



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

# UIT-T

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

# H.263

**Anexo W**  
(11/2000)

SERIE H: SISTEMAS AUDIOVISUALES Y  
MULTIMEDIOS

Infraestructura de los servicios audiovisuales –  
Codificación de imágenes vídeo en movimiento

---

Codificación de vídeo para comunicación a baja  
velocidad binaria

**Anexo W: Especificación de información  
adicional de mejora suplementaria**

Recomendación UIT-T H.263 – Anexo W

(Anteriormente Recomendación del CCITT)

---

RECOMENDACIONES UIT-T DE LA SERIE H  
SISTEMAS AUDIOVISUALES Y MULTIMEDIOS

CARACTERÍSTICAS DE LOS SISTEMAS VIDEOTELEFÓNICOS	H.100–H.199
INFRAESTRUCTURA DE LOS SERVICIOS AUDIOVISUALES	
Generalidades	H.200–H.219
Multiplexación y sincronización en transmisión	H.220–H.229
Aspectos de los sistemas	H.230–H.239
Procedimientos de comunicación	H.240–H.259
<b>Codificación de imágenes vídeo en movimiento</b>	<b>H.260–H.279</b>
Aspectos relacionados con los sistemas	H.280–H.299
SISTEMAS Y EQUIPOS TERMINALES PARA LOS SERVICIOS AUDIOVISUALES	H.300–H.399
SERVICIOS SUPLEMENTARIOS PARA MULTIMEDIOS	H.450–H.499

*Para más información, véase la Lista de Recomendaciones del UIT-T.*

## Recomendación UIT-T H.263

### Codificación de vídeo para comunicación a baja velocidad binaria

#### ANEXO W

### Especificación de información adicional de mejora suplementaria

#### Resumen

*Información adicional suplementaria de mejora* opcional que puede ser añadida a un tren de bits H.263 para proporcionar compatibilidad mejorada hacia atrás, incluyendo:

- Indicación del uso de una IDCT de punto fijo específica.
- Mensajes de imagen, incluyendo los tipos de mensaje siguientes:
  - datos binarios arbitrarios;
  - texto (arbitrario, derechos de autor, subtítulo, descripción de vídeo, o identificador uniforme de recurso);
  - repetición de encabezamiento de imagen (actual, previa, siguiente con referencia temporal fiable, o siguiente con referencia temporal no fiable);
  - indicaciones de campo entrelazado (superior o inferior); e
  - identificación de imagen de referencia de reserva.

#### Orígenes

El anexo W a la Recomendación UIT-T H.263, preparado por la Comisión de Estudio 16 (2001-2004) del UIT-T, fue aprobado por el procedimiento de la Resolución 1 de la AMNT el 17 de noviembre de 2000.

## PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución 1 de la AMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

## NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

## PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 2001

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

## ÍNDICE

	<b>Page</b>
Anexo W – Especificación de información adicional de mejora suplementaria.....	1
W.1 Introducción .....	1
W.2 Referencias.....	1
W.3 Valores adicionales de FTYPE .....	1
W.4 Número máximo recomendado de octetos PSUPP .....	1
W.5 IDCT de punto fijo.....	2
W.5.1 Funcionamiento del decodificador .....	2
W.5.2 Eliminación de la actualización forzada.....	2
W.5.3 Referencia IDCT 0.....	2
W.6 Mensaje de imagen .....	10
W.6.1 Continuación (CONT) (1 bit) .....	11
W.6.2 Posición del bit final (EBIT, <i>end bit position</i> ) o número de pista (3 bits).....	11
W.6.3 Tipo de mensaje ( <i>MTYPE, message type</i> ) (4 bits).....	11

## Recomendación UIT-T H.263

### Codificación de vídeo para comunicación a baja velocidad binaria

#### ANEXO W

### Especificación de información adicional de mejora suplementaria

#### W.1 Introducción

Este anexo describe el formato de la información adicional de mejora suplementaria enviada en el campo PSUPP de la capa de imagen especificada en la Recomendación H.263, que amplía la funcionalidad definida en el anexo L. Se puede señalar por medios externos (por ejemplo, utilizando UIT-T H.245) la capacidad de un decodificador para proporcionar alguna, o todas, las capacidades descritas en el presente anexo. Los decodificadores que no soporten las capacidades adicionales pueden simplemente descartar cualesquiera de los bits de información PSUPP de nueva definición que aparecen en el tren de bits. La presencia de esta información de mejora suplementaria viene indicada por la presencia de tanto el bit PEI como la del octeto PSUPP que le sigue, y cuyo campo FTYPE tenga uno de los dos valores de nueva definición. La interpretación básica de PEI, PSUPP, FTYPE, y DSIZE es idéntica a la del anexo L y a la de las cláusulas 5.1.24 y 5.1.25.

#### W.2 Referencias

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes.

- ISO/CEI 10646-1:2000, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*.
- IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax*.

#### W.3 Valores adicionales de FTYPE

Los dos valores que aparecían como reservados en el cuadro L.1 del anexo L, se definen en el cuadro W.1.

