



(19)



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 353 007**

(51) Int. Cl.:
H04N 7/24 (2006.01)

(12)

TRADUCCIÓN DE PATENTE EUROPEA

T3

(96) Número de solicitud europea: **03796040 .8**

(96) Fecha de presentación : **21.11.2003**

(97) Número de publicación de la solicitud: **1568231**

(97) Fecha de publicación de la solicitud: **31.08.2005**

(54) Título: **Método para grabación de datos, método para recuperar conjuntos de datos, archivo de datos, estructura de datos y soporte de grabación.**

(30) Prioridad: **06.12.2002 EP 02293016**

(45) Fecha de publicación de la mención BOPI:
24.02.2011

(45) Fecha de la publicación del folleto de la patente:
24.02.2011

(73) Titular/es: **THOMSON LICENSING**
1-5, rue Jeanne d'Arc
92130 Issy-les-Moulineaux, FR

(72) Inventor/es: **Edge, Robert, C. y**
Damstra, Nicolaas, Johannes

(74) Agente: **Arpe Fernández, Manuel**

ES 2 353 007 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

[0001] La solicitud de patente europea EP 0794667 A2 se refiere a un método de procesamiento de video digital que comprende disponer un paquete de entrada en el que están grabadas posiciones de paquetes previos.

[0002] La invención tiene como objetivo a una estructura de datos de fácil acceso a datos encapsulados (sean estos video, audio y/o metadatos) tanto en dirección hacia delante o como hacia atrás.

[0003] Con este fin, la invención propone un método para grabar de datos encapsulados que pueden ser recuperados en un orden opuesto al orden de grabación, con sucesivas etapas de:

- grabar un elemento KLV que comprende:
- una primera clave indicando el tipo de datos encapsulados;
- un primer indicador de longitud sobre la longitud de los datos encapsulados y
- los datos encapsulados;

comprendiendo además dicho método:

- grabar, inmediatamente después del elemento KLV, como un puntero relativo al comienzo del contenedor de datos precedente, un elemento KLV de retro-puntero que tiene una longitud de retro-puntero y que comprende:

- una segunda clave indicando otros datos encapsulados de tipo de retro-puntero;

- un segundo indicador de longitud sobre la longitud de los otros datos encapsulados;

- como los otros datos encapsulados, un valor de longitud de dicho elemento KLV o la longitud acumulada del valor de longitud de dicho elemento KLV y la longitud del elemento de retro-puntero;

- repetir las etapas de grabación de un elemento KLV y grabación de un elemento KLV de retro-puntero.

[0004] Preferiblemente, el método también tiene una o ambas de las etapas siguientes:

- 5 - grabación del segundo indicador de longitud;
 - grabación de la segunda clave que indica otros datos encapsulados de tipo de retro-puntero.

[0005] Este método de grabación permite leer fácilmente hacia atrás, los conjuntos de datos con el siguiente método
10 concurrentemente propuesto por la invención: un método para recuperar un elemento KLV que tiene en un soporte una longitud de elemento KLV en un orden contrario al orden de grabación a partir de un elemento KLV accedido actualmente, que comprende etapas de:

- 15 - acceder a un elemento retro-puntero que tiene una longitud de retro-puntero y que precede al mencionado elemento KLV accedido;
 - acceder a una clave que indica que los datos encapsulados son de tipo retro-puntero;
20 - lectura, como datos encapsulados, de un valor que puede ser bien la longitud de elemento KLV o la longitud acumulada de la longitud del elemento KLV y la longitud del elemento retro-puntero;
 - acceder al mencionado elemento KLV usando dicho valor.

25 **[0006]** De manera preferida, el método además comprende la lectura de un indicador de longitud sobre la longitud de los datos encapsulados antes de leer los datos encapsulados.

[0007] La invención por tanto propone un archivo de datos que comprende bloques consecutivos, comprendiendo sucesivamente
30 cada bloque:

- un contenedor de datos que tiene una longitud de contenedor;
 - una clave que indica el tipo de datos encapsulados;

- un indicador de longitud sobre la longitud de los datos encapsulados y

- los datos encapsulados;

- un elemento KLV retro-puntero que tiene una longitud de retro-puntero y que comprende

- una clave de retro-puntero;

- un indicador de longitud;

- como los otros datos encapsulados, un valor que puede ser o la longitud del elemento KLV o la longitud acumulada de dicha longitud del elemento KLV, y la mencionada longitud del retro-puntero; y

- un soporte portador, como un archivo de datos.

