6 CAPA DE VÍDEO

6.1 Estructura de datos de vídeo codificados

Los datos de vídeo codificados consisten en un conjunto ordenado de trenes de bits de vídeo (*video bitstream*) denominados capas. Si hay sólo una capa, se dice que tienen una jerarquía no escalable. Si hay dos o más capas, los datos de vídeo codificado, tienen una jerarquía escalable.

La primera capa se denomina capa básica, y se puede decodificar siempre de forma independiente. El resto de capas, se denominan capas de mejora y sólo se puede decodificar juntas con todas las capas más bajas que comienzan con la capa básica. Explicaremos el resto de capas en el último apartado de este capítulo.

Como este proyecto está dedicado al análisis y decodificación de esta capa, prestaremos especial atención, definiendo la sintaxis de todas las funciones a implementar en un decodificador genérico y la semántica de los parámetros fundamentales.

6.1.1 Jerarquía de la capa básica

La primera capa forma una trama estructurada de forma jerárquica con el objetivo de aumentar la flexibilidad del sistema y la intercomunicación entre el codificador y el decodificador. Mediante esta estructura de trama pueden resultar compatibles codificadores que realicen procesos lógicos distintos.

La estructura de la trama empieza con una cabecera de la capa superior y continúa con las cabeceras e informaciones de las capas siguientes. Las capas utilizadas, en orden jerárquico descendiente, son las de secuencia, grupo de imágenes, imagen, slice, macrobloque y bloque.

Las relaciones de estas capas en el *bitstream* de vídeo se representan de forma esquemática en la figura 6.1.

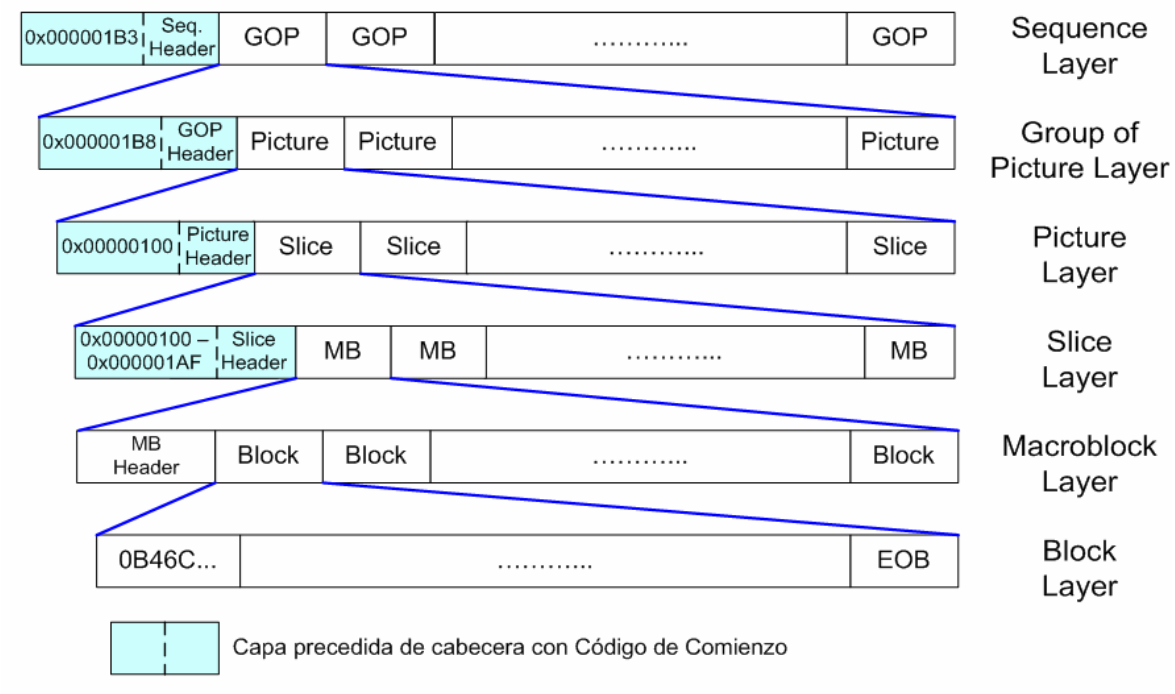

Figura 6.1Estructura jerárquica del tren de bits de vídeo

6.1.2 Códigos de comienzo

Cada capa se delimita mediante un código de comienzo. Estos códigos son esquemas de bits específicos que se repiten dentro de la propia información codificada y están formados por cuatro bytes. A estos códigos se les denomina Código de Comienzo (*Start code*).