**Cuadro W.1/H.263 – Valores del tipo de función FTYPE**

13	IDCT de punto fijo
14	Mensaje de imagen

#### W.4 Número máximo recomendado de octetos PSUPP

Cuando se utilice alguna de las funciones FTYPE antes citadas, y que se definen en este anexo, se deberá mantener razonablemente pequeño, en relación con el tamaño de la imagen codificada, el

número total de octetos PSUPP por imagen, que no deberá ser mayor de 256 octetos, con independencia del tamaño de la imagen codificada.

NOTA – Algunos de los protocolos de transmisión utilizados en el transporte del tren de bits de vídeo pueden dar soporte a la repetición externa de los contenidos de los encabezamientos de las imágenes con el fin de mejorar la elasticidad a errores, y pueden limitar la cantidad de dichos datos que se pueden repetir de un encabezamiento de imagen (por ejemplo, 504 bits en el formato de paquetes de IETF RFC 2429). La inclusión de un número mayor de octetos PSUPP puede conllevar la falta de un protocolo externo de este tipo que proporcione la repetición completa de los contenidos de los encabezamientos de imagen.

## W.5 IDCT de punto fijo

La función IDCT de punto fijo indica el uso de una aproximación IDCT particular en la construcción del tren de bits. DSIZE será igual a 1 para la función IDCT de punto fijo. El octeto de datos PSUPP que sigue, especifica la implementación IDCT particular. Un valor igual a 0 indica la referencia IDCT 0, como se describe en W.5.3; los valores 1 a 255 están reservados.

### W.5.1 Funcionamiento del decodificador

La capacidad de un decodificador para llevar a cabo una IDCT de punto fijo determinada, puede ser señalizada al codificador utilizando medios externos (por ejemplo, UIT-T H.245). Al recibir un tren de bits codificados con la indicación de IDCT de punto fijo, un decodificador deberá utilizar la IDCT de punto fijo en cuestión, si es capaz de hacerlo.

### W.5.2 Eliminación de la actualización forzada

El anexo A especifica los requisitos de exactitud para la transformada discreta de coseno inversa (IDCT), permitiendo numerosas implementaciones que los cumplen. Para controlar la acumulación de errores debidos al desajuste de las IDCT entre el codificador y el decodificador, la cláusula 4.4 que trata de la actualización forzada, exige que los macrobloques se codifiquen en modo INTRA al menos una vez cada 132 veces cuando se transmiten los coeficientes.

Si el tren de bits indica el tipo de función IDCT de punto fijo, entonces se elimina el requisito de actualización forzada, y se deja sin regular la frecuencia a la que debe producirse la codificación INTRA. Sin embargo, un codificador deberá seguir utilizando la actualización forzada, a no ser que se haya asegurado por medios externos de que el decodificador es capaz de operar con la IDCT de punto fijo particular aquí especificada; en otro caso, se producirá un desajuste.

### W.5.3 Referencia IDCT 0

La referencia IDCT 0 es cualquier implementación que, para cada bloque de entrada, genere valores de salida idénticos a los del programa C con el listado de código fuente que sigue.

NOTA – Esta IDCT de punto fijo cumple con el anexo A/H.263, pero no cumple con el requisito de rango ampliado de valores del anexo A de UIT-T Rec. H.262 | ISO/CEI 13818-2.

```
/*
 *
 *                               FIXED-POINT IDCT
 *
 * Fixed-point fast, separable idct
 * Storage precision: 16 bits signed
 * Internal calculation precision: 32 bits signed
 * Input range: 12 bits signed, stored in 16 bits
 * Output range: [-256, +255]
 * All operations are signed
 *
 */
/*
 * Includes
 */
```

```

#include <stdlib.h>
#include <stdio.h>

/*
 * Typedefs
 */

typedef short int REGISTER; /* 16 bits signed */
typedef long int LONG; /* 32 bits signed */

/*
 * Global constants
 */

const REGISTER cpo8 = 0x539f; /* 32768*cos(pi/8)*1/sqrt(2) */
const REGISTER spo8 = 0x4546; /* 32768*sin(pi/8)*sqrt(2) */
const REGISTER cpo16 = 0x7d8a; /* 32768*cos(pi/16) */
const REGISTER spo16 = 0x18f9; /* 32768*sin(pi/16) */
const REGISTER c3po16 = 0x6a6e; /* 32768*cos(3*pi/16) */
const REGISTER s3po16 = 0x471d; /* 32768*sin(3*pi/16) */
const REGISTER Oor2 = 0x5a82; /* 32768*1/sqrt(2) */

/*
 * Function declarations
 */

void Transpose(REGISTER block[64]);
void HalfSwap(REGISTER block[64]);
void Swap(REGISTER block[64]);
void Scale(REGISTER block[64], signed char sh);
void Round(REGISTER block[64], signed char sh,
           const REGISTER min, const REGISTER max);
REGISTER Multiply(const REGISTER a, REGISTER x, signed char sh);
void Rotate(REGISTER *x, REGISTER *y,
            signed char sha, signed char shb,
            const REGISTER a, const REGISTER b,
            int inv);
void Butterfly(REGISTER column[8], char pass);
void IDCT(REGISTER block[64]);

/*
 * Transpose():
 *   Transpose a block
 * Input:
 *   REGISTER block[64]
 * Output:
 *   block
 * Return value:
 *   none
 */
void Transpose(REGISTER block[64])
{
    int i, j;
    REGISTER temp;

    for (i=0; i<8; i++) {
        for (j=0; j<i; j++) {
            temp = block[8*i+j];
            block[8*i+j] = block[8*j+i];
            block[8*j+i] = temp;
        }
    }
    return;
}

/*
 * HalfSwap():
 *   One-dimensional swap
 * Input:
 *   REGISTER block[64]