La invención además proporciona una estructura de datos que tiene sucesivamente:

- un elemento KLV con una longitud de elemento KLV (l_e ; l_m) dada y comprendiendo:

- una clave que indica el tipo de datos encapsulados;

- un indicador de longitud sobre la longitud de los datos encapsulados y

- los datos encapsulados; y

- un elemento KLV retro-puntero con una longitud de retro-puntero que comprende;

- una clave del retro-puntero;

- un indicador de longitud;

- como otros datos encapsulados un valor que es la longitud de elemento KLV o la longitud acumulada de la longitud de elemento KLV y la longitud de retro-puntero.

Preferiblemente, la estructura de datos tiene uno o ambos de los siguientes campos:

- los otros indicadores de longitud;

- la clave de retro-puntero.

[0008] Gracias al elemento KLV retro-puntero situado inmediatamente después del elemento KLV e indicativo de su

longitud, el salto desde el final del elemento KLV (por ejemplo, el comienzo del siguiente elemento KLV) al comienzo del elemento KLV es rápido y sencillo, lo cual facilita la lectura hacia atrás del archivo.

5 **[0009]** La invención también propone codificaciones KLVL y KLVLK.

10 **[0010]** Otras características de la invención aparecerán a la luz de la siguiente descripción de las realizaciones preferentes de la invención hecha con referencia a las figuras detalladas donde:

 - La figura 1 representa una primera realización de la estructura de datos de acuerdo a la invención;

 - La figura 2 representa una segunda realización de la estructura de datos de acuerdo a la invención;

15 - La figura 3 representa un ejemplo de la novedosamente propuesta codificación KLVL;

 - La figura 4 representa un ejemplo de la novedosamente propuesta codificación KLVLK.

20 **[0011]** La figura 1 muestra una estructura de datos que puede ser usada particularmente para grabar de datos de extracto donde el número de octetos por contenedor puede variar. Por ejemplo, el extracto es un conjunto de tramas de datos de video comprimido con longitud desigual. En esta aplicación particular la estructura de datos de la figura 1, es
25 utilizada por una grabadora de video cuando graba una secuencia de video. Esta estructura de datos es utilizada de una manera similar por un reproductor de video que reproduce la secuencia de video.

30 **[0012]** Cada trama de la secuencia de video está codificada en KLV; esto es por tanto descrito por un campo clave K_e indicando que esos datos son datos de video, por ejemplo, datos de video codificados en MPEG, un campo longitud L_e ,

indica la longitud de esos datos de video y un campo valor V_e contiene los datos de video (extracto).

[0013] Después de cada trama del conjunto, un elemento KLV retro-puntero es insertado. Este elemento KLV retro-puntero es un puntero relativo al comienzo de las tramas precedentes (esto es, al contenedor de datos precedente). Su función es indicada con su clave K_{bp} y su valor V_{bp} indica la longitud de la trama de video precedente (o hablando de una forma más genérica del contenedor de datos precedentes). Por ejemplo, su valor es la longitud I_e del elemento codificado en KLV representando la trama ($V_{bp}=I_e$). Como posible variación, la longitud total (acumulada) I_t del elemento KLV de extracto y el elemento KLV retro-puntero podrían ser utilizados en su lugar.

[0014] Habitualmente, el campo longitud L_{bp} representa la longitud del campo valor V_{bp} . Una solución práctica es tener una longitud L_{bp} fija para el elemento retro-puntero, pero una longitud variable también es posible como se explica más adelante.

[0015] Cuando se graban 3 tramas F_1 , F_2 y F_3 , un registrador de video que utiliza la estructura de datos de la figura 1 grabará así la siguiente secuencia: $K_e L_1 F_1 K_{bp} L_{bp} I_1 K_e L_2 F_2 K_{bp} L_{bp} I_2 K_e L_3 F_3 K_{bp} L_{bp} I_3$, donde L_i es la longitud de los datos codificados F_i e I_i es la longitud del elemento KLV ($K_e L_i F_i$) que contiene los datos codificados F_i .

[0016] Estos conjuntos de datos pueden ser fácilmente recuperados por un reproductor de video en direcciones hacia delante y hacia atrás como se explica más adelante.

[0017] Cuando el reproductor de video lee los conjuntos en dirección hacia delante (esto es en la misma dirección en la que los conjuntos fueron grabados), la estructura KLV del archivo facilita saltar elementos. Particularmente, los

elementos KLV retro-punteros pueden ser saltados fácilmente usando su campo longitud L_{bp} .

[0018] El reproductor de video puede también leer fácilmente los conjuntos en dirección hacia atrás. Asumiendo que el
 5 elemento KLV actualmente accedido es el que representa la trama F3, el acceso al elemento KLV retro-puntero precedente $K_{bp}L_{bp}I_2$ es inmediato cuando la longitud L_{bp} es fija. Mediante la lectura del valor I_2 del elemento KLV retro-puntero, el reproductor de video puede entonces acceder inmediatamente al
 10 elemento KLV precedente representativo de la trama F2 saltando I_2 octetos hacia atrás.