Están compuestos por un prefijo de código (*prefix code*) de tres bytes, común para todos ellos, con valor en hexadecimal de 0x000001. El cuarto byte se denomina valor de código de comienzo y sirve para identificar el tipo de información que viene a continuación. En la tabla 6.1 se muestran los distintos valores de este byte.

	Valor del código de comienzo	
Nombre	**Hexadecimal**	**Decimal**
picture_start_code	00	0
slice_start_code	01 – AF	1 – 175
reserved	B0	176
reserved	B1	177
user_data_start_code	B2	178
sequence_header_code	B3	179
sequence_error_code	B4	180
extensión_start_code	B5	181
reserved	B6	182
sequence_end_code	B7	183
group_start_code	B8	184
system start code	B9 hasta FF	185 – 255
NOTA – system start code se definen en la capa de sistema		

Tabla 6.1 Valores de código de comienzo

Todos estos códigos de comienzo están alineados en bytes. Eso se logra insertando bits con el valor cero antes de l prefijo de modo que el primer bit del prefijo de código es el bit más significativo de un byte.

6.1.3 Reglas semánticas para estructuras sintácticas más altas

La capa de vídeo se ha de encargar de la descripción de la secuencia de bits con información de vídeo y de la forma de acceder a cada una de las capas (con sus correspondientes cabeceras) para poder extraer la información codificada en ellas. Además de las capas básicas pueden aparecer extensiones que codifican datos adicionales. La decodificación de una secuencia de vídeo terminará cuando se encuentra el delimitador de final de secuencia, a partir del cual ya no son válidos los parámetros de la anterior cabecera de secuencia.

Los elementos sintácticos de nivel más alto pueden combinarse juntos para producir un tren de bits legal, cumpliendo el siguiente código.

```
video_sequence() {
    next_start_code()
    sequence_header()
    if (nextbits() = = extensión_start_code) {
        sequence_extension()
        do {
            extension_and_user_data(0)
            do {
                if (nextbits() = = group_start_code()
                    group _of_pictures_header()
                    extension_and_user_data(1)
                }
                picture_heade()
                picture_coding_extension()
                extension_and_user_data(2)
                picture_data()
            } while ((nextbits() = = picture_start_code) ||
                    (nextbits() = = group_start_code))
            if (nextbits() != sequence_end_code) {
                sequence_header()
                sequence_extension()
            }
        } while (nextbits() != sequence_ende_code)
    } else {
        /* ISO/IEC 11172-2 */
    }
    sequence_end_code
}
```

Además, se deben cumplir ciertas reglas básicas que se enumeran a continuación. En la figura 6.2 se ilustra el diagrama de flujo de las cabeceras definido en MPEG-2.

- Si el primer sequence_header() de la secuencia no está seguido por sequence_extension(), el tren se conformará con ISO/CEI 11172-2 y no está documentado en MPEG-2.
- Si el primer sequence_header() está seguido por una sequence_extension(), todas las apariciones subsiguientes de secuence_header() estarán seguidos inmediatamente por una sequence_extension().
- La sequence_extension() sólo se producirá inmediatamente después de un sequence_header().
- Después de un sequence_header() habrá por lo menos una imagen codificada antes de otro sequence_header() o un sequence_end_code(). Esto implica que la sequence_extension() no precederá inmediatamente a un código de fin de secuencia.

- Si se produce una sequence_extension() en el tren de bits, cada picture_header() estará seguido inmediatamente por una picture_coding_extension().
- El sequence_end_code se situará al final del tren binario de modo que, tras la codificación y la reordenación de trama, no falte ninguna trama.
- la picture_coding_extension() sólo ocurrirá inmediatamente después de un picture_header().
- La primera trama codificada que sigue a un group_of_pictures_header() será una trama I codificada.