```



```

* Output:
*   block
* Return value:
*   none
*/
void HalfSwap(REGISTER block[64])
{
    int i;
    REGISTER temp;

    for (i=0; i<8; i++) {
        temp = block[8+i];
        block[8+i] = block[32+i];
        block[32+i] = temp;
        temp = block[24+i];
        block[24+i] = block[48+i];
        block[48+i] = temp;
        temp = block[40+i];
        block[40+i] = block[56+i];
        block[56+i] = temp;
    }
    return;
}

/*
* Swap():
*   Swap and transpose a block
* Input:
*   REGISTER block[64]
* Output:
*   block
* Return value:
*   none
*/
void Swap(REGISTER block[64])
{
    HalfSwap(block);
    Transpose(block);
    HalfSwap(block);
}

/*
* Scale():
*   Scale a block
* Input:
*   REGISTER block[64]
*   signed char sh
* Output:
*   block
* Return value:
*   none
*/
void Scale(REGISTER block[64], signed char sh)
{
    int i;

    if (sh>0) {
        for (i=0; i<64; i++)
            block[i] >>= sh;
    }
    else {
        for (i=0; i<64; i++)
            block[i] <<= -sh;
    }
}

/*
* Round():
*   Performs the final rounding of an 8x8 block
* Input:
*   REGISTER block[64]

```

```

*      signed char sh
*      const REGISTER min
*      const REGISTER max
* Output:
*      block
* Return value:
*      none
*/
void Round(REGISTER block[64], signed char sh,
           const REGISTER min, const REGISTER max)
{
    int i;

    for (i=0; i<64; i++) {
        if (block[i] < 0x00007FFF - (1<<(sh-1)))
            block[i] += (1<<(sh-1));
        else
            block[i] = 0x00007FFF;
        block[i] >>= sh;
        block[i] = (block[i]<min) ? min : ((block[i]>max) ? max : block[i]);
    }
    return;
}

```

```

/*
* Multiply():
*      Multiply by a constant with shift
* Input:
*      const REGISTER a
*      REGISTER x
*      signed char sh
* Output:
*      none
* Return value:
*      REGISTER, the result of the multiply
*/
REGISTER Multiply(const REGISTER a, REGISTER x, signed char sh)

```

```

{
    LONG tmp;
    REGISTER reg_out;

    /* multiply */
    tmp = (LONG)a * (LONG)x;

    /* shift */
    if (sh > 0)
        tmp >>= sh;
    else
        tmp <<= -sh;

    /* rounding and saturating */
    if (tmp < 0x7FFFFFFF - 0x00007FFF)
        tmp = tmp + 0x00007FFF;
    else
        tmp = 0x7FFFFFFF;

    reg_out = (REGISTER)(tmp >>16);

    return(reg_out);
}

```

```

/*
* Rotate():
*      Perform rotate operation on two registers
* Input:
*      REGISTER *x          pointer to the 1st register
*      REGISTER *y          pointer to the 2nd register
*      signed char sha      shift associated with factor a
*      signed char shb      shift associated with factor b
*      const REGISTER a     factor a
*      const REGISTER b     factor b

```

```

*      int inv          1 for inverse dct, 0 for forward dct
* Output:
*      *x, *y
* Return value:
*      none
*/
void Rotate(REGISTER *x, REGISTER *y,
            signed char sha, signed char shb,
            const REGISTER a, const REGISTER b,
            int inv)
{
    LONG tmp1a, tmp1y, tmp1x, tmp1y;
    LONG tmp11, tmp12;

    /*
     * intermediate calculation
     */

    tmp1a = (LONG)(*x) * (LONG)a;
    if (sha > 0)
        tmp1a >>= sha;
    else
        tmp1a <<= -sha;

    tmp1y = (LONG)(*y) * (LONG)a;
    if (sha > 0)
        tmp1y >>= sha;
    else
        tmp1y <<= -sha;

    tmp1x = (LONG)(*x) * (LONG)b;
    if (shb > 0)
        tmp1x >>= shb;
    else
        tmp1x <<= -shb;

    tmp1y = (LONG)(*y) * (LONG)b;
    if (shb > 0)
        tmp1y >>= shb;
    else
        tmp1y <<= -shb;

    /*
     * rounding and rotation
     */

    if (inv) {
        tmp1a += 0x00007FFF;
        tmp1x += 0x00007FFF;

        tmp11 = tmp1x - tmp1y;
        tmp12 = tmp1a + tmp1y;
    }
    else {
        tmp1y += 0x00007FFF;
        tmp1y += 0x00007FFF;

        tmp11 = tmp1x + tmp1y;
        tmp12 = -tmp1a + tmp1y;
    }

    /*
     * final rounding
     */

    *x = (REGISTER) (tmp11 >>16);
    *y = (REGISTER) (tmp12 >>16);

    return;
}