[0019] El reproductor de video puede o leer el contenido del elemento KLV $K_eL_2F_2$ para decodificar la trama F2 o saltar inmediatamente hacia atrás a la trama F1 con un utilización
 15 similar del elemento retro- puntero $K_{bp}L_{bp}I_1$.

[0020] Gracias a la utilización de los elementos retro-punteros, el salto rápido de trama a trama hacia atrás es posible, con la opción cada vez de decodificar o no la trama. Los elementos retro-punteros son por tanto particularmente
 20 útiles para permitir un modo rápido hacia atrás que muestre sólo alguna de las imágenes encontradas.

[0021] Debe tenerse en cuenta que los elementos retro-punteros proporcionan esta ventaja incluso cuando su longitud L_{bp} no es fija. En este caso, el elemento KLV retro-puntero
 25 precedente puede ser accedido a través de búsquedas hacia atrás de la clave K_{bp} . Esto es, por supuesto, mucho más rápido que buscar hacia atrás la clave extracto precedente K_e cuando el elemento KLV retro-puntero es mucho más corto que el elemento KLV trama (extracto).

[0022] Es también posible codificar el retro-puntero con codificación KLVL o codificación KLVLK como se describe más adelante. Como explicábamos, estos tipos de codificación

permiten leer el elemento en dirección hacia delante o hacia atrás.

[0023] En el ejemplo antes mencionado, la utilización del elemento retro-puntero ha sido descrito en asociación con los datos de extracto. Por supuesto, tal elemento retro-puntero puede ser usado de modo similar con metadatos o cualquier otro tipo de datos.

[0024] La figura 2 muestra un ejemplo de utilización de un elemento retro-puntero asociado con metadatos.

[0025] Un conjunto de K metadatos $V_1, \dots, V_k$ son codificados en KLV y combinados como el campo valor V_m del elemento KLV generalmente denominado "elemento KLV conjunto de metadatos". El elemento conjunto de metadatos tiene un campo clave específico K_m y un campo longitud que indica la longitud del campo valor V_m .

[0026] Un elemento KLV retro-puntero $K_{bp}L_{bp}V_{bp}$ sigue inmediatamente al elemento KLV conjunto de metadatos $K_mL_mV_m$. La clave K_{bp} indica que el presente elemento KLV es un puntero relativo al comienzo del elemento KLV precedente (o contenedor de datos), en este caso el comienzo del conjunto de metadatos. El campo longitud L_{bp} indica la longitud del campo valor V_{bp} la cual es igualmente indicativa de la longitud I_m del elemento KLV precedente, en este caso el conjunto de metadatos.

[0027] Cuando los datos son grabados por un registrador como se representa de izquierda a derecha (dirección hacia delante) en la figura 2, estos pueden ser fácilmente accesibles (y posiblemente leídos) por un reproductor en ambas direcciones hacia delante y hacia atrás como se explica ahora.

[0028] En la dirección hacia delante, K_m es el primero accedido, informando el reproductor que el conjunto de datos accedido es un conjunto de metadatos. El reproductor puede o

utilizar el siguiente campo longitud L_m para saltar el conjunto de metadatos, o leer el campo valor V_m para realmente acceder a uno o varios metadatos.

5 **[0029]** Si el conjunto de metadatos se salta, el reproductor encuentra el elemento KLV retro-puntero, identifica éste como tal gracias a su clave K_{bp} y lo salta gracias al campo longitud L_{bp} .

10 **[0030]** Por otro lado, si el campo valor V_m es realmente leído, el reproductor puede acceder fácilmente a los metadatos $V_1, \dots, V_k$, uno a uno en dirección hacia delante gracias a su estructura KLV. Entonces, el reproductor accede al elemento KLV retro-puntero y salta éste como se describió anteriormente.

15 **[0031]** En dirección hacia atrás, el reproductor tiene que identificar que datos componen el elemento retro-puntero. Como se describió anteriormente, una solución práctica es decidir de antemano que los elementos retro-punteros tengan una longitud fija. El reproductor puede en este caso llevar esta longitud fija en la dirección hacia atrás y así acceder
20 a la clave K_{bp} .

[0032] Incluso en casos donde la longitud del elemento retro-puntero puede variar, el reproductor puede fácilmente acceder a la clave K_{bp} buscándola hacia atrás. Es por supuesto más rápido, más fácil y mucho más fiable buscar hacia atrás una
25 clave dada (K_{bp}) que intentar leer el conjunto de metadatos hacia atrás (lo cual consistiría en buscar cualquier posible clave entre una gran cantidad de datos, sabiendo que nuevos metadatos con sus correspondientes nuevas claves pueden ser introducidos).