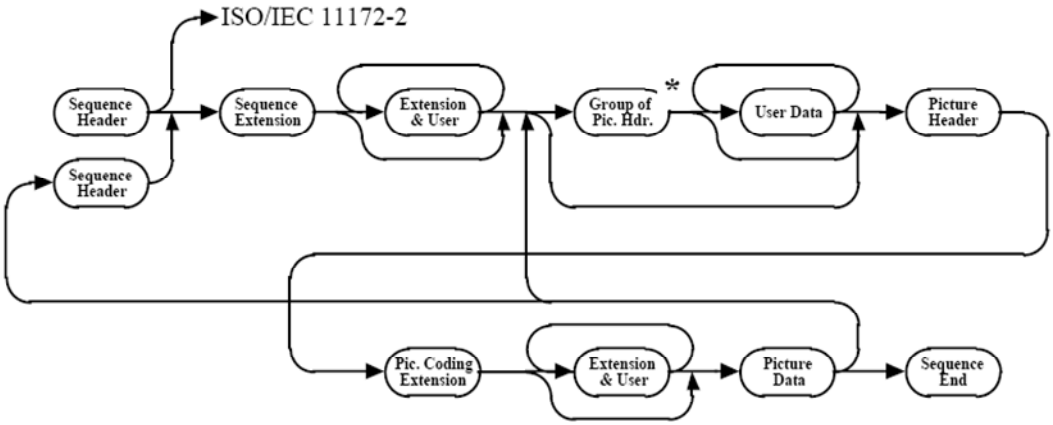

Figura 6.2 Organización del tren de bits de alto nivel

Se definen varias extensiones diferentes además de la extensión de secuencia y la extensión de codificación de imagen. El conjunto de extensiones autorizadas es diferente en cada punto de la sintaxis donde se permiten extensiones. La tabla 6.2 define un identificador de código de comienzo de extensión (extension_start_code_identifier) de 4 bits para cada extensión. Estas extensiones son una novedad del estándar MPEG-2 respecto al MPEG-1, y se incorporan para proporcionar flexibilidad ya que dependiendo del vídeo codificado se deberán tener unos parámetros u otros.

Extensión_start_code_identifier	*Nombre*
0000	reseved
0001	Sequence Extension ID
0010	Sequence Display Extension ID
0011	Quant Matriz Extension ID
0100	Copyright Extension ID
0101	Sequence Sclable Extension ID
0110	reserved
0111	Picture Display Extension ID
1000	Picture Coding Extension ID
1001	Picture Spatial Scalable Extension ID
1010	Picture Temporal Scalable Extension ID
1011	reserved
1100	reserved
1101	
…	
1111	reserved

Tabla 6.2 Códigos de identificador de código de comienzo de extensión

En cada punto donde se permiten extensiones en el tren de bits, se puede incluir cualquier número de extensiones del conjunto autorizado definido. Sin embargo, cada tipo de extensión no se producirá más de una vez. Cuando un decodificador encuentre una extensión con una identificación de extensión que se describe como "reservado", el decodificador descartará todos los datos que siguen hasta el siguiente código de comienzo. Esto permite la definición futura de extensiones compatibles con este estándar.

6.2 Sintaxis y semántica del bitstream de vídeo

En los siguientes apartados describiré algunos de los detalles que se incorporan en las cabeceras de cada capa para hacernos una idea de la información que porta cada una de ellas.

Cada una de estas capas está formada por unas funciones con una sintaxis determinada que se debe cumplir para formar un bitstream legal. Los códigos se recogen en el Anexo A. Se han incluido en primer lugar, y como comentario, la sintaxis recomendada en el

estándar y a continuación, el código para MATLAB desarrollado. En la mayoría de los casos se ha mantenido la misma estructura de funciones y llamadas a otras funciones, salvo pequeñas excepciones en las que se han incluido funciones dentro de otras o creado funciones no definidas en la recomendación. Será en el capítulo 7 donde se expliquen los detalles más importantes de estas funciones, desde el punto de vista de las tareas que realiza cada una para la decodificación.