```

```

/*
 * Butterfly():
 *   Perform 1D IDCT on a column
 * Input:
 *   REGISTER column[8]
 *   char pass
 * Output:
 *   column
 * Return value:
 *   none
 */
void Butterfly(REGISTER column[8], char pass)
{
    int i;
    REGISTER shadow_column[8];

    /*
     * For readability, we use a shadow column
     * that contains the state of column at the
     * preceding stage of the butterfly.
     */

    /*
     * Initialization
     */

    for (i=0; i<8; i++)
        shadow_column[i] = column[i];

    /*
     * First Phase
     */

    Rotate(column+2, column+6, pass-2, pass-1, cpo8, spo8, 1);
    Rotate(column+1, column+7, pass-1, pass-1, cpo16, spo16, 1);
    Rotate(column+3, column+5, pass-1, pass-1, c3po16, s3po16, 1);

    if (pass) {
        int a, tmp=column[4], b=column[0];
        a = b+tmp;
        b = b-tmp;
        column[0] = (a - ((tmp<0) ? 1 : 0)) >> 1;
        column[4] = (b - ((tmp<0) ? 1 : 0)) >> 1;
    }
    else {
        column[0] = shadow_column[0] + shadow_column[4];
        column[4] = shadow_column[0] - shadow_column[4];
    }

    for (i=0; i<8; i++)
        shadow_column[i] = column[i];

    /*
     * Second Phase
     */

    column[1] = shadow_column[1] - shadow_column[3];
    column[3] = shadow_column[1] + shadow_column[3];

    column[7] = shadow_column[7] - shadow_column[5];
    column[5] = shadow_column[7] + shadow_column[5];

    column[0] = shadow_column[0] + shadow_column[6];
    column[6] = shadow_column[0] - shadow_column[6];

    column[4] = shadow_column[4] + shadow_column[2];
    column[2] = shadow_column[4] - shadow_column[2];

    for (i=0; i<8; i++)
        shadow_column[i] = column[i];

```

```

/*
 * Third Phase
 */

column[7] = shadow_column[7] - shadow_column[3];
column[3] = shadow_column[7] + shadow_column[3];

column[1] = Multiply(OoR2, shadow_column[1], -2);
column[5] = Multiply(OoR2, shadow_column[5], -2);

for (i=0; i<8; i++)
    shadow_column[i] = column[i];

/*
 * Fourth Phase
 */

column[4] = shadow_column[4] + shadow_column[3];
column[3] = shadow_column[4] - shadow_column[3];

column[2] = shadow_column[2] + shadow_column[7];
column[7] = shadow_column[2] - shadow_column[7];

column[0] = shadow_column[0] + shadow_column[5];
column[5] = shadow_column[0] - shadow_column[5];

column[6] = shadow_column[6] + shadow_column[1];
column[1] = shadow_column[6] - shadow_column[1];

return;
}

/*
 * IDCT():
 * Perform 2D IDCT on a block
 * Input:
 * REGISTER block[64]
 * Output:
 * block
 * Return value:
 * none
 */
void IDCT(REGISTER block[64])
{
    int i;

    Scale(block, -4);

    for (i=0; i<8; i++)
        Butterfly(block+8*i, 0);

    Transpose(block);

    for (i=0; i<8; i++)
        Butterfly(block+8*i, 1);

    Round(block, 6, -256, 255);

    Swap(block);
}

```

Con fines informativos se presenta a continuación la implementación de una transformada discreta del coseno directa (FDCT, *forward discrete cosine transform*), que tiene relación con los temas aquí tratados, aunque no forma parte integral de la presente Recomendación.

```

/*****
*
*                               FIXED-POINT FDCT
*
* Fixed-point fast, separable fdct
* Storage precision: 16 bits signed
* Internal calculation precision: 32 bits signed
* Input range: 9 bits signed, stored in 16 bits
* Output range: [-2048, +2047]
* All operations are signed
*
*****/

/*
* Function declarations
*/

void FButterfly(REGISTER column[8]);
void FDCT(REGISTER block[64]);

/*
* FButterfly():
*   Perform 1D FDCT on a column
* Input:
*   REGISTER column[8]
* Output:
*   column
* Return value:
*   none
*/
void FButterfly(REGISTER column[8])
{
    int i;
    REGISTER shadow_column[8];

    /*
     * For readability, we use a shadow column
     * that contains the state of column at the
     * preceding stage of the butterfly.
     */

    /*
     * Initialization
     */

    for (i=0; i<8; i++)
        shadow_column[i] = column[i];

    /*
     * First Phase
     */

    for (i=0; i<4; i++) {
        column[i] = shadow_column[i] + shadow_column[7-i];
        column[7-i] = shadow_column[i] - shadow_column[7-i];
    }

    for (i=0; i<8; i++)
        shadow_column[i] = column[i];

    /*
     * Second Phase
     */

    column[0] = shadow_column[0] + shadow_column[3];
    column[3] = shadow_column[0] - shadow_column[3];

    column[1] = shadow_column[1] + shadow_column[2];
    column[2] = shadow_column[1] - shadow_column[2];
}