30 **[0033]** Como se ha explicado previamente, el retro-puntero puede ser también codificado en KLVL o en KLVLK.

[0034] Cuando el elemento retro-puntero es identificado, el campo valor V_{bp} es leído con el valor I_m y el reproductor

consecuentemente salta L_m octetos hacia atrás accediendo por lo tanto inmediatamente al elemento KLV conjunto de metadatos $K_m L_m V_m$.

- [0035]** El reproductor puede entonces o leer el elemento conjunto de metadatos bien adquiriendo algunos de los metadatos (en dirección hacia delante, como se explicó anteriormente) o ignorar el conjunto de metadatos y continuar la lectura hacia atrás accediendo al nuevo elemento KLV retro-puntero como $K_{bp} L_{bp} I_3$ en la figura 1.
- [0036]** Como una posible alternativa a la figura 2, el elemento retro-puntero $K_{bp} L_{bp} V_{bp}$ podrían ser considerado como parte del conjunto de metadatos $k_m L_m V_m$. Los principios de funcionamiento continuarían siendo los mismos, tanto en dirección hacia delante como hacia atrás. En este caso, sin embargo, si el campo valor V_{bp} aún representa la longitud de los elementos de todos los conjuntos de metadatos, el reproductor tiene que deducir por si mismo la longitud del elemento retro-puntero desde el valor V_{bp} antes de saltar hacia atrás. Otras variaciones son factibles mientras el campo valor V_{bp} sea indicativo de la longitud en octetos entre el comienzo del contenedor de datos (aquí clave K_m) y el elemento retro-puntero (clave K_{bp}).

- [0037]** El elemento retro-puntero ha sido diseñado en el ejemplo previo para saltar hacia atrás sobre el conjunto total de metadatos pero naturalmente es posible proveer elementos retro-punteros para saltar hacia atrás sobre subconjuntos de metadatos dentro del conjunto de metadatos.

- [0038]** La figura 1 y la figura 2 proponen el uso del elemento retro-puntero para simplificar enormemente la lectura hacia atrás del conjunto de datos, particularmente datos codificados en KLV en un archivo MXF.

[0039] Ventajosamente, el encabezamiento de este archivo tiene una bandera (por ejemplo un bit) que indica donde o no

son usados los elementos retro-punteros en el archivo. Otra información, como por ejemplo la utilización de codificaciones KLVL o KLVK o la longitud fija del elemento retro-puntero (si existe) podría incluirse en el
5 encabezamiento de archivo. El encabezamiento puede adjuntarse al archivo como metadatos.

[0040] Lo siguiente es una descripción de codificaciones KLVK y KLVK que son propuestas por la invención. Aunque éstas son preferiblemente aplicadas sólo a los elementos retro-punteros
10 como se describió en la descripción anterior, se pueden aplicar a cualquier clase de datos.

[0041] La figura 3 representa una codificación KLVL de varios elementos metadatos dando un ejemplo de codificación KLVL. Cada elemento representa una información dada (o parámetro)
15 teniendo un valor Vi. Para ser transportado y recuperado, cada conjunto de datos es encapsulado como se describe más adelante.

[0042] Una clave Ki (por ejemplo codificada en 16 octetos) es determinada para indicar que clase de información es
20 representada por cada elemento. La longitud (por ejemplo en octetos) de los datos representativos de la información o valor Vi, se determina también como indicador de longitud Li (por ejemplo codificado en 4 octetos).

[0043] Al grabar los metadatos, el registrador escribe para
25 cada elemento la siguiente secuencia (en el orden indicado): Ki Li Vi Li. Con el ejemplo de longitud-octeto de antes, la secuencia Ki Li Vi Li es (24 + Li) octetos de longitud.

[0044] En el ejemplo representado en la figura 3, las 3
30 siguientes piezas (elementos) de información de metadatos son consideradas:

- título de la grabación (clave K1) codificado en 256 octetos (L1 significa 256) definido en datos V1;

- técnica de compresión de video (clave K2) codificado en 2 octetos (L2 significa 2) definido en datos V2;

- duración en segundos de la grabación (clave K3) codificada en 4 octetos (L3 significa 4) definido en datos V3.

[0045] El registrador de video grabará estos metadatos de acuerdo a la siguiente secuencia: K1 L1 V1 L1 K2 L2 V2 L2 K3 L3 V3 L3 como se ilustra en la figura 3.

[0046] Los metadatos son por supuesto escritos incluidos en una estructura mayor como la provista por el formato de intercambio de material, por ejemplo como se describe en la solicitud de patente WO 02/21845. Sin embargo, la codificación KLV se sustituye por codificación KLV L al menos para algunos de los elementos.

[0047] Cuando otra máquina lee el archivo, los metadatos pueden ser accedidos.

