaptadas, cuando se ejecuta en el procesador 403 para ejecutar los procedimientos descritos anteriormente en ese documento.

Se puede proporcionar una realización de un descodificador 400 como parte 601 de un terminal móvil 600 (ilustrado en la figura 6), una tableta, un ordenador o similares.

- 20 Una realización del descodificador 400 puede recibir el flujo de datos de una realización del codificador 300 a través de la red de comunicaciones, tal como internet, una red celular, una red de radio o similares o una combinación de ello.

- 25 En la figura 5, se ilustra una realización alternativa 500 de un codificador de vídeo (descodificador). El codificador de vídeo (descodificador) 500 comprende secciones de entrada y salida, como las descritas con referencia a las figuras 3 y 4 (no mostradas en la figura 5), un módulo codificador (descodificador) 501, un módulo de conjuntos 502, un módulo de listas 503 y, opcionalmente, un módulo de ordenamiento 504. El módulo de conjuntos 502 y el módulo de listas 503 están configurado para operar de acuerdo con lo que se ha descrito anteriormente en este documento. En particular, el módulo de conjuntos 502 está configurado para insertar imágenes de referencia en un primer conjunto de imágenes de referencia o en un segundo conjunto de imágenes de referencia basándose en los valores respectivos del identificador de la escalabilidad asociado con las imágenes de referencia y en un valor del identificador de la escalabilidad asociado con la imagen actual. Adicionalmente, el módulo de listas 503 está configurado para insertar el primer conjunto de imágenes de referencia y el segundo conjunto de imágenes de referencia en al menos una lista de imágenes de referencia.

- 35 Si existe, el módulo de ordenamiento 503 está configurado para operar de acuerdo con lo que se ha descrito anteriormente en este documento, en particular para ordenar las imágenes de referencia comprendidas en el primer conjunto de imágenes de referencia y en el segundo conjunto de imágenes de referencia de acuerdo con sus respectivos valores del identificador de la escalabilidad.

Se puede proporcionar una realización del codificador (descodificador) 500 como parte 601 de un terminal móvil 600 (ilustrado en la figura 6), una tableta, un ordenador o similares.

- 40 Los expertos en la técnica comprobarán que la invención no pretende limitarse a las realizaciones descritas anteriormente. Por el contrario, son posibles muchas modificaciones y variaciones dentro del alcance de las reivindicaciones adjuntas.

## REIVINDICACIONES

1. Un método (200) para construir una primera lista de imágenes de referencia para la predicción entre capas de una imagen actual de una secuencia de vídeo multi capa, que comprende:
  - 5 insertar (210) imágenes de referencia que tienen valores respectivos de un indicador de la toma, que indican una posición relativa espacial de la captura de la cámara, que son menores que el valor del identificador de la toma asociado con la imagen actual dentro de un primer conjunto (101) de imágenes de referencia,
  - insertar (210) imágenes de referencia que tienen valores respectivos del identificador de la toma que son mayores que el valor del identificador de la toma asociado con la imagen actual dentro de un segundo conjunto (102) de imágenes de referencia,
  - 10 insertar (220) el primer conjunto (101) de imágenes de referencia en la primera lista de imágenes de referencia al frente del segundo conjunto (102) de imágenes de referencia, e
  - insertar (220) el segundo conjunto (102) de imágenes de referencia en la segunda lista de imágenes de referencia al frente del primer conjunto (101) de imágenes de referencia.
2. El método de acuerdo con la reivindicación 1, que comprende además:
  - 15 ordenar (230) las imágenes de referencia comprendidas en el primer conjunto de imágenes de referencia y en el segundo conjunto de imágenes de referencia de acuerdo con sus respectivos valores del identificador de la toma.
3. El método de acuerdo con la reivindicación 2, en el que las imágenes de referencia se ordenan (230) según el valor decreciente del identificador de la toma.
4. El método de acuerdo con la reivindicación 2, en el que las imágenes de referencia se ordenan (230) según el  
20 valor creciente del identificador de la toma.
5. El método de acuerdo con la reivindicación 2, en el que las imágenes de referencia se ordenan (230) según la diferencia absoluta decreciente de sus valores respectivos del identificador de la toma y del valor del identificador de la toma asociado con la imagen actual.
6. El método de acuerdo con la reivindicación 2, en el que las imágenes de referencia se ordenan (230) según la  
25 diferencia absoluta creciente de sus valores respectivos del identificador de la toma y del valor del identificador de la toma asociado con la imagen actual.
7. El método de acuerdo con cualquiera de las reivindicaciones 1 a 6, en el que:
  - el primer conjunto de imágenes de referencia se inserta (220) en la primera lista de imágenes de referencia después de un primer conjunto de imágenes de referencia cercanas,
  - 30 el primer conjunto de imágenes de referencia se inserta (220) al final de la segunda lista de imágenes de referencia,
  - el segundo conjunto de imágenes de referencia se inserta (220) en la segunda lista de imágenes de referencia después del segundo conjunto de imágenes de referencia cercanas y antes del primer conjunto de imágenes de referencia cercanas, y
  - el segundo conjunto de imagen de referencia se inserta (220) al final de la primera lista de imágenes de referencia.
8. Un método de codificar un video que comprende el método de acuerdo con cualquiera de las reivindicaciones 1 a 7.
9. Un método de decodificar un video que comprende el método de acuerdo con cualquiera de las reivindicaciones 1 a 7.
10. Un programa de ordenador (305; 405) que consta de instrucciones ejecutables por ordenador para hacer que un  
40 dispositivo (300; 400) realice el método de acuerdo con cualquiera de las reivindicaciones 1 a 9, cuando las instrucciones ejecutables por ordenador se ejecutan en una unidad de procesamiento (303; 403) comprendida en el dispositivo.
11. Un producto de programa de ordenador que consta de un medio de almacenamiento interpretable por ordenador (304; 404), disponiendo el medio de almacenamiento interpretable por ordenador de un programa de ordenador (305; 405) incorporado en el mismo de acuerdo con la reivindicación 10.
- 45 12. Un dispositivo (300; 400; 500; 601) para construir una primera lista de imágenes de referencia y una segunda lista de imágenes de referencia para la predicción entre capas de una imagen actual de una secuencia de vídeo multi capa, comprendiendo el dispositivo medios (301-305; 401-405; 501-504) configurados para:

- insertar imágenes de referencia que tienen valores respectivos de un identificador de la toma, que indican una posición de captura relativa espacial de la cámara, que es menor que el valor de un identificador de la toma asociado con la imagen actual en un primer conjunto (101) de imágenes de referencia,
- 5 insertar imágenes de referencia que tienen valores respectivos de un identificador de la toma mayores que el valor del identificador de la toma asociado con la imagen actual en un segundo conjunto (102) de imágenes de referencia,
- insertar el primer conjunto (101) de imágenes de referencia en la primera lista de imágenes de referencia al frente del segundo conjunto (102) de imágenes de referencia, e
- insertar el segundo conjunto (102) de imágenes de referencia en la segunda lista de imágenes de referencia al frente del primer conjunto (101) de imágenes de referencia.
- 10 13. El dispositivo de acuerdo con la reivindicación 12, estando los medios configurados además para ordenar las imágenes de referencia comprendidas en el primer conjunto de imágenes de referencia y en el segundo conjunto de imágenes de referencia de acuerdo con sus respectivos valores del identificador de la toma.
14. El dispositivo de acuerdo con la reivindicación 13, en el que están los medios configurados para ordenar las imágenes de referencia según el valor decreciente del identificador de la toma.
- 15 15. El dispositivo de acuerdo con la reivindicación 13, en el que están los medios configurados para ordenar las imágenes de referencia según el valor creciente del identificador de la toma.
16. El dispositivo de acuerdo con la reivindicación 13, en el que están los medios configurados para ordenar las imágenes de referencia según la diferencia absoluta decreciente de sus respectivos valores del identificador de la toma y el valor del identificador de la toma asociado con la imagen actual.
- 20 17. El dispositivo al acuerdo con la reivindicación 13, en el que están los medios están configurados para ordenar las imágenes de referencia según la diferencia absoluta creciente de sus respectivos valores del identificador de la escalabilidad y del valor del identificador de la escalabilidad asociado con la imagen actual.
18. El dispositivo de acuerdo con cualquiera de la reivindicaciones 12 a 17, en el que están los medios configurados para:
- 25 insertar el primer conjunto de imágenes de referencia en la primera lista de imágenes de referencia después de un primer conjunto de imágenes de referencia cercanas,
- insertar el primer conjunto de imágenes de referencia al final de la segunda lista de imágenes de referencia,
- insertar el segundo conjunto de imágenes de referencia en la segunda lista de imágenes de referencia después del segundo conjunto de imágenes de referencia cercanas y antes del primer conjunto de imágenes de referencia cercanas, e
- 30 insertar el segundo conjunto de imágenes de referencia al final de la primera lista de imágenes de referencia.
19. El dispositivo de acuerdo con cualquiera de las reivindicaciones 12 a 18, que es un codificador de vídeo (400).
20. El dispositivo donde acuerdo con cualquiera de las reivindicaciones 12 a 18, que es un decodificador de vídeo (500).
- 35 21. Un terminal móvil (600) que comprende el dispositivo (601) de acuerdo con cualquiera de las reivindicaciones 12 a 20.

Fig. 1

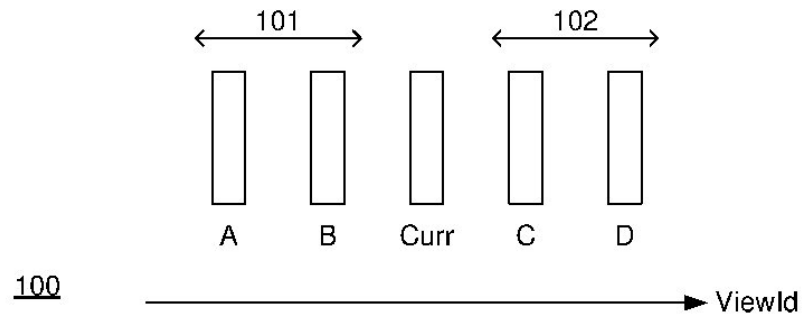


Fig. 2

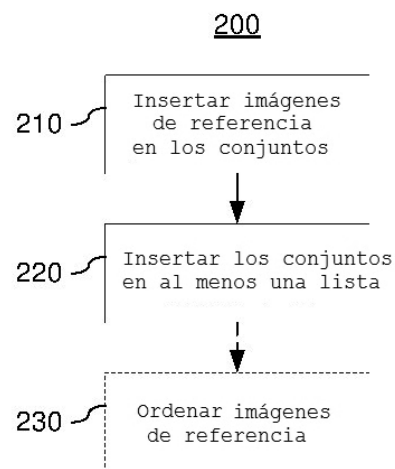


Fig. 3

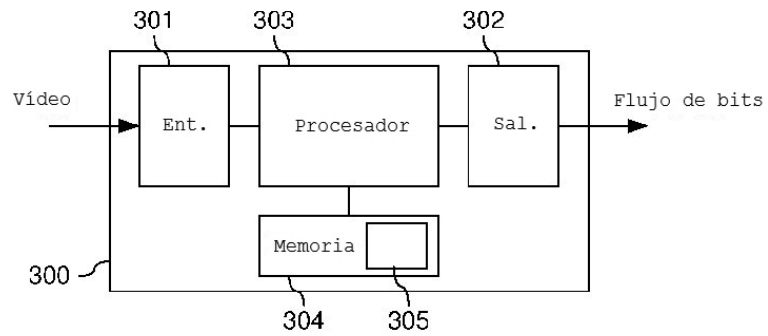


Fig. 4

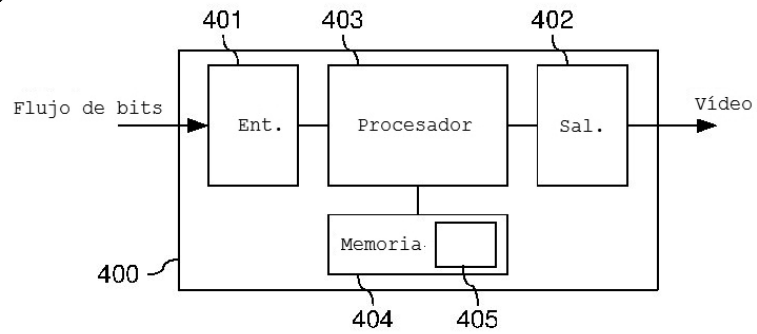


Fig. 5

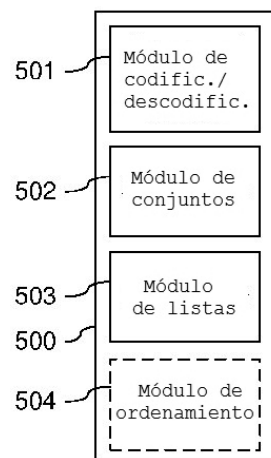


Fig. 6