```

```

column[4] = Multiply(OoR2, shadow_column[4], -2);
column[7] = Multiply(OoR2, shadow_column[7], -2);

column[6] = shadow_column[6] - shadow_column[5];
column[5] = shadow_column[6] + shadow_column[5];

for (i=0; i<8; i++)
    shadow_column[i] = column[i];

/*
 * Third Phase
 */

column[0] = shadow_column[0] + shadow_column[1];
column[1] = shadow_column[0] - shadow_column[1];

column[6] = shadow_column[6] - shadow_column[4];
column[4] = shadow_column[6] + shadow_column[4];

column[7] = shadow_column[7] - shadow_column[5];
column[5] = shadow_column[7] + shadow_column[5];

for (i=0; i<8; i++)
    shadow_column[i] = column[i];

/*
 * Fourth Phase
 */

Rotate(column+2, column+3, -2, -1, cpo8, spo8, 0);
Rotate(column+4, column+5, -1, -1, cpo16, spo16, 0);
Rotate(column+6, column+7, -1, -1, c3po16, s3po16, 0);

return;
}

/*
 * FDCT():
 *     Perform 2D FDCT on a block
 * Input:
 *     REGISTER block[64]
 * Output:
 *     block
 * Return value:
 *     none
 */
void FDCT(REGISTER block[64])
{
    int i;

    for (i=0; i<8; i++)
        FButterfly(block+8*i);

    Transpose(block);

    for (i=0; i<8; i++)
        FButterfly(block+8*i);

    Round(block, 3, -2048, 2047);

    Swap(block);
}

```

## W.6 Mensaje de imagen

La función mensaje de imagen indica la presencia de uno o más octetos que representan datos del mensaje. El primer octeto de los datos del mensaje es un encabezamiento de mensaje con la siguiente estructura, como se muestra en la figura W.1.

CONT	EBIT	MTYPE
------	------	-------

**Figura W.1/H.263 – Estructura del primer octeto del mensaje**

DSIZE será igual al número de octetos en los datos del mensaje que corresponden a una función mensaje de imagen, y que incluye al primer octeto que se muestra en la figura W.1.

Los decodificadores deberán analizar los datos del mensaje de imagen como requiere la sintaxis PSUPP básica, aunque, por otra parte está sin definir la respuesta del decodificador a los mensajes de imagen.

### **W.6.1 Continuación (CONT) (1 bit)**

Si es igual a "1", CONT indica que los datos del mensaje asociados con esta función del mensaje de imagen son parte del mismo mensaje lógico que los datos del mensaje asociados con la siguiente función del mensaje de imagen. Si es igual a "0", CONT indica que los datos del mensaje asociados con esta función del mensaje de imagen termina el actual mensaje lógico. Se puede utilizar CONT, por ejemplo, para incluir mensajes lógicos que se extienden por más de 14 octetos.

### **W.6.2 Posición del bit final (EBIT, *end bit position*) o número de pista (3 bits)**

Para mensajes de imágenes que no son textos, EBIT especifica el número de los bits menos significativos que serán ignorados en el octeto del último mensaje. En mensajes de imágenes que no son textos, si CONT es "1", o si sólo hay un octeto de mensaje, (es decir el octeto de la figura W.1), EBIT será igual "0". El número de bits del mensaje válidos para una función de mensaje de imagen que no es texto, sin tener en cuenta los bits CONT/EBIT/MTYPE, es igual a  $(DSIZE - 1) \times 8 - EBITS$ . El número de bits válidos del mensaje para un mensaje lógico puede ser mayor debido a la prolongación del mensaje.

Para tipos de mensajes de imagen que contienen información de texto, EBIT contendrá un número de pista de texto. El significado preciso del número de pista de texto no se especifica aquí, pero deberá indicar un tipo particular (por ejemplo, idioma) del texto. Se considerará un número de pista cero como pista por defecto.

### **W.6.3 Tipo de mensaje (MTYPE, *message type*) (4 bits)**

MTYPE indica el tipo de mensaje. Los tipos definidos se presentan en el cuadro W.2.