[0048] Si el archivo de video-audio se lee en dirección hacia delante (esto es en la misma dirección en la que fue grabado), la clave K1 es la primera leída, luego se lee la longitud L1, permitiendo determinar la longitud en octetos de los datos V1 y así leer el valor V1. Los 4 octetos (L1) siguientes a V1 son ignorados y puede accederse a K2. Las demás recuperaciones de metadatos pueden hacerse de manera similar.

[0049] Si el archivo de video-audio se lee en dirección hacia atrás, la longitud L3 es la primera a la que se accede. Esta da inmediatamente la longitud de octetos a leer como valor V3. Una vez el valor V3 es leído, los siguientes 4 octetos en dirección hacia atrás (también representados como L3) son ignorados y entonces se accede a K3. V3 y K3 son así inmediatamente determinados cuando se lee hacia atrás sin necesidad de una tabla de índice.

[0050] Por supuesto, las demás recuperaciones de metadatos en dirección hacia atrás son llevadas a cabo con la misma simple combinación: L2 es leído, dando de este modo la posibilidad de leer V2; la aparición de L2 entre V2 y K2 es ignorada y se lee K2. Finalmente, L1 es leída, dando de ese modo inmediatamente acceso al valor L1 y saltando los 4 octetos de L1 a la clave K1.

[0051] Como se muestra con este ejemplo, la codificación KLVL permite la fácil recuperación de datos encapsulados (v) cuando la secuencia KLVL es leída o en dirección hacia delante o en dirección hacia atrás.

[0052] Como una posible variación, cuando se lee por segunda vez el campo indicador de longitud L (o en dirección hacia delante o hacia atrás), ésta puede ser comparada con la primera lectura del indicador de longitud (en lugar de simplemente ignorar la segunda lectura del indicador de longitud como se describió anteriormente). Esto permite comprobar si el archivo tiene el formato esperado y sin errores, y por lo tanto si el algoritmo propuesto de salto hacia atrás de los datos, está todavía sincronizado con la codificación KLVL del archivo.

[0053] De manera equiparable, la codificación KLVLK se obtiene con la grabación de un elemento Vi de acuerdo a la siguiente secuencia: Ki Li Vi Li Ki.

[0054] La figura 4 representa la codificación KLVLK de varios elementos metadatos dados, como un ejemplo de codificación KLVLK. Cada elemento representa una información dada (o parámetro) que tiene un valor Vi. Para ser transportado y recuperado, cada conjunto de datos es encapsulado como se describe a continuación.

[0055] Una clave Ki (por ejemplo codificada en 16 octetos) se determina para indicar que clase de información es representada por el elemento. La longitud (por ejemplo en

octetos) de datos representativa de la información o el valor V_i también es determinada como indicador de longitud L_i (por ejemplo codificado en 4 octetos).

[0056] Al grabar los metadatos, el registrador escribe para
5 cada elemento la siguiente secuencia (en el orden indicado):
 $K_i L_i V_i L_i K_i$. Con el ejemplo de longitud en octetos
anterior, la secuencia $K_i L_i V_i L_i$, es $(40 + L_i)$ octetos de
largo.

[0057] En el ejemplo descrito en la figura 4, las dos
10 siguientes piezas (elementos) de información de metadatos son
consideradas:

- título de la grabación (clave K_1) codificado en 256 octetos (L_1 significa 256) definido en datos V_1 ;

- técnica de compresión de video (clave K_2) codificado
15 en 2 octetos (L_2 significa 2) definido en datos V_2 .

[0058] El registrador de video grabará estos metadatos de acuerdo a la siguiente secuencia: $K_1 L_1 V_1 L_1 K_1 K_2 L_2 V_2 L_2 K_2$ como se ilustra en la figura 4.

[0059] Cuando otra máquina lee el archivo, puede accederse a
20 los metadatos.

[0060] Si el archivo de video-audio se lee en la dirección hacia delante (esto es en la misma dirección en la que fue grabado), la clave K_1 es la primera leída, luego se lee la longitud L_1 permitiendo determinar la longitud en octetos de
25 los datos V_1 y así leer el valor V_2 . Los 20 octetos ($L_1 K_1$) siguientes a V_1 son ignorados y K_2 puede ser accesible. Las demás recuperaciones de metadatos pueden hacerse de manera similar.

[0061] Si el archivo de video-audio se lee en dirección hacia
30 atrás, la clave K_2 es la primera accesible entonces la longitud L_2 es leída permitiendo la longitud en octetos de los datos V_2 y por tanto leer el valor V_2 . Los restantes 20 octetos ($L_2 K_2$) pueden ser ignorados.

[0062] Las demás recuperaciones de metadatos en dirección hacia atrás son llevadas a cabo con la misma combinación. De hecho, como la grabación es simétrica con la codificación KLVLK, la lectura hacia atrás utiliza el mismo algoritmo que
5 la lectura hacia delante y es por tanto tan simple como aquella.