6.2.1 Video Sequence

En esta capa podemos englobar varias funciones. La más importante es la función de video_sequence que define el resto de llamadas a otras funciones. Además podemos destacar las funciones de sequence_header y sequence_extension en las que se definen los parámetros básicos para la decodificación del vídeo. Como dijimos la sequence_extension debe aparecer siempre en el caso de MPEG-2. Los parámetros más importantes son:

Sequence header

- **sequence_header_code:** código de comienzo que identifica esta función y está formado por la cadena en hexadecimal '000001B3'.
- **horizontal_size_value:** 12 bits menos significativos del tamaño horizontal de la imagen (horizontal_size). Este tamaño indica el número de muestras en el sentido horizontal que tiene la parte visualizable de la componente de luminancia.
- **vertical_size_value:** 12 bits menos significativos del tamaño vertical de la imagen. (vertical_size). Tiene el mismo significado pero en sentido vertical.
- **frame_rate_code:** 4 bits que se emplean para definir el valor de velocidad de trama en función de una tabla.
- **bit_rate_value:** 18 bits menos significativos de la velocidad binaria.
- **vbv_buffer_size_value:** 10 bits menos significativos que indican el tamaño, en unidades de 16384 bits, que se debe reservar al buffer para la correcta decodifiación.
- **load_intra_quantiser_matrix:** bandera que si está a 1 indica que a continuación está definida la matriz de cuantización para los coeficientes de un macroblock intra.

- **intra_quantiser_matrix:**
- **load_non_intra_quantiser_matrix:** bandera que si está a 1 indica que a continuación está definida la matriz de cuantización para los coeficientes de un macroblock non-intra.
- **non_intra_quantiser_matrix:**

Sequence Extension

- **extension_start_code_identifier:** 4 bits que identifica el tipo de extensión, en este caso tendrá el valor en binario '0001'.
- **profile_and_level_indication:** 8 bits que identifican el perfil y el nivel.
- **progressive_sequence:** bandera que si está a 1 indica que la secuencia de vídeo contiene solamente imágenes de trama progresiva. Si está a 0, pueden aparecer tanto imágenes de trama como de campo, y la imagen de trama pueden ser progesivas o entrelazadas.
- **chroma_format:** 2 bits que indican el formato de crominancia según la siguiente tabla

chroma_format	*Meaning*
00	reserved
01	4:2:0
10	4:2:2
11	4:4:4

Tabla 6.3 Significado de chroma_format

En función de este parámetro también se define el número de bloques que hay en cada macroboque según la tabla 6.4.

chroma_format	*block_count*
4:2:0	6
4:2:2	8
4:4:4	12

Tabla 6.4 Número de bloques en función de chroma_format

- **horizontal_size_extension:** 2 bits más significativos del tamaño horizontal.
- **vertical_size_extension:** 2 bits más significativos del tamaño vertical.

- **bit_rate_extension:** 12 bits más significativos de la velocidad binaria.
- **vbv_buffer_size_extension:** 8 bits más significativos del tamaño del buffer
- **low_delay:** bandera que si está a 1 indica que la secuencia no contiene ninguna imagen B, que no es necesaria la reordenación y que pueden aparecer "imágenes grandes".
- **frame_rate_extension_n:** 2 bits usados para determinar la velocidad de trama.
- **frame_rate_extension_d:** 2 bits usados para determinar la velocidad de trama.

6.2.2 Group of Pictures

Esta capa sólo engloba una función. El grupo de imágenes siempre empieza con una imagen del tipo I, seguido de un número variable de imágenes. Esta capa permite el acceso aleatorio en la secuencia, mediante la búsqueda de su código de comienzo, además que incluye información de tiempo. Los principales parámetros son.

Group of pictures header

- **group_start_code:** código de comienzo de grupo formado por la cadena hexadecimal '000001B8'.
- **time_code:** 25 bits que contienen otros parámetros tal y como se indica en la tabla 6.5. Este código de tiempo se refiere a la primera imagen después del encabezamiento del grupo.

time_code	*Rango de valores*	*Nº de bits*
drop_frame_flag		1
time_code_hours	0 – 23	5
tiem_code_minutes	0 – 59	6
marker_bit	1	1
time_code_seconds	0 – 59	6
time_code_pictures	0 – 59	6

Tabla 6.5 Estructura time_code

- **closed_gop:** bandera que indica la naturaleza de las predicciones utilizadas en las primeras imágenes B consecutivas (si las hubiere) que siguen inmediatamente a la primera imagen del GOP. Si está a 1, las imágenes B se han codificado empleando sólo predicción hacia atrás o intracodificación.

- **broken_link:** bandera que se pone a 0 durante la codificación. Se pone a 1 para indicar que las primeras imágenes B consecutivas que siguen a la primera imagen del GOP pueden ser decodificadas de forma incorrecta porque la imagen de referencia que se utiliza no está disponible

6.2.3 Picture

Esta capa incluye varias funciones. Las más importantes serán picture_header y picture_coding_extension (obligatoria). Ambas contienen información necesaria para la decodificación de cada imagen. Además, la función picture_data es la encargada de llamar a la función slice tantas veces como slices haya definidas. Los parámetros más importantes son:

Picture header

- **picture_start_code:** código de comienzo de imagen formado por la cadena hexadecimal '00000100'.
- **temporal_referenca:** 10 bits que indican una referencia temporal con la posición de la imagen en la visualización del vídeo.
- **picture_coding_type:** 3 bits que identifica si una imagen es tipo I, P o B. El significado de estos bits se detalla en la tabla 6.6.

picture_coding_type	*Método de codificación*
000	Prohibido
001	Intracodificado (I)
010	Con codificación predictiva (P)
011	Con codificación predictiva bidirecciona (B)
100	No se usa
101	Reservado
110	Reservado
111	Reservado

Tabla 6.6 Tipo de codificación de imagen

Picture coding extension

- **extension_start_code_identifier:** 4 bits que identifica el tipo de extensión, en este caso tendrá el valor en binario '1000'.

- **f_code[s][t]:** 4 bits que se utilizan en la decodificación de vectores de movimiento. El valor cero está prohibido y los valores 10 a 14 reservados. En una imagen I debe tomar el valor 15 (no se utiliza)
- **intra_dc_precision:** 2 bits que indica la precisión de los coeficientes DC intracodificados. Se definen sus valores en la tabla 6.7.

intra_dc_precision	***Precisión (bits)***
00	8
01	9
10	10
11	11

Tabla 6.7 Precisión coeficientes DC intracodificados

- **picture_structure:** 2 bits que indican la estructura de la imagen. Cuando una trama se codifica en forma de dos imágenes de campo, ambos campos deben ser del mismo tipo de codificación de imagen, salvo cuando el primer campo codificado es una imagen I, en cuyo caso el segundo puede ser una imagen I o una imagen P. El significado de estos 2 bits lo podemos ver en la tabla 6.8.

picture_structure	***Significado***
00	Reservado
01	Campo superior
10	Campo inferior
11	Imagen de trama

Tabla 6.8 Significado de picture_structure

El primer campo de una trama puede ser un campo superior o inferior y el siguiente debe ser de paridad opuesta.

Cuando una imagen se codifica en forma de dos imágenes de campo, los siguientes elementos de sintaxis pueden fijarse independientemente en cada uno de los campos.

- f_code[0][0], f_code[0][1].
- f_code [1][0], f_code[1][1].
- intra_dc_precision, concealment_motion_vectors, q_scale_type.
- intra_vlc_format, alternate_scan.

  - vbv_delay.
  - temporal_referernce.

- **top_field_first:** 1 bit cuyo significado depende de otros parámetros.
- **frame_pred_frame_dct:** bandera que si está a 1 significa que sólo se utilizan DCT de trama y predicción de trama. En una imagen de campo será 0.
- **concealment_motion_vectors:** bandera que si está a 1 indica que se codifican vectores de movimiento en macrobloques intracodificados.
- **q_acale_type:** bandera que afecta a la cuantificación inversa.
- **intra_vlc_format:** bandera que afecta a la decodificación de datos de coeficientes transformados
- **alternate_scan:** bandera que afecta al orden de escaneo de los coeficientes transformados.
- **repeat_first_field:** bandera que solo se aplica a una imagen de trama.
- **progressive_frame:** bandera que si está a 0 indica que los dos campos de la trama son campos entrelazados entre los cuales existe un intervalo de tiempo de un periodo de campo.

6.2.4 Slice

Esta capa incluye únicamente una función. Es la estructura de vídeo de más bajo nivel que tiene un delimitador de inicio, lo cual la convierte en indispensable para recuperar el sincronismo a nivel de bit en el caso de que se pierda. Los parámetros más importantes son:

Slice

- **slice_start_code:** código de comienzo de slice cuyos primeros tres bytes están formados por la cadena en hexadecimal ‘000001’ y el último byte forma el slice_vetical_position.
- **slice_vertical_position:** 8 bits que dan la posición vertical en unidades de macrobloque (16 líneas) del primer macrobloque de la rebanada. Este campo toma el valor 1 en la primera fila de macrobloques. Algunas slices pueden tener el mismo valor en este campo, ya que puede haber varios slices en cada fila.

- **quantiser_scale_code:** 5 bits que se utiliza en el proceso de cuantificación de los coeficientes transformados. El decodificador emplea este valor hasta que se encuentre otro quantiser_scale_code en el slice o macrobloque. El valor cero está prohibido.
- **intra_slice_flag:** bandera que si está a 1 indica la presencia de otros campos a continuación
- **intra_slice:** bandera que se pone a 0 si cualquiera de los macrobloques de esta slice son macrobloques no intracodificados. Si todos los macrobloques son intracodificados se pone a 1. Si este parámetro se omite se supone a uno.

6.2.5 Macroblock

Esta capa engloba varias funciones que indican cómo decodificar todo tipo de macrobloques y la información de los vectores de movimiento. En una de estas funciones aparece la cabecera de nivel más bajo de toda la sintaxis, aunque sin delimitador específico, por lo que para determinar el comienzo de un macrobloque es necesario haber leído los anteriores a él y para determinar el final, es necesario leer todos los datos del macrobloque en cuestión. Todas las funciones que engloba son importantes, y los parámetros más destacados son:

Macroblock

- **macroblock_address_increment:** código de longitud variable entre 1 y 11 bits que indica la diferencia entre dos parámetros llamados macroblock_address y previous_macroblock_address. La tabla B.1 de la recomendación ISO/IEC 13818-2 se encarga de definirlo.

  El parámetro macroblock_address es una variable que define la posición absoluta del macrobloque actual y el parámetro previous_macroblock_address es una variable que define la posición absoluta del último macrobloque no omitido (non-skipped), salvo al comienzo del slice donde se debe reiniciar.

  Salvo al comienzo de un slice, si el valor de macroblock_address difiere del de previous_macroblock_address en más de uno, algunos macrobloques habrán sido omitidos. Se debe verificar que

  - No haya macrobloques omitidos (skipped macroblocks) en imágenes I.

- En una imagen B no habrá macrobloques omitidos después de un macrobloque en el cual macroblock_intra es uno.

- **quantiser_scale_code:** 5 bits con el mismo significado que en la función slice.

Macroblock modes

- **macroblock_type:** código de longitud variable entre 1 y 9 bits que inica el método de codificación y contenido del macrobloque. Las tablas B.2 a B.8 de la recomendación ISO/IEC 13818-2 se encargan de definirlo, seleccionadas por picture_coding_type y scalable_mode. Se definen los siguientes parámetros a partir de este:
  - **macroblock_quant**
  - **macroblock_motion_forward**
  - **macroblock_motion_backward**
  - **macroblock_pattern**
  - **macroblock_intra**
  - **spatial_temporal_weight_code_flag**
- **frame_motion_type:** 2 bits que indican la predicción de movimiento de macrobloque. Puede no aparecer en la secuencia de bits si frame_pred_frame_dct es igual a 1, en cuyo caso tomará el valor 10. Sus valores se definen en la tabla 6.9 y además en función de su valor definen otros cuatro parámetros.

Cód.	**spatial_temporal_weight_class**	***Tipo de predicción***	**motion_vector count**	**mv_format**	**dmv**
00		Reservado			
01	0, 1	Basada en campo	2	Campo	0
01	2, 3	Basada en campo	1	Campo	0
10	0, 1, 2, 3	Basada en trama	1	Trama	0
11	0, 2, 3	Dual-Prima	1	Campo	1

Tabla 6.9 Significado frame_motion_type

- **motion_vector_count:** indica el número de vectores de movimiento.
- **mv_format:** indica si el vector de movimiento es un vector de movimiento de campo o de trama.

- **Dmv**
- **spatial_temporal_wieght_class**

- **field_motion_type:** 2 bits que indican la predicción de movimiento de macrobloque. Sus valores se definen en la tabla 6.10 y además en función de su valor definen los mismos parámetros que en el caso anterior.

Cód.	**spatial_temporal_weight_class**	***Tipo de predicción***	**motion_vector count**	**mv_format**	**dmv**
00		Reservado			
01	0, 1	Basada en campo	1	Campo	0
10	0, 1	16 x 8 MC		Campo	0
11	0	Dual-Prima	1	Campo	1

Tabla 6.10 Siginficado field_motion_type

- **dct_type:** bandera que indica si el macrobloque es DCT de trama codificada o DCT de campo codificada.

Motion vectors

Emplea los parámetros definidos por frame_motion_type o field_motion_type y es desde donde se hacen las llamadas a la siguiente función.

- **motion_vertical_field_select[r][s]:** bandera que indica qué campo de referencia se usa para formar la predicción. Si es 0 se usa el campo de referencia superior y viceversa.

Motion vector

- **motion_code[r][s][t]:** código de longitud variable entre 1 y 11 bits que se utiliza para la decodificación del vector de movimiento. La tabla B.10 de la recomendación ISO/IEC 13818-2 se encarga de definirlo.
- **motion_residual[r][s][t]:** 8 bits que se utilizan en la decodificación del vector de movimiento.
- **dmvector[t]:** código de longitud variable que se utiliza en la decodificación del vector de movimiento. La tabla B.11 de la recomendación ISO/IEC 13818-2 se encarga de definirlo.

Coded block pattern

- **Coded_block_pattern_420:** código de longitud variable que se usa para definir la variable cbp de acuerdo con la tabla B.9 de la recomendación ISO/IEC 13818-2. esta variable servirá para saber qué bloques están codicados.

6.2.6 Block

Esta capa engloba una única función. Contiene la información relativa a la componente de continua (DC), los coeficientes AC y la información de finalización de la codificación de bloque (EOB). Los parámetros más importantes son:

Block

- **dct_dc_size_luminance:** código de longitud variable entre 2 y 9 bits que indica la longitud del campo dct_dc_differential para la luminancia en función de la tabla B.12 de la recomendación ISO/IEC 13818-2.
- **dct_dc_size_chrominance:** código de longitude variable entre 2 y 10 bits que indica la longitud del campo dct_dc_differential para la crominancia en función de la tabla B.13 de la recomendación ISO/IEC 13818-2.
- **dct_dc_differential:** código de lontitud fija definida por los campos anteriores y que contiene un valor diferencial al que se añade un predictor para recuperar el coeficiente real (habrá tres predictores , uno para cada una de las componentes de color. Después, el predictor se pondrá al valor del coeficiente que acaba de decodificarse. En distintas ocasiones los predictores serán reiniciados. El valor de reiniciación se deriva del parámetro intra_dc_precision, como se detalla en la tabla 6.11.

intra_dc_precision	*Bits de precisión*	*Valor de reiniciación*
0	8	128
1	9	256
2	10	512
3	11	1024

Tabla 6.11 Relación entre intra_dc_precision y el valor de reiniciación del predictor

Como hemos dicho, los predictores se reiniciarán en los momentos siguientes:

- Al principio de un slice.

- Cuando se decodifica un macrobloque no intracodificado (non_intra).
- Siempre que se omita un macrobloque.

- **First DCT coefficient:** código de longitud variable entre 2 y 24 bits que proporciona el valor del coeficiente DC en el caso de que no sea un macrobloque intracodificado. La tabla B.14 y B.16 de la recomendación ISO/IEC 13818-2 se encarga de definirlo.
- **Subsequent DCT coefficients:** código de longitud variable entre 3 y 24 bits que proporciona el valor de los coeficientes AC. Las tablas B.14, B.15 y B.16 de la recomendación ISO/IEC 13818-2 se encargan de definirlo. La elección entre la tabla B.14 y B.15 se hace en función de la tabla 6.12.

	Intra_vlc_format	
	0	***1***
Intra blocks (macroblock_intra=1)	B-14	B-15
Non-intra blocks (macroblock_intra=0)	B-14	B-14

Tabla 6.12 Selección de tablas VLC de coeficientes DC

- **End of block:** 2 o 4 bits que indican el final del bloque.

6.3 Extensiones escalables

Los instrumentos de escalabilidad se diseñan para sustentar otras aplicaciones además de las admitidas por el vídeo de una sola capa. Entre las aplicaciones más importantes podemos citar:

- Las telecomunicaciones de vídeo.
- El vídeo por redes en el modo de transferencia asíncrono (ATM)
- Interfuncionamiento de normas de vídeo
- Jerarquías de servicios de vídeo con múltiples resoluciones espaciales, temporales y de calidad
- Televisión de alta definición (HDTV) con televisión incrustad

- Sistemas que permiten pasar a televisión de alta definición con resolución temporal más alta.

Los instrumentos de escalabilidad básicos ofrecidos son:

- Partición de datos
- Escalabilidad SNR (relación señal/ruido)
- Escalabilidad espacial
- Escalabilidad temporal

Además, se admiten también combinaciones de estas técnicas formando una esalonabilidad híbrida

6.3.1 Extensión escalable espacial

Se emplea en sistemas de vídeo con la característica común de que necesitan un mínimo de dos capas de resolución espacial. Por lo tanto, se generan dos capas de vídeo a partir de una sola fuente de video. La capa más baja es codificada pos sí misma para proporcionar la resolución espacial básica y la capa de mejora emplea la capa anterior y transporta la resolución espacial completa.

Con esta técnica se puede dar el caso de que en la capa más baja se emplee la información de vídeo del estándar MPEG-1 y en la capa de mejora los aspectos necesarios de MPEG-2.

Este tipo de escalabilidad ofrece flexibilidad en la elección de los formatos de vídeo que se han de emplear en cada capa. Otra ventaja es la capacidad de proporcionar elasticidad a los errores de transmisión, pues los datos más importantes de la capa más baja se pueden enviar por el canal con una mejor característica de error, mientras que los datos de la capa de mejora, menos críticos, se pueden enviar por un canal con una característica de error mediocre.

6.3.2 Extensión escalable SNR

Se emplea en sistemas de vídeo con la característica común de que necesitan un mínimo de dos capas de resolución espacial. Por lo tanto, se generan dos capas de vídeo a partir de una sola fuente de video. Por lo tanto, se generan dos capas de vídeo con la misma resolución espacial pero con diferentes calidades de vídeo a partir de una sola fuente

vídeo, de modo que la capa más baja es codificada por sí misma para proporcionar la calidad de vídeo básica y la capa de mejora se codifica para mejorar la capa más baja.

La capa de mejora, cuando se añade a la capa más baja, regenera una reproducción de calidad más alta del vídeo de entrada.

Al igual que en el caso anterior podemos encontrar en la capa más baja MPEG-1. De nuevo, esta técnica ofrece elasticidad a los errores de transmisión.

6.3.3 Extensión escalable temporal

Se emplea en sistemas de vídeo en los que puede ser necesario pasar de sistemas de resolución temporal más baja a sistemas de resolución temporal más alta. En muchos casos, los sistemas vídeo de resolución temporal más baja pueden ser los sistemas existentes o los sistemas de la generación anterior menos costosos, con la motivación de introducir gradualmente sistemas más complejos.

La escalabilidad temporal conlleva la partición de tramas vídeo en capas, de modo que la capa más baja es codificada por sí misma para proporcionar la velocidad temporal básica y la capa de mejora se codifica con predicción temporal con respecto a la capa más baja; estas capas son decodificadas y multiplexadas temporalmente para obtener toda la resolución temporal de la fuente vídeo. Los sistemas de resolución temporal más baja sólo pueden decodificar la capa más baja para proporcionar resolución temporal básica, mientras que los sistemas más complejos del futuro podrán decodificar ambas capas y proporcionar vídeo de resolución temporal alta a la vez que mantienen el interfuncionamiento con los sistemas de la generación anterior.

Nuevamente ofrece elasticidad a los errores de transmisión.

6.3.4 Extensión de partición de datos

La partición de datos es un instrumento concebido para ser utilizado cuando se dispone de dos canales para transmisión y/o almacenamiento de un tren de bits de vídeo. El tren de bits se divide entre estos canales de modo que las partes más críticas del tren de bits (tales como los encabezamientos, vectores de movimiento, coeficientes DCT de baja frecuencia) se transmiten por el canal con la mejor característica de error y los datos menos críticos (tales como coeficientes DCT de frecuencia más alta) se transmiten por el canal con una característica de error mediocre. De este modo, se minimiza la

degradación debida a los errores de canal porque las partes críticas de un tren de bits están mejor protegidas. Los datos de cualquiera de los dos canales pueden ser decodificados en un decodificador que no esté concebido para decodificar trenes de bits de datos divididos.