**Cuadro W.2/H.263 – Valores para el tipo de mensaje, MTYPE**

0	Datos binarios arbitrarios
1	Texto arbitrario
2	Texto de derechos de autor
3	Texto de subtítulo
4	Texto de descripción de vídeo
5	Texto de identificador uniforme de recurso
6	Repetición del encabezamiento de la imagen presente
7	Repetición del encabezamiento de la imagen previa
8	Repetición del encabezamiento de la imagen siguiente, TR fiable
9	Repetición del encabezamiento de la imagen siguiente, TR no fiable
10	Indicación de campo entrelazado superior



**Cuadro W.2/H.263 – Valores para el tipo de mensaje, MTYPE (*fin*)**

11	Indicación de campo entrelazado inferior
12	Número de imagen
13	Imágenes de referencia de reserva
14..15	Reservado

### **W.6.3.1 Datos binarios arbitrarios**

El tipo datos binarios arbitrarios se utiliza para transportar cualquier mensaje binario codificado que no sea del tipo ISO/CEI 10646-1 UTF-8. La interpretación de los contenidos de los datos binarios arbitrarios está fuera del alcance de la presente Recomendación, aunque deberán empezar con algún patrón identificativo (por ejemplo, un código identificador de cuatro octetos) que ayude a distinguir a uno entre estos tipos de datos.

### **W.6.3.2 Texto arbitrario**

Se utiliza el texto arbitrario para transportar un mensaje genérico de texto codificado según ISO/CEI 10646-1 UTF-8. Mensajes de texto más específicos, tal como los de información de derechos de autor, se representarán por otros tipos de mensajes a propósito (por ejemplo, texto de derechos de autor).

### **W.6.3.3 Texto de derechos de autor**

El texto de derechos de autor se utilizará sólo para el transporte de información de propiedad intelectual en relación con la autoría de la representación codificada en el tren de bits. El mensaje derechos de autor se codificará según ISO/CEI 10646-1 UTF-8.

### **W.6.3.4 Texto de subtítulo**

El texto de subtítulo se utilizará únicamente para transportar la información de subtítulo asociada con la imagen actual, y con las subsiguientes del tren de bits. El mensaje de subtítulo se codificará de acuerdo con ISO/CEI 10646-1 UTF-8. El texto de subtítulo estará insertado en el tren de bits como si se fuese a presentar en un área de texto separada en la que se añade texto nuevo al final del texto previo, y el texto preexistente se desplaza a partir del punto de inserción. El código de control de Avance de Página (hexadecimal "0x000C") se utilizará para indicar el borrado del área de texto visible. El código de control Fin del Medio Físico (hexadecimal "0x0019") se utilizará para indicar la situación del "fin de texto de subtítulo". Sin embargo, esta Recomendación no pone restricciones sobre la forma en la que en realidad se presenta y almacena este texto de subtítulo.

### **W.6.3.5 Texto de descripción de vídeo**

Se utilizará el texto de descripción de vídeo sólo para transportar información descriptiva asociada con el contenido de la información del tren de bits actual. La descripción de vídeo se codificará de acuerdo con ISO/CEI 10646-1 UTF-8. El texto de la descripción de vídeo se insertará en el tren de bits como si fuese a ser presentado en un área de texto separada en la que el nuevo texto se añade al final del texto previo, y el texto preexistente se desplaza a partir del punto de inserción. El código de control de Avance de Página (hexadecimal "0x000C") se utilizará para indicar el borrado del área de texto visible. El código de control Fin del Medio Físico (hexadecimal "0x0019") se utilizará para indicar la situación del "fin de texto de descripción". Sin embargo, esta Recomendación no pone restricciones sobre la forma en la que en realidad se presenta y almacena este texto de descripción de vídeo.

### **W.6.3.6 Identificador uniforme de recurso (URI, *uniform resource identifier*)**

El mensaje consiste en un identificador uniforme de recurso (URI) del tipo definido en IETF RFC 2396. Se codificará el URI de acuerdo con ISO/CEI 10646-1 UTF-8.

### **W.6.3.7 Repetición del encabezamiento de la imagen presente**

En este mensaje se repite el encabezamiento de imagen de la imagen presente. Los bits repetidos no incluyen ninguna información de mejora suplementaria (PEI/PSUPP). Deberán incluirse todos los demás bits hasta los de las capas de GOB o de ranura, sujetos a las limitaciones de W.4.

### **W.6.3.8 Repetición del encabezamiento de la imagen previa**

En este mensaje se repite el encabezamiento de imagen de la imagen previamente transmitida. Los bits repetidos no incluyen a los dos primeros bytes del código de comienzo de imagen (PSC, *picture start code*), ni a ninguna información de mejora suplementaria (PEI/PSUPP). Deberán incluirse todos los demás bits hasta los de las capas de GOB o de ranura, sujetos a las limitaciones de W.4.

### **W.6.3.9 Repetición del encabezamiento de la imagen siguiente, referencia temporal fiable**

En este mensaje se repite el encabezamiento de imagen de la siguiente imagen a transmitir. Los bits repetidos no incluyen a los dos primeros bytes del código de comienzo de imagen (PSC), ni a ninguna información de mejora suplementaria (PEI/PSUPP). Deberán incluirse todos los demás bits hasta los de las capas de GOB o de ranura, sujetos a las limitaciones de W.4.