[0063] Como una posible variación, cuando se lee por segunda vez los campos indicador de longitud L y de clave K (tanto en dirección hacia delante como hacia atrás), ésta puede ser
10 comparada con los primeros indicadores de longitud y clave leídos (en lugar de simplemente ignorarlos como se describió anteriormente). Esto permite comprobar si el archivo tiene el formato esperado y sin errores, y por tanto si el algoritmo propuesto está todavía sincronizado con la codificación KLVL
15 del archivo.

[0064] La invención no está limitada a las realizaciones arriba descritas. Se pueden hacer variaciones a estas realizaciones sin alejarse del ámbito de la invención.

REIVINDICACIONES

1. Método para grabar datos encapsulados (V_e, V_m) que pueden ser recuperados en orden opuesto al de grabación,
 5 incluyendo dicho método las etapas sucesivas de:

- a) grabar, como contenedor de datos, un elemento codificado en KLV que comprende
 - una primera clave (K_e, K_m) que indica un tipo de datos encapsulados;
 - 10 - un primer indicador de longitud (L_e, L_m) sobre la longitud de los datos encapsulados;
 - los datos encapsulados (V_e, V_m);

caracterizado por:

b) Grabar, después del elemento codificado en KLV, como
 15 un puntero relativo al comienzo del contenedor de datos precedente, un elemento codificado en KLV de retro-puntero que tiene una longitud de retro-puntero y que comprende:

- una segunda clave (K_{bp}) que indica otros datos encapsulados que son de tipo retro-puntero;
- 20 - un segundo indicador de longitud (L_{bp}) sobre la longitud de los otros datos encapsulados;
- como los otros datos encapsulados (V_{bp}) un valor de longitud (I_e) del elemento codificado en KLV o un valor de longitud acumulada (I_t, I_m) de la mencionada longitud del
 25 elemento codificado en KLV y la longitud de retro-puntero; y
- repetir las etapas a) y b).

2. Método de acuerdo a la reivindicación 1, donde después del elemento codificado en KLV de retro-puntero, se graba de nuevo el segundo indicador de longitud.

30 3. Método de acuerdo a la reivindicación 2, donde después del segundo indicador de longitud, se graba de nuevo la segunda clave.

4. Método para recuperar un elemento codificado en KLV ($K_e, L_e, V_e; K_m, L_m, V_m$) que ha sido grabado de acuerdo a la reivindicación 1, presentando en un soporte, una longitud de elemento KLV en orden opuesto al orden de grabación a partir de un elemento KLV actualmente accedido, incluyendo dicho método las etapas de:

- acceder a la segunda clave (k_{bp}) indicando que los datos encapsulados son de tipo retro-puntero y donde los datos encapsulados tienen una longitud fija;
- 10 - lectura de los otros datos encapsulados (V_{bp}) que indican la longitud acumulada (I_t) del mencionado elemento KLV y de la mencionada longitud de retro-puntero;
- acceder al elemento KLV saltando un cierto número de octetos más hacia atrás, donde el número de octetos es igual al mencionado valor de longitud acumulada (4).
- 15

5. Método para recuperar un elemento codificado en KLV ($K_e, L_e, V_e; K_m, L_m, V_m$) que ha sido grabado de acuerdo a la reivindicación 1, presentando en un soporte una longitud de elemento KLV, en orden opuesto al orden de grabación a partir de un elemento KLV actualmente accedido, incluyendo dicho método etapas de:

- acceder a la segunda clave (k_{bp}) indicando que los datos encapsulados son de tipo retro-puntero y donde los datos encapsulados tienen una longitud no fija;
- 25 - lectura de un segundo indicador de longitud (L_{bp}) sobre la longitud de los datos encapsulados antes de leer el valor (V_{bp});
- lectura de los otros datos encapsulados (V_{bp}) que indican la longitud acumulada (I_t) del mencionado elemento KLV y de la mencionada longitud de retro-puntero;
- 30 - acceder al elemento KLV saltando un cierto número de octetos más hacia atrás, donde el número de octetos es igual al mencionado valor de longitud acumulada (4).