### **W.6.3.10 Repetición del encabezamiento de la imagen siguiente, referencia temporal no fiable**

En este mensaje se repite el encabezamiento de imagen de la siguiente imagen a transmitir. Los bits repetidos no incluyen a los tres primeros bytes del encabezamiento de imagen, ni a ninguna información de mejora suplementaria (PEI/PSUPP). Deberán incluirse todos los demás bits hasta los de las capas de GOB o de ranura, sujetos a las limitaciones de W.4. Cualquiera de los bits de TR o ETR del encabezamiento de imagen repetido, no tienen porque ser necesariamente los mismos que los bits correspondientes del encabezamiento de la siguiente imagen.

### **W.6.3.11 Indicaciones de campo entrelazado**

En el caso de indicaciones de campo entrelazado, el mensaje consiste en una indicación de la codificación del campo entrelazado. Esta indicación no afecta al proceso de decodificación. Sin embargo, indica que la imagen actual no ha sido barrida como una imagen de barrido progresivo. En otras palabras, indica que la imagen codificada actual contiene sólo la mitad de las líneas que la imagen fuente con resolución completa. En las indicaciones de campo entrelazado DSIZE será 1, CONT será 0, y EBIT será 0. En el caso de codificación de campo entrelazado, cada incremento de la referencia temporal denota el tiempo entre el muestreo de los campos alternados de media imagen, en lugar del tiempo entre dos imágenes completas. En el caso de una indicación de campo entrelazado superior, la imagen actual contiene a las líneas primera (la superior), tercera, quinta, etc. de la imagen completa. En el caso de una indicación de campo entrelazado inferior, la imagen actual contiene a las líneas segunda, cuarta, sexta, etc. de la imagen completa. Cuando se envían indicaciones de campo entrelazado, el codificador cumplirá las siguientes normas:

- 1) El codificador utilizará una frecuencia de reloj de imagen (frecuencia de reloj de imagen personalizada, en caso necesario) tal que, cada nuevo campo del vídeo fuente original se corresponda con un incremento 1 de la referencia temporal.
- 2) El codificador utilizará un tamaño de imagen (tamaño de imagen personalizado, en caso necesario) tal que, las dimensiones de la imagen se correspondan con las de un único campo.
- 3) El codificador utilizará una relación de aspecto de píxel (relación de aspecto de píxel personalizada, en caso necesario) tal que, la relación de aspecto de la altura total de la imagen se corresponda con la relación de aspecto de imagen, derivada de la relación de aspecto de píxel del campo único representado por la imagen actual codificada.

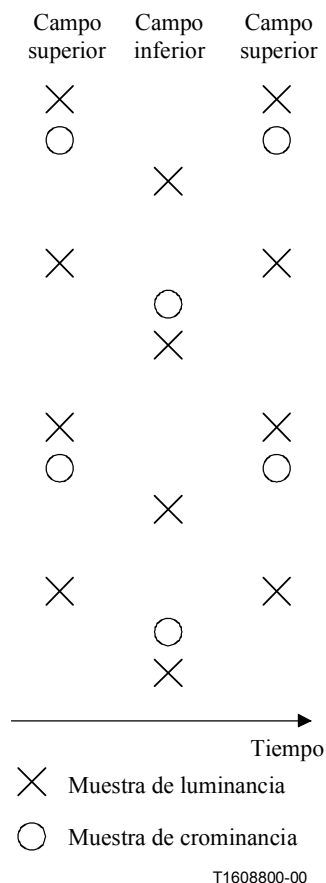
Originalmente se introdujo el barrido entrelazado de campos como una técnica de compresión de vídeo analógico. Aunque el barrido progresivo de imagen se considera generalmente superior para la compresión y presentación digitales, el uso del barrido entrelazado de campos ha persistido en

muchos diseños de cámaras y visualizadores. La codificación de campo entrelazado (que puede implementarse con un retardo inferior que, tanto la imagen completa entrelazada, o que la codificación de la imagen con barrido progresivo a velocidad mitad que la del campo entrelazado) está por tanto soportada por las indicaciones dadas aquí.

Un codificador no enviará indicaciones de campo entrelazado a no ser que se haya establecido previamente, por medios externos (por ejemplo, UIT-T H.245) la capacidad del decodificador para recibir y procesar adecuadamente este tipo de imágenes basadas en el campo. El fallo en el establecimiento de esta capacidad del decodificador puede producir un molesto temblor vertical de pequeña amplitud en la imagen decodificada, recibida y presentada por un decodificador.

Por ejemplo, un codificador puede utilizar la codificación de campo entrelazado aplicando el modo selección de la imagen de referencia (especificado en el anexo N) o el modo selección mejorada de imagen de referencia (especificado en el anexo U) para permitir el direccionamiento de más de un campo previo. Para una codificación de campo entrelazado "525/60" de una imagen con relación de aspecto 4:3, con 704 muestras de luminancia codificadas por línea y 240 líneas de luminancia codificadas por campo, el codificador utilizará un tamaño de imagen personalizado que tenga una anchura de imagen de 704 y una altura de imagen de 240, una relación de aspecto de píxel personalizada de 5:11, y una frecuencia de reloj de imagen personalizada especificada con un código de conversión de reloj "1" y un divisor de reloj de 30. Para una codificación de campo entrelazado "625/50" de una imagen con relación de aspecto 4:3, con 704 muestras de luminancia codificadas por línea y 288 líneas de luminancia codificadas por campo, el codificador utilizará un tamaño de imagen personalizado que tenga una anchura de imagen de 704 y una altura de imagen de 288, una relación de aspecto de píxel personalizada de 6:11, y una frecuencia de reloj de imagen personalizada especificada con un código de conversión de reloj "0" y un divisor de reloj de 36.