6. Archivo de datos comprende bloques sucesivos, comprendiendo sucesivamente cada bloque:

- un elemento codificado en KLV, como contenedor de datos, que comprende:

5 - una primera clave (K_e , K_m) que indica un tipo de datos encapsulados;

- un primer indicador de longitud (L_e , L_m) sobre la longitud de los datos encapsulados;

- los datos encapsulados (V_e, V_m);

10 **caracterizado por:**

- después del elemento codificado en KLV, como puntero relativo al comienzo del contenedor de datos precedente, un elemento codificado en KLV de retro-puntero que tiene una longitud de retro-puntero y que comprende:

15 - una segunda clave (K_{bp}) que indica otros datos encapsulados son de tipo retro-puntero;

- un segundo indicador de longitud (L_{bp}) sobre la longitud de los otros datos encapsulados;

20 - como los otros datos encapsulados (V_{bp}), un valor de longitud (I_e) del elemento codificado en KLV o un valor de longitud acumulada (I_t , I_m) de la mencionada longitud del elemento codificado en KLV y de la longitud del retro-puntero;

25 7. Archivo de datos de acuerdo a la reivindicación 6, donde al mencionado elemento codificado en KLV de retro-puntero se adjunta el segundo indicador de longitud para representar un formato codificado en KLVL.

30 8. Archivo de datos de acuerdo a la reivindicación 6, donde al mencionado elemento codificado en KLV de retro-puntero KLV, se adjuntan el segundo indicador de longitud y la segunda clave para representar un formato codificado en KLVLK.

9. Soporte que contiene un archivo de datos de acuerdo a una de las reivindicaciones de 6 a 8.

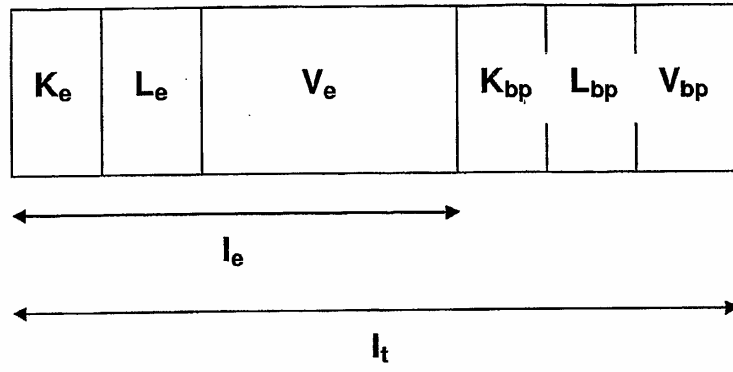


Fig. 1

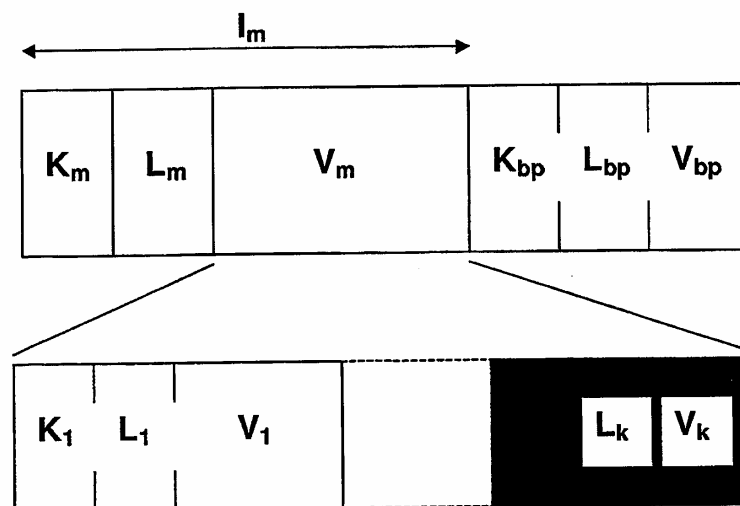


Fig. 2

20

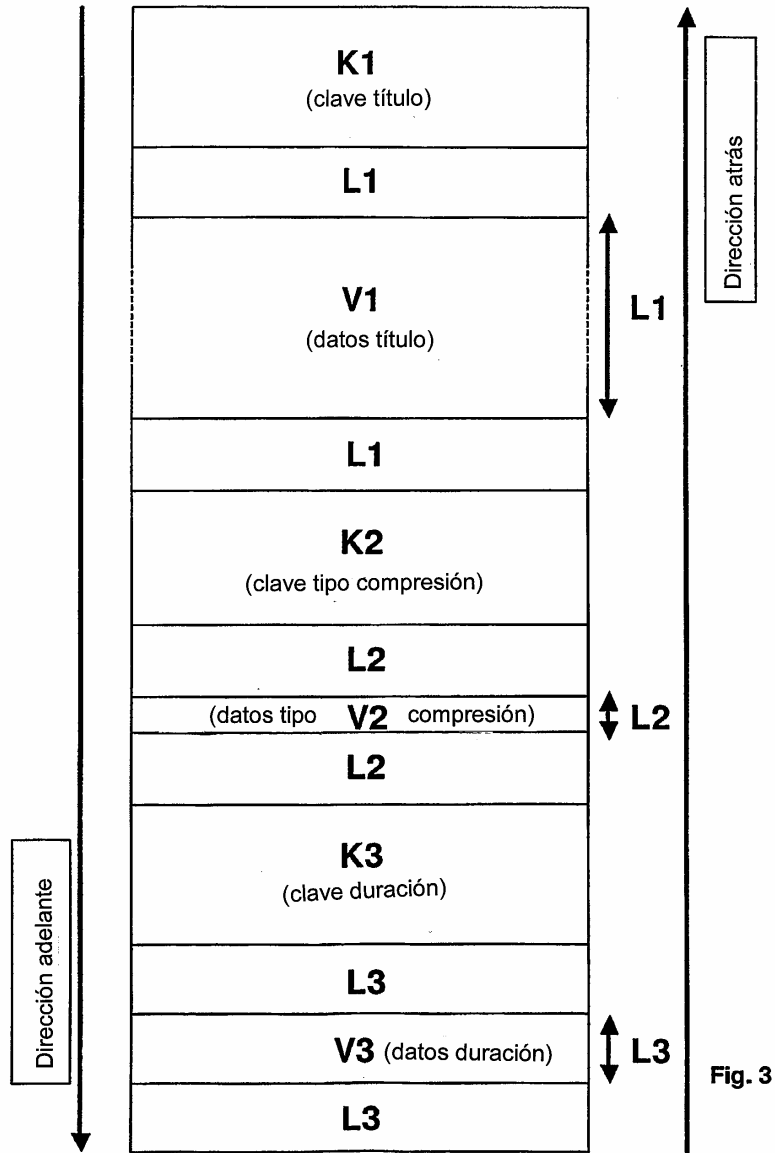


Fig. 3

21

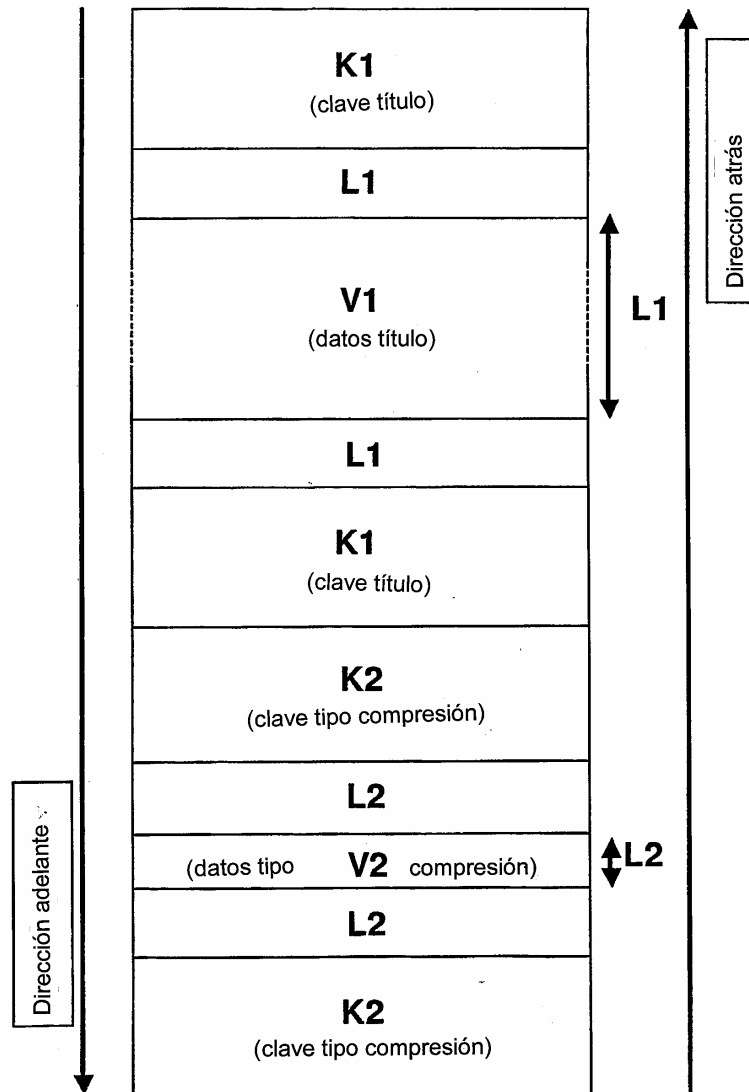


Fig. 4

REFERENCIAS CITADAS EN LA DESCRIPCIÓN

La lista de referencias citada por el solicitante lo es solamente para utilidad del lector, no formando parte de los documentos de patente europeos. Aún cuando las referencias han sido cuidadosamente recopiladas, no pueden excluirse errores u omisiones y la OEP rechaza toda responsabilidad a este respecto.

Documentos de patente citados en la descripción

• EP 0794667 A2 [0001]

• WO 0221845 A [0046]