Las posiciones verticales de muestreo de las muestras de crominancia en la codificación de campo entrelazado de una imagen de campo superior, se especifican como si estuviesen desplazadas hacia arriba en 1/4 de la altura de las muestras de luminancia con referencia a la rejilla de muestreo del campo, de forma que dichas muestras queden alineadas verticalmente con la posición normal relativa de la rejilla de muestreo de la imagen completa. Las posiciones verticales de muestreo de las muestras de crominancia en la codificación de campo entrelazado de una imagen de campo inferior, se especifican como si estuviesen desplazadas hacia abajo en 1/4 de la altura de las muestras de luminancia con referencia a la rejilla de muestreo del campo, de forma que dichas muestras queden alineadas verticalmente con la posición normal en relación con la rejilla de muestreo de la imagen completa. Las posiciones de muestreo horizontal de las muestras de crominancia se especifican como si no estuviesen afectadas por la aplicación de la codificación de campo entrelazado. En la figura W.2 se muestran las posiciones verticales de muestreo con sus correspondientes posiciones de muestreo en función del tiempo.



**Figura W.2/H.263 – Alineación vertical y temporal de las muestras de crominancia en la codificación de campo entrelazado**

### W.6.3.12 Número de imagen

No se hará uso de este mensaje si está en uso el anexo U. El mensaje contiene dos bytes de datos que albergan a un número de imagen de 10 bits. Por tanto, DSIZE será 3, CONT será 0, y EBIT será 6. El número de imagen se incrementará en 1, en una operación de módulo 10, cada vez que se codifique y transmita una imagen I o P, o una trama PB o PB mejorada. Para imágenes EI y EP, el número de imagen se incrementará para cada imagen EI o EP que esté dentro de la misma capa de mejora de escalabilidad. Para las imágenes B, se incrementará el número de imagen en relación con el valor que tiene en la imagen no B más reciente en la capa de referencia de la imagen B que precede a la imagen B en el orden del tren de bits (una imagen que es temporalmente subsiguiente a la imagen B). Si imágenes adyacentes en la misma capa de mejora tienen la misma referencia temporal, y si está en uso el modo selección de imagen de referencia (véase el anexo N), el decodificador deberá considerar esta situación como una indicación de que se han enviado copias redundantes con aproximadamente el mismo contenido de escena de la imagen, y que todas estas imágenes compartirán el mismo número de imagen. Si la diferencia (módulo 1024) de los números de imagen de dos imágenes no B, recibidas consecutivamente en la misma capa de mejora, no es 1, y si las imágenes no representan aproximadamente el mismo contenido de escena de la imagen, como se ha descrito antes, el decodificador inferirá una pérdida de imágenes o una corrupción de los datos.

### W.6.3.13 Imágenes de referencia de reserva

Los codificadores pueden utilizar este mensaje para instruir a los decodificadores acerca de qué imágenes se parecen lo suficiente a la imagen actual de referencia para la compensación de movimiento, de manera que una de ellas se pueda utilizar como imagen de referencia si se perdiese en la transmisión la imagen de referencia actual. Si un decodificador no dispone de una imagen de

referencia actual, pero puede acceder a una imagen de referencia de reserva, no pedirá una actualización de imagen INTRA. Depende de los codificadores la elección de las imágenes de referencia de reserva, en caso de elegir alguna. Los bytes de los datos del mensaje contienen los números de imagen de las imágenes de referencia de reserva, en orden de preferencia (apareciendo como primera la de mayor preferencia). Los números de imagen se refieren a los valores que se transmiten de acuerdo con el anexo U o W.6.3.12. Se puede utilizar este mensaje para los tipos de imágenes P, B, PB, PB mejorada y EP. Sin embargo, si se está utilizando el anexo N, o el anexo U, y si la imagen está asociada con varias imágenes de referencia, no se utilizará este mensaje. En el caso de imágenes EP, sólo se utilizará el mensaje en la predicción hacia adelante, mientras que la predicción hacia arriba se hace siempre a partir de la imagen de la capa de referencia que corresponde temporalmente. Para los tipos de imágenes B, PB, y PB mejorada, especifica una imagen para ser utilizada como referencia de predicción hacia adelante. Este mensaje no se utilizará si la imagen es del tipo I o EI.

## **SERIES DE RECOMENDACIONES DEL UIT-T**

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
<b>Serie H</b>	<b>Sistemas audiovisuales y multimedios</b>
Serie I	Red digital de servicios integrados
Serie J	Transmisiones de señales radiofónicas, de televisión y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsimil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación