

# Capítulo 5

## Postprocesado de la información y extracción de *keyframes*

### 5.1 INTRODUCCIÓN

En los anteriores capítulos, hemos descrito el proceso para localizar los cortes y las transiciones graduales en una secuencia. El resultado de esta localización son unos ficheros de texto, que contienen los números de los fotogramas “frontera” entre planos.

El objetivo de este capítulo es combinar esta información de “fronteras” para poder segmentar temporalmente la secuencia en sus planos. También se presentarán algunas aplicaciones de postprocesado de la segmentación, programadas en este proyecto.

La información de la segmentación será utilizada para las siguientes aplicaciones:

- Determinar los *shots* o planos de la secuencia.
- Extracción de fotogramas clave de cada plano.
- Visualización de los fotogramas clave y de los efectos de transición.
- Generación de información textual, describiendo el contenido de la secuencia.

Algunas de estas aplicaciones, necesitarán los resultados de otras (por ejemplo, es evidente que para visualizar los fotogramas clave, previamente han de haber sido extraídos); y estos resultados se guardarán en ficheros de texto, tal y como hacíamos para el análisis temporal. A continuación, desarrollaremos los fundamentos de cada una de estas etapas o fases, y su programación definitiva en el PDIWIN32.

## 5.2 DETERMINAR LOS PLANOS DE UNA SECUENCIA

Ya vimos en los capítulos 3 y 4 los mecanismos necesarios para detectar los cortes, los fundidos y las cortinillas. Una vez detectados estos efectos, la información es almacenada en los siguientes ficheros:

- Cortes  $\rightarrow$  *indice1.txt* y *cuts.txt* .- El primero se trata de un índice, para una secuencia de  $N$  fotogramas tendrá  $N - 1$  elementos. Sus elementos serán 0, excepto en los diferencias que constituyan un corte, que tendrá un 1. El fichero *cuts.txt* contendrá los números de los fotogramas (uno por línea) que constituyen un corte. Podrá haber números consecutivos, y se corresponderán a los cortes con fotogramas intermedios.
- Fundidos  $\rightarrow$  *indice2.txt* y *dissolves.txt* .- El primero se trata de un índice, para una secuencia de  $N$  fotogramas tendrá  $N$  elementos. Sus elementos serán 0, excepto en los fotogramas pertenecientes a un fundido, que tendrá un 2. El fichero *dissolves.txt* contendrá los números de los fotogramas que constituyen un fundido, con la siguiente sintaxis: una línea por fundido, y para cada uno contendrá dos números enteros (primer y último fotograma del fundido) separados por el carácter “ : ”.
- Cortinillas  $\rightarrow$  *indice3.txt* y *wipes.txt* .- El primero se trata de un índice, para una secuencia de  $N$  fotogramas tendrá  $N$  elementos. Sus elementos serán 0, excepto en los fotogramas pertenecientes a una cortinilla horizontal, que tendrá un 3, o en los fotogramas pertenecientes a una cortinilla vertical, que tendrá un 4. El fichero *wipes.txt* contendrá los números de los fotogramas que constituyen una cortinilla, con la siguiente sintaxis: una línea por cortinilla, y para cada una contendrá dos números enteros (primer y último fotograma de la cortinilla) separados por el carácter “ h ”, cuando la cortinilla sea horizontal o por el carácter “ v ”, cuando sea vertical.

Para una determinada secuencia, no es necesario detectar todos los posibles efectos. Si se sabe a priori que, por ejemplo, no tiene ninguna cortinilla, el intentar localizar cortinillas es una pérdida de tiempo. Además, en el momento de segmentarla temporalmente, debemos poder elegir que información del análisis temporal vamos a utilizar.

### 5.2.1 Método de combinación de la información

En el capítulo de conceptos básicos, definimos un plano o *shot* temporal como un conjunto de fotogramas consecutivos, cuyo contenido está relacionado y que habitualmente representa un evento o una secuencia de acciones continuas. También se comentó que

hay dos formas para que un plano pueda realizar una transición al siguiente: transición abrupta o corte y transición gradual (fundidos, efectos de incrustaciones como las cortinillas, etc).

Por tanto, una vez se disponen de los cortes, fundidos y cortinillas, estaremos en situación de poder determinar cada uno de los planos.

Hay dos cuestiones a tener en cuenta y que pueden crear confusión. ¿Qué ocurre con los fotogramas intermedios de un fundido? y, ¿se debe considerar fundido un corte con algún fotograma intermedio?. Para la primera pregunta hemos tomado la siguiente determinación: si tenemos un fundido entre dos planos, el último fotograma del plano inicial y el primero del plano final, junto con los fotogramas intermedios, serán tratados como un plano más. Es decir, en un fundido de dos planos, tras la segmentación tendremos tres planos: plano inicial, plano fundido y plano final. Se podrían haber planteado otras posibles formas de trabajo: incluir los fotogramas intermedios del fundido en el plano inicial, incluirlos en el plano final, incluir la mitad en cada plano, etc. Nos ha parecido que esta es la forma más intuitiva y correcta de realizarlo. La segunda cuestión hace referencia a los cortes con fotogramas intermedios. Por definición, un corte no abrupto, es decir, cuando incluya uno o dos fotogramas intermedios, es un fundido; y como tal deberá tratarse. Aunque conceptualmente no hay ninguna confusión en esto, el problema surge porque los fundidos “cortos” (con solo uno o dos fotogramas intermedios) pueden ser detectados con  $d_c(X, Y)$  y  $d_\rho(X, Y)$  (las métricas de detección de cortes), y por tanto, la información necesaria para segmentar usando estos fundidos estará en los ficheros *indice1.txt* y *cuts.txt*. En conclusión diremos que, para tener en cuenta todos los fundidos a la hora de segmentar, se ha de utilizar la información proveniente de la detección de los cortes y de los fundidos propiamente dicha. Y que aunque seleccionemos segmentar utilizando solo los cortes, si alguno tiene fotogramas intermedios, se obtendrán planos fundido con ellos.

Fijados estos criterios y considerando que vamos a utilizar toda la información disponible, el método de combinación de ésta, consistirá en tener un índice con tantos elementos como fotogramas tenga la secuencia a analizar. Los elementos del índice se inicializarán a 0, y se irán modificando de la siguiente forma:

1. Obtener de *cuts.txt* los información de los cortes (y fundidos “cortos”). Los elementos del índice que correspondan a un corte se pondrán a 1, y los que correspondan a un fundido se pondrán a 2.
2. Obtener de *dissolves.txt* la información de los fundidos. Los elementos del índice que correspondan a un fundido se pondrán a 2.
3. Obtener de *wipes.txt* la información de las cortinillas. Los elementos del índice que correspondan a una cortinilla horizontal se pondrán a 3 y los que correspondan a una vertical se pondrán a 4.

Al concluir, tendremos incorporado al índice toda la información necesaria para la segmentación. Para segmentar recorreremos uno a uno los elementos, mientras sean 0 estaremos en el mismo plano, y cuando encontremos un número diferente indicará un cambio de plano.

En la sección siguiente, se comentará como se ha programado todo esto en el PDI-WIN32.

### 5.2.2 Determinación de los planos con el PDIWin32

Para determinar los planos de una secuencia se ha situado una función, **Determinar Shots**, dentro del “POPUP” **Postprocesado**, tal y como se ve en la figura 5.1.

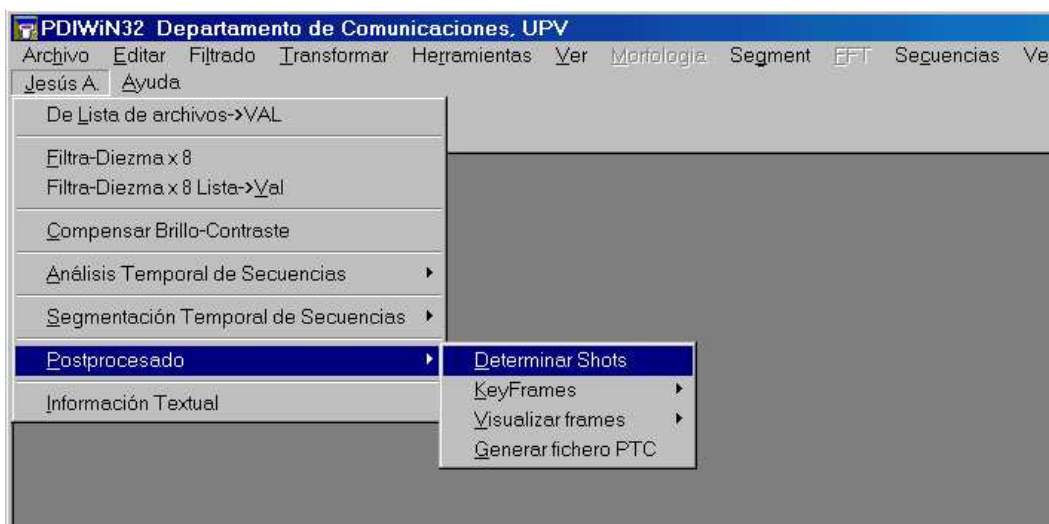


Figura 5.1: Entrada de menú para determinar los planos de una secuencia.

Al igual que ocurre con todas las funciones anteriores que hemos programado en este proyecto, la imagen activa ha de ser una secuencia *VAL*, ya que en caso contrario se nos mostrará un mensaje de error.

Lo primero que hará será mostrarnos un cuadro de diálogo, figura 5.2, para indicarle que información del análisis temporal: cortes, fundidos y cortinillas, va a utilizarse para determinar los planos.

Dependiendo de las opciones elegidas, se buscarán los ficheros que contengan la información necesaria. Si se va a utilizar la información de los cortes, se buscará el fichero *cuts.txt*, y en el caso de no encontrarlo nos mostrará una ventana de error. De igual forma, cuando se quiera utilizar la información de fundidos y/o cortinillas buscará los ficheros *dissolves.txt* y *wipes.txt*, respectivamente. Si no encuentra alguno de los necesarios, nos mostrará una ventana de error similar.

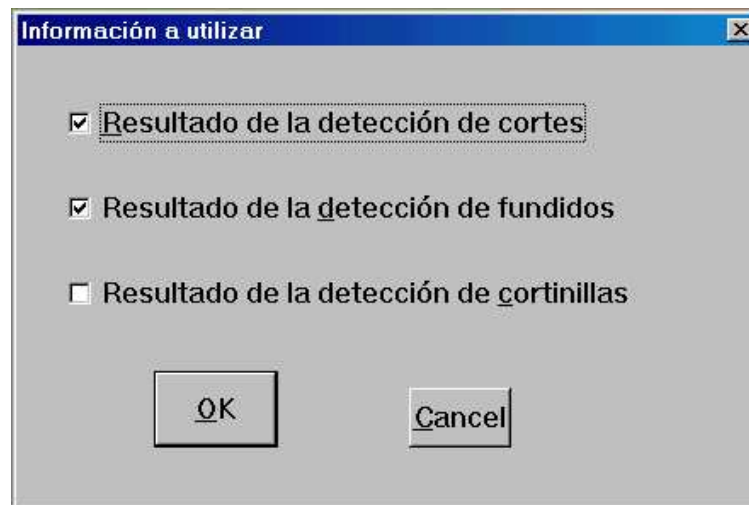


Figura 5.2: Ventana para configurar los parámetros del proceso de detección de fundidos.

Una vez dispone de toda la información requerida, comienza el proceso descrito en el apartado anterior. A partir de la secuencia activa obtendrá su número de fotogramas, y con ello creará el vector que contiene al índice. Se inicializa el índice a cero y de acuerdo a la información de los efectos deseados, se modifican sus elementos. Una vez disponemos del índice, ya es posible obtener cada plano de la secuencia. Para ello se ha programado la siguiente función:

```
P8DetectarShots(char *nameFile,int *indice,short numFrames),
```

con los parámetros: `*nameFile` es el puntero a la cadena de texto con el nombre del fichero donde guardar los resultados, `*indice` es el puntero al vector que contiene los elementos del índice y `numFrames` es el número de fotogramas de la secuencia, y que se corresponde con la dimensión del índice. Esta función toma el índice y lo recorre empezando desde el principio. El primer plano comenzará con el primer fotograma y no terminará hasta que en el índice no se encuentre un elemento distinto de 0. Si ese elemento es un 1, indicará que el primer plano terminaba en el fotograma anterior, y que éste es el primero del segundo plano. En el caso de tratarse de un 2, un 3, o un 4, ocurrirá lo mismo, pero el segundo plano terminará cuando se encuentre un 0 (mientras dure un efecto gradual, todos sus fotogramas tendrán el mismo número distinto de 0 y 1). De igual forma se procederá con el resto de la secuencia, teniendo en cuenta que el último plano terminará en el último fotograma.

De esta forma para cada plano disponemos de los siguientes datos: número, fotograma inicial, fotograma final y tipo (plano normal o lineal, fundido, cortinilla horizontal

y cortinilla vertical). La función `P8DetectarShots()` guarda estos datos en el fichero `*nameFile`. La sintaxis será: una línea por plano, y para cada uno:

- Un entero correspondiente al número de plano.
- Otro entero separado por un espacio, para el primer fotograma.
- Otro entero separado por otro espacio, para el último fotograma.
- Y después de un espacio, podrán aparecer uno o dos caracteres juntos. Indicarán el tipo de plano, y serán: “l” para los lineales, “f” para los fundidos, “ch” para las cortinillas horizontales y “cv” para las cortinillas verticales.

Si todo sale bien, al finalizar nos muestra una ventana con los ficheros donde guarda los resultados.

Se trata de dos ficheros. El primero, *indice.txt*, contiene al índice creado anteriormente, es decir, para una secuencia de  $N$  fotogramas tendrá  $N$  elementos y sus elementos serán 0, excepto en los fotogramas pertenecientes a un efecto, que tendrá un número que indique el efecto. Este fichero puede ser visto como la superposición de los ficheros *indice1.txt*, *indice2.txt* e *indice3.txt*.

El nombre del otro fichero se construye de la siguiente manera: se toma el nombre de la secuencia activa, se elimina la extensión “.val”, y se añade la extensión “.sht”. Su contenido es el resultado de la función `P8DetectarShots()`. Llegados a este punto del proceso de la segmentación, nos ha parecido más interesante que el nombre del fichero resultado incluyera el de la secuencia, y no como ocurría hasta ahora que los nombres de los ficheros resultado eran siempre el mismo. La razón es bien sencilla: esta información ya es definitiva y bastante útil para posteriores aplicaciones, por tanto interesará tenerla asociada a su secuencia. Un ejemplo de fichero “.sht” es el siguiente (corresponde a *news\_8.sht*):

```
1 0 37 l
2 38 75 l
3 76 97 l
4 98 116 l
5 117 130 l
6 131 152 l
7 153 168 l
8 169 188 l
9 189 281 l
10 282 303 ch
11 304 333 l
12 334 341 f
13 342 557 l
```

```
14 558 568 1
15 569 569 f
16 570 760 1
17 761 875 1
18 876 1054 1
19 1055 1265 1
20 1266 1296 1
21 1297 1328 1
22 1329 1373 1
23 1374 1515 1
24 1516 1638 1
25 1639 1653 cv
26 1654 1665 1
27 1666 1906 1
```

Pueden resultar curiosas algunas cosas del contenido de *news\_8.sht*. Como se observa, se ha segmentado en 27 planos; y el primer fotograma del primer plano es el 0 (esto ocurre porque en el PDIWIN32, el primer fotograma de una secuencia siempre es el 0). Contiene dos fundidos, una cortinilla horizontal y una cortinilla vertical. El plano número 15 es un fundido de un solo fotograma: se trata de un ejemplo de corte con fotograma intermedio (aunque en esta secuencia, se corresponde con una falsa transición).

En las secciones siguientes, veremos la utilidad de estos ficheros.

## 5.3 EXTRACCIÓN DE FOTOGRAMAS CLAVE

Ya se dijo en la introducción al proyecto, que un acceso eficiente a la información de las secuencias de vídeo es bastante difícil debido a su formato no estructurado, y en la mayoría de los casos, a su gran longitud. Las técnicas conocidas como abstracción y sumarización (descripción del contenido) pretenden resolver esta dificultad. Se basan principalmente en detección de cambios de plano, para su segmentación, y en la extracción de fotogramas clave.

El uso de los fotogramas clave reduce en gran medida los datos requeridos en indexación de vídeo y provee de un método de trabajo ordenado para la búsqueda de contenido en vídeo.

En el apartado anterior hemos conseguido separar los planos de la secuencia. Después de segmentada, los fotogramas clave pueden ser extraídos para cada plano. Un **fotograma clave** es aquel fotograma que puede representar el contenido completo de un plano. Dependiendo de la complejidad en contenido de un plano, pueden ser extraídos uno o más fotogramas clave.

Al igual que hicimos para los cortes, a continuación realizaremos una revisión de los métodos más utilizados para la extracción de fotogramas clave. Después nos centraremos en los programados para el proyecto. Se han realizado muchos progresos en este área, sin embargo, existen aproximaciones que o bien son muy caras computacionalmente o no pueden capturar de forma efectiva el contenido del plano.

Las técnicas más utilizadas en la extracción de los fotogramas clave son [25] [36]:

- **Aproximaciones basadas en las fronteras del plano:** Después de segmentar la secuencia de vídeo en sus planos, un camino natural y fácil de extracción de *keyframes* es usar el primer fotograma como fotograma clave del plano. Demasiado simple, el número de fotogramas clave para plano se limita a uno, sin mirar la complejidad visual del plano. Otro inconveniente es que, el primer fotograma normalmente no es “estable” y no captura todo el contenido visual del plano.
- **Aproximaciones basadas en el contenido visual.** Se han propuesto varias alternativas,
  - *Criterio basado en el plano:* El primer fotograma será siempre seleccionado como primer fotograma clave; pero, si se necesitan más, se tomarán mediante otro criterio.
  - *Criterio basado en las características de color:* El fotograma actual del plano se compara con el último fotograma clave. Si han ocurrido cambios significativos, el fotograma actual se selecciona como nuevo fotograma clave.
  - *Criterio basado en el movimiento:* Para un plano que sea un *zoom*, como mínimo deberán seleccionarse dos fotogramas: el primero y el último como representación global, mientras que otros pueden representar un punto de vista intermedio. Para un plano panorámica, se deben seleccionar como fotogramas clave el 30% de los fotogramas.
- **Aproximaciones basadas en análisis de movimiento:** Primero se calcula el flujo óptico de cada fotograma, y después se computa una métrica sencilla de movimiento basada en el flujo óptico. Finalmente, se analiza la métrica como una función del tiempo para seleccionar los fotogramas clave que tienen una actividad local de movimiento mínima. La justificación de esta aproximación es que en la mayoría de los planos, los fotogramas clave se identifican por su escaso movimiento, la cámara se para, para enfatizar lo importante.
- **Aproximaciones basadas en la actividad del plano:** Primero se computa un indicador de la actividad de cada fotograma (usando una combinación de varias características) y se obtiene una curva de actividad del plano, los mínimos locales se seleccionan como fotogramas clave.



Las primeras aproximaciones, basadas en los límites del plano son relativamente rápidas. Sin embargo, no son efectivas para capturar el contenido visual, ya que el primer fotograma no tiene por qué ser un fotograma clave. Las dos últimas aproximaciones son más sofisticadas, por el análisis de movimiento y de actividad; sin embargo, son computacionalmente muy caras y sus aproximaciones mediante mínimos locales no son necesariamente correctas.

Idealmente, los fotogramas clave deben capturar la semántica de un plano. Sin embargo, en el estado actual, la técnicas de Visión por Computador no han avanzado lo suficiente para la generación correcta de los fotogramas clave. Actualmente las técnicas más vanguardistas se basan en el *clustering* o agrupamiento de fotogramas, utilizando como información características visuales de bajo nivel (color, textura, formas, bordes de los objetos, etc).

### 5.3.1 Criterios de extracción de fotogramas clave elegidos para el proyecto

A la hora de enfrentarnos al problema de los fotogramas clave, hemos optado por soluciones sencillas, tanto conceptual como computacionalmente. Se ha propuesto la posibilidad de utilizar tres criterios diferentes para seleccionar los *keyframes*:

1. **Primer fotograma de cada plano:** Es la más sencilla y rápida, pero como se ha comentado en el apartado anterior, no es muy efectiva por no capturar siempre el contenido visual del plano.
2. **Mínima diferencia entre fotogramas:** Para poder aplicar esta técnica es necesaria la información sobre el módulo de la diferencia entre fotogramas:  $d_c(X, Y)$ ; que como sabemos, se encuentra en fichero *moddifer.txt*. Consiste en localizar la mínima diferencia para los fotogramas del plano correspondiente. Este mínimo indicará los dos fotogramas del plano que más se parecen, y se considerará que el primero es el fotograma clave. Con los resultados obtenidos para las secuencias de prueba, podemos decir que esta técnica produce resultados muy similares a la anterior. A nuestro modo de ver, el problema se encuentra en la dificultad de capturar el contenido de un plano con un solo fotograma; para solucionar esto hemos propuesto una tercera alternativa.
3. **Utilización de una técnica de *clustering* o agrupamiento:** En este caso, podrán tenerse una o más fotogramas clave.

Imaginemos que tenemos un plano de una secuencia de vídeo,  $s = \{f_1, f_2, \dots, f_N\}$  obtenido mediante nuestros algoritmos de segmentación temporal. El objetivo es agrupar los  $N$  fotogramas en  $M$  *clusters* o grupos,  $\sigma_1, \sigma_2, \dots, \sigma_M$ , donde todos los fotogramas de un grupo serán consecutivos. Para medir la similitud o parecido

en contenido visual de dos fotogramas, utilizaremos, al igual que en la técnica anterior  $d_c(X, Y)$ <sup>1</sup>.

Como cualquier técnica de *clustering*, nuestro algoritmo tiene un umbral,  $\delta$ , que controla la densidad de agrupamiento. Cuanto mayor es el valor de  $\delta$ , menor será el número de grupos. Básicamente, este algoritmo consiste en: Para cada fotograma del plano, se toma su diferencia con el anterior y si la diferencia es inferior al umbral, el fotograma se considera del actual grupo, si la diferencia es mayor del umbral, se considera que el fotograma actual es el primero de un nuevo grupo. Una vez se han agrupado todos los fotogramas del plano, se considerarán fotogramas clave, el primer fotograma de cada grupo.

En el siguiente apartado comentaremos como se han programado estas tres técnicas en el proyecto.

### 5.3.2 Programación en el PDIWin32

Para extraer los fotogramas clave de una secuencia se ha situado una función, **Extracción KeyFrames**, tal y como se ve en la figura 5.3.

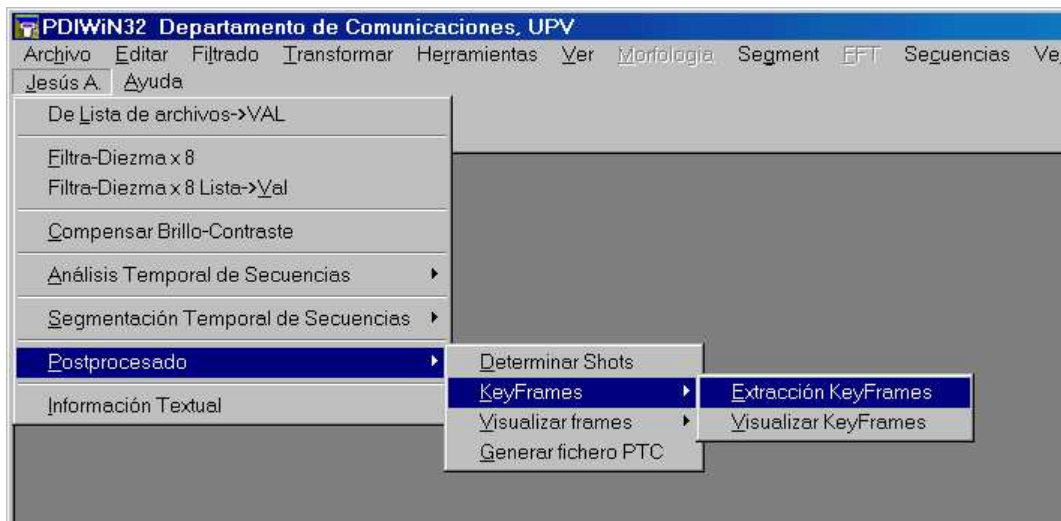


Figura 5.3: Entrada de menú para extraer los fotogramas clave de una secuencia.

Al igual que ocurre con todas las funciones anteriores que hemos programado en este proyecto, la imagen activa ha de ser una secuencia *VAL*, ya que en caso contrario se nos mostrará un mensaje de error.

<sup>1</sup>Se podrían programar técnicas más sofisticadas, que utilizarasen como medida de la similitud características de color, textura, bordes y formas de los objetos o combinación de varias.

Lo primero que hará será mostrarnos un cuadro de diálogo, figura 5.4, para indicarle el método a utilizar en la extracción. Las tres alternativas son las comentadas en el apartado anterior.



Figura 5.4: Ventana para elegir el método de extracción de los fotogramas clave.

Dependiendo de la opción elegida, se buscarán los ficheros que contengan la información necesaria. Para cualquiera de los tres métodos es necesario que previamente se haya segmentado temporalmente la secuencia; por tanto, siempre se buscará el fichero ".sht" (recordemos que se construye de la siguiente manera: se toma el nombre de la secuencia activa, se elimina la extensión ".val", y se añade la extensión ".sht"). En el caso de no encontrarlo nos mostrará una ventana de error indicándolo.

Si se toma como método de extracción de fotogramas uno de los dos últimos, tal y como hemos visto en el apartado anterior, será necesario disponer también del fichero *moddifer.txt*. Y en el caso de no encontrarlo en el directorio actual nos mostrará una ventana de error similar a la anterior.

Una vez dispone de toda la información requerida, comienza el proceso descrito en el apartado anterior. A partir de la secuencia activa obtendrá su nombre, y con él el del fichero ".sht" con el resultado de la segmentación temporal. Si se elige el método del primer fotograma de cada secuencia, con la información de este fichero (recordemos que incluye: número de plano, primer fotograma, último fotograma y tipo) será suficiente.

Si se elige la segunda opción, mínima diferencia entre fotogramas, el proceso se complica un poco más. Para ello, se ha programado la función:

```
int P8ExtraeKeyFrames1(char *nameFile),
```

con el parámetro *\*nameFile*, que es el puntero a la cadena de texto con nombre del fichero ".sht" donde están guardados los resultados de la segmentación temporal. Lo

primero que hace esta función es comprobar si encuentra el fichero *moddifer.txt*. En caso positivo, y con la información de cada plano (primer y último fotograma) más la información del módulo de la diferencia buscará la diferencia mínima, y el fotograma a que corresponde. Éste será el fotograma clave. El resultado de esta función se guarda en un fichero de texto, *cluster.txt*, cuyo contenido consiste en una línea por plano, y para cada uno se almacenan dos datos: número de plano y fotograma clave. Este fichero se utiliza inmediatamente.

En el caso de elegir la opción de agrupamiento, el proceso es todavía más complejo y viene implementado por la función:

```
int P8ExtraeKeyFrames2(char *nameFile).
```

Similar a la anterior, también utiliza la información de *moddifer.txt* para realizar el *clustering*. En el apartado anterior ya hemos comentado que consiste en, una vez fijado el umbral (lo hemos dejado fijo, con un valor de  $\delta = 1.5$ , por obtener con él buenos valores en las secuencias de prueba), tomar los valores del módulo de la diferencia y comprobar si superan o no el umbral, de tal forma que surja un nuevo grupo o no, respectivamente. El resultado de esta función también se guarda en el fichero *cluster.txt*, y su contenido es similar al obtenido para `P8ExtraeKeyFrames1()`, con la diferencia que aquí podremos tener varios fotogramas clave para un plano (una línea por plano, y para cada uno se almacena el número de plano, luego un espacio en blanco, y después los fotogramas clave separados por el carácter “&”).

Una vez disponemos del fichero *cluster.txt* (recordemos que para el método del primer fotograma no es necesario), se tiene toda la información sobre los fotogramas clave de cada plano y por tanto, esta información podrá guardarse en el fichero correspondiente.

El nombre del fichero se construye de la siguiente manera: se toma el nombre de la secuencia activa, se elimina la extensión “.val”, y se añade la extensión “.key”. La sintaxis será: una línea por plano, y para cada uno se construye una cadena de texto con la siguiente información

- Un entero correspondiente al número de plano, después un espacio.
- Carácter “(”.
- Otro entero para el primer fotograma.
- Carácter “-”
- Otro entero para el último fotograma.
- Carácter “)”, después un espacio.

- Podrán aparecer uno o dos caracteres juntos. Indicarán el tipo de plano, y serán: “l” para los lineales, “f” para los fundidos, “ch” para las cortinillas horizontales y “cv” para las cortinillas verticales.
- Y después de un espacio, aparecerá una cadena de texto con los fotogramas clave separados por “& ”.

Un ejemplo de fichero “.key” es el siguiente (corresponde a *news\_8.key* eligiendo la opción de extracción de fotogramas clave por mínima diferencia):

```
1(0-37) l 0
2(38-75) l 38
3(76-97) l 83
4(98-116) l 100
5(117-130) l 125
6(131-152) l 151
7(153-168) l 164
8(169-188) l 171
9(189-281) l 274
10(282-303) ch 282
11(304-333) l 315
12(334-341) f 341
13(342-557) l 472
14(558-568) l 561
15(569-569) f 569
16(570-760) l 742
17(761-875) l 808
18(876-1054) l 987
19(1055-1265) l 1165
20(1266-1296) l 1294
21(1297-1328) l 1315
22(1329-1373) l 1339
23(1374-1515) l 1475
24(1516-1638) l 1522
25(1639-1653) cv 1653
26(1654-1665) l 1656
27(1666-1906) l 1906
```

Como se puede observar, además de la información de los fotogramas clave, también incluye toda la información del fichero *news\_8.sht*. Todos estos datos se utilizarán para las funciones de visualización que describiremos en el siguiente apartado.

## 5.4 VISUALIZACIÓN DE TRANSICIONES GRADUALES Y *KEYFRAMES*

En esta sección vamos a describir las funciones añadidas al proyecto para visualizar las transiciones graduales y los fotogramas clave sobre la secuencia *VAL*.

Por tratarse de secuencias, el manejo de ficheros *VAL* en el PDIWIN32 permite la reproducción continua de la misma, el retroceso y avance fotograma a fotograma, situarse en un fotograma deseado de forma aleatoria, etc . . . , tal como se hace por ejemplo, con las secuencias *AVI* en otros programas multimedia.

Con las siguientes funciones, podremos visualizar de forma automática aquellos fotogramas más interesantes como resultado del proceso de segmentación.

### 5.4.1 Visualización de fotogramas clave en el PDIWin32

Teniendo activa una secuencia y habiendo extraído sus fotogramas clave (es decir, debe existir un fichero *.key* con el nombre de la secuencia), es posible visualizar solo sus fotogramas clave. En las figuras 5.5 y 5.6 aparecen las dos entradas de menú correspondientes a esta función (se ha incluido dos veces para poder mantener un orden lógico y estructurado, que facilitará la tarea del futuro usuario).



Figura 5.5: Entrada de menú para visualizar los fotogramas clave de una secuencia.

El funcionamiento es bien sencillo, y un ejemplo de su ejecución puede observarse en la figura 5.7. Esta función leerá línea a línea el fichero *.key*, y a partir de la información contenida:

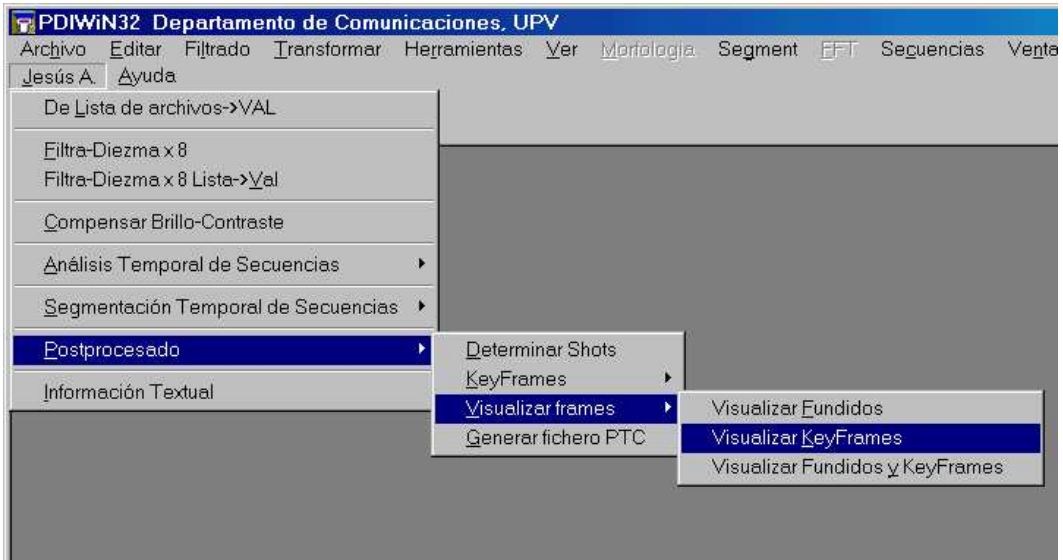


Figura 5.6: Entrada de menú para visualizar los fotogramas clave de una secuencia.

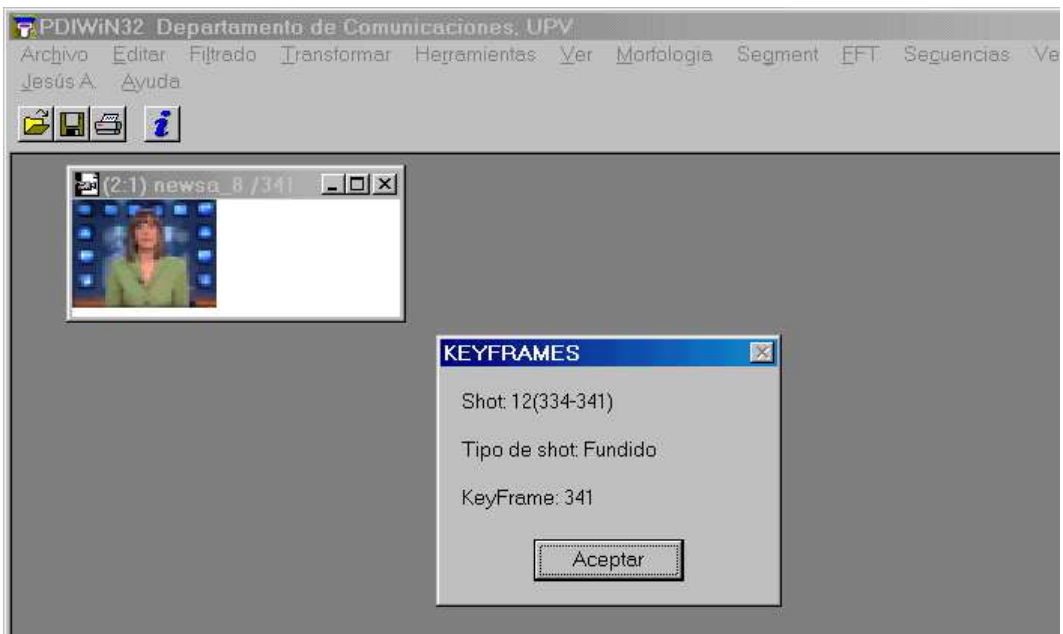


Figura 5.7: Ejemplo de visualización de los fotogramas clave en el PDIWIN32.

- Número de plano.
- Primer y último fotograma del plano.
- Tipo de plano.
- Fotogramas clave (separados por “& ”).

nos sacará una ventana de información (con todos los datos anteriores) para cada fotograma clave; y al mismo tiempo, la ventana de la secuencia mostrará este fotograma.

Con este modo de visualización (y siempre que la extracción de fotogramas clave haya sido correcta) es posible darse una idea del contenido semántico de una secuencia, a través de un conjunto reducido de sus fotogramas.

#### 5.4.2 Visualización de transiciones graduales en el PDIWin32

Teniendo activa una secuencia y habiendo determinado sus planos (es decir, debe existir un fichero “.sht” con el nombre de la secuencia), es posible visualizar sus transiciones graduales (fundidos y cortinillas). En la figura 5.8 aparece la entrada de menú correspondiente a esta función.

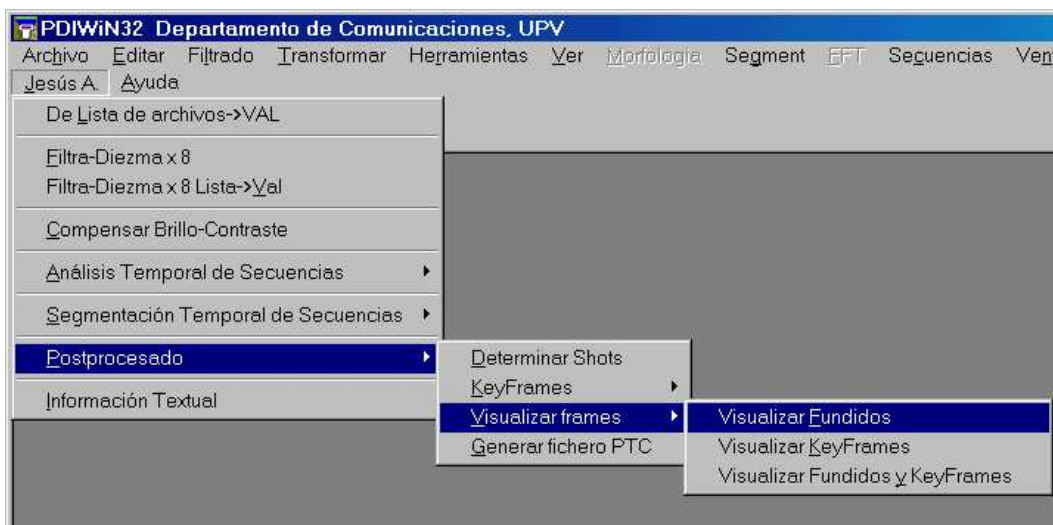


Figura 5.8: Entrada de menú para visualizar las transiciones graduales de una secuencia.

Al igual que con los fotogramas clave, su funcionamiento es sencillo, y un ejemplo de su ejecución se observa en la figura 5.9 Esta función leerá línea a línea el fichero “.sht”, y a partir de la información contenida:

- Número de plano.



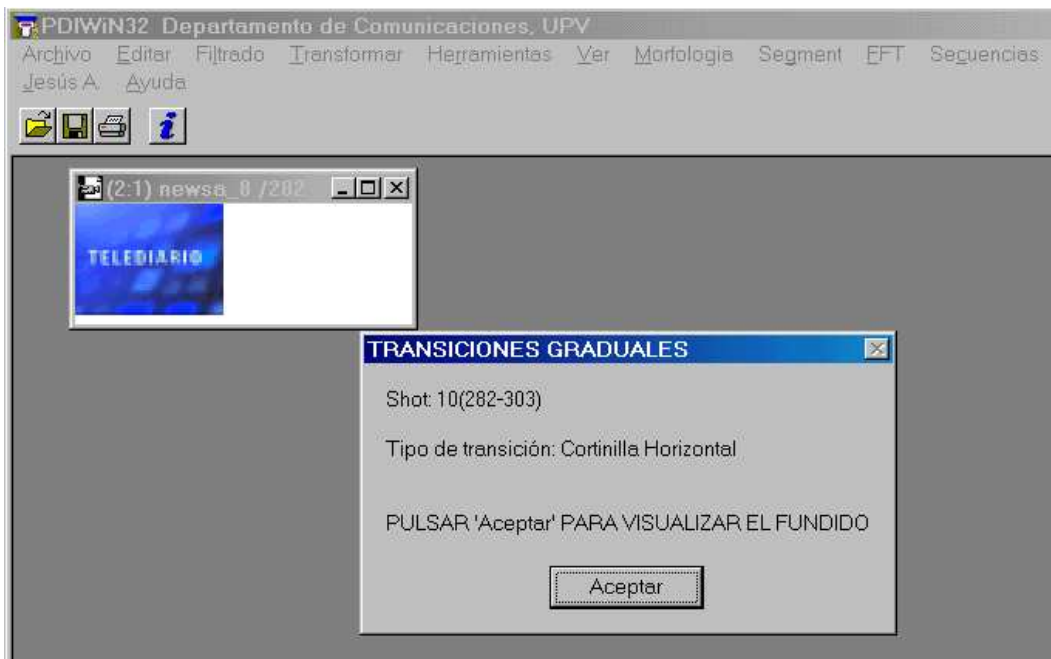


Figura 5.9: Ejemplo de visualización de los transiciones graduales en el PDIWIN32.

- Primer y último fotograma del plano.
- Tipo de plano (codificado mediante dos caracteres).

nos sacará una ventana de información (con todos los datos anteriores) para cada transición gradual; al pulsar el botón de “Aceptar”, sobre la ventana de la secuencia podrán visualizarse el efecto gradual.

Con este modo de visualización, podremos comprobar si se han detectado de forma correcta los fundidos y las cortinillas. En caso de encontrarse muchos errores, puede ser interesante volver a realizar todo el proceso de segmentación temporal, utilizando umbrales diferentes, y refinando de esta forma el resultado final. En los fundidos cortos (de dos o tres fotogramas) será muy difícil observar el efecto de transición.

### 5.4.3 Visualización de fotogramas clave y transiciones graduales

Además de las dos funciones comentadas anteriormente, hemos añadido la posibilidad de realizarlas de forma conjunta, figura 5.10

Según el criterio elegido en este proyecto, consideraremos que todos los fotogramas de una transición gradual están en el mismo plano. Al extraer sus fotogramas clave, si utilizamos el criterio del primer fotograma o el de menor módulo de la diferencia, solo

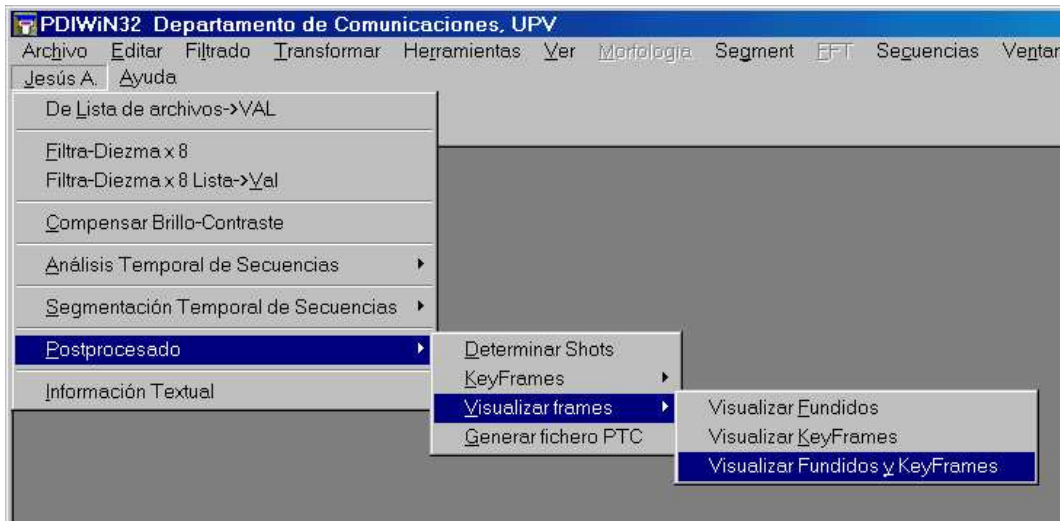


Figura 5.10: Entrada de menú para visualizar fotogramas clave y transiciones graduales de una secuencia.

tendremos un fotograma y al visualizarlo no tendrá mucho sentido (en cualquier caso, sabremos que pertenece a una transición gradual, por la información que nos muestra la ventana de visualización). De igual forma, si elegimos el criterio de extracción basado en agrupamiento, se tomarán la mayoría de los fotogramas del plano y el visualizarlos uno a uno tampoco tiene mucho sentido.

Por eso, consideramos que sería más interesante visualizar simultáneamente fotogramas clave y fundidos. De esta forma cuando nos encontremos con un plano “normal”, visualizaremos los fotogramas clave (uno o más, dependiendo del criterio) que nos indicarán su contenido. Y cuando un plano se corresponda con un fundido o una cortinilla, lo visualizaremos totalmente.

## 5.5 GENERACIÓN DE INFORMACIÓN TEXTUAL

En el capítulo de conceptos básicos ya definimos lo que es la indexación de vídeo. Esta clara la dificultad de conseguir una descripción automática del contenido de una secuencia, y además si se quiere que esta descripción incluya el contenido semántico a un nivel de abstracción alto, por ejemplo: “Imágenes del Caso Marey: Declaraciones de Segundo Marey en el tribunal supremo. Plano de llegada de Marey al tribunal acompañado de su abogada...”<sup>2</sup>, se ha de añadir el trabajo de una persona.

Desde hace muchos años la mayoría de las televisiones disponen de personal encargado del indexado manual de sus programas. En general, consiste simplemente en información sobre fechas de emisión, títulos de crédito y una breve descripción del contenido. Pero para algunos casos, como los informativos, la descripción es muchísimo más detallada. La persona encargada del trabajo visiona el vídeo y describe todo lo que aparece en él, plano a plano. A pesar de la profesionalidad y destreza, resulta evidente que es un proceso lento y costoso. Una forma de facilitar la tarea y añadir rapidez, consistiría en segmentar previamente la secuencia de vídeo en sus planos y extraer los fotogramas clave; de esta forma, el operario solo tendría que visualizar los fotogramas clave para describir el contenido de cada plano.

### 5.5.1 Información textual con el PDIWin32

A lo largo de todo el proyecto, hemos pasado por diversos niveles de abstracción en el estudio de las secuencias de vídeo, pasando del análisis numérico de fotogramas a la extracción de los fotogramas clave. Se ha tratado de un proceso ascendente, cuyo final lógico es la obtención de un fichero de texto con la descripción del contenido de la secuencia. Y este es el objetivo de la función que hemos programado.

Al igual que para todas las funciones mostradas en este capítulo, se debe tener activa una secuencia para poder construir el fichero de texto que contendrá la información textual. En la figura 5.11 aparece la entrada de menú correspondiente a esta función.

Esta función, necesita como información de **entrada**:

- Fichero “.sht” ⇒ Contiene el resultado de la segmentación de la secuencia en sus planos.
- Fichero “.key” ⇒ Contiene el resultado de la extracción de los fotogramas clave de cada plano.

---

<sup>2</sup>Extraído de la descripción del contenido de la secuencia *news\_a\_8.val*, procedente de las Bases de Datos Documentales de Radiotelevisión Española, ref: CDEPH001.



Figura 5.11: Entrada de menú para la descripción textual de la secuencia.

Como ya comentamos, el nombre de estos ficheros se construye de la siguiente manera: se toma el nombre de la secuencia activa, se elimina la extensión *".val"*, y se añade la extensión correspondiente. De esta misma manera se construye el nombre para el fichero **resultado** del proceso de indexado:

- Fichero *".txt"*  $\Rightarrow$  Contiene la descripción textual del contenido semántico de cada uno de los planos de la secuencia, así como otra información de la secuencia.

Al seleccionar la función, el primer cuadro de diálogo que aparece es el de la figura 5.12. Este cuadro tiene una triple utilidad,

- Muestra información de la secuencia activa: nombre del fichero, número de fotogramas, ancho y alto de la imagen, etc. Todo esta información aparece por defecto, con los datos de la secuencia; pero puede ser modificada por el usuario.
- Permite introducir un título y un código para identificar la secuencia. La fecha que aparece es la del reloj del sistema.
- Se puede elegir la opción de describir el contenido de cada plano (está activa por defecto).

Al aceptar este primer cuadro, pueden ocurrir dos cosas. Si no se elige la opción de describir el contenido por plano, el programa generará el fichero *".txt"* con la información introducida en el cuadro y la que dispone de cada plano, procedente de los ficheros *".sht"* y *".key"*. Al finalizar nos mostrará un mensaje para preguntarnos si queremos

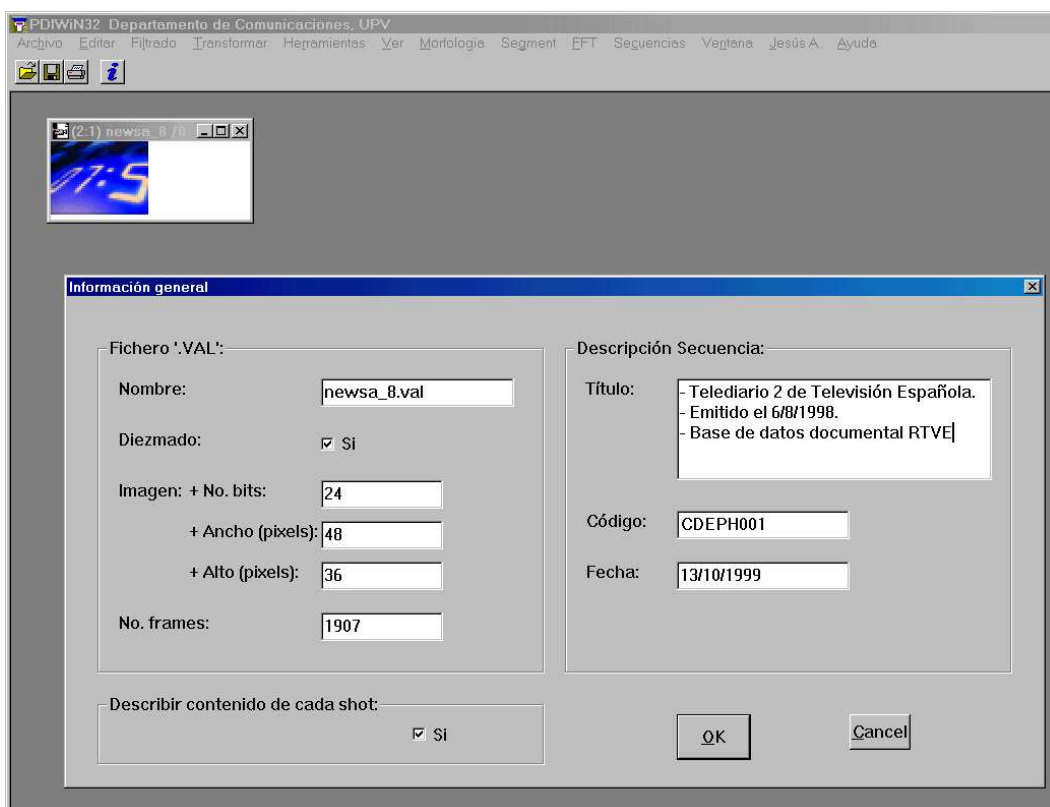


Figura 5.12: Cuadro de diálogo mostrando información general de la secuencia.

ver su contenido; si se acepta, creará una nueva ventana con un editor de texto donde aparecerá el contenido del fichero.

Pero si se elige la opción de describir el contenido de cada plano, nos aparecerá un cuadro de diálogo como el de la figura 5.13. La primera vez nos mostrará toda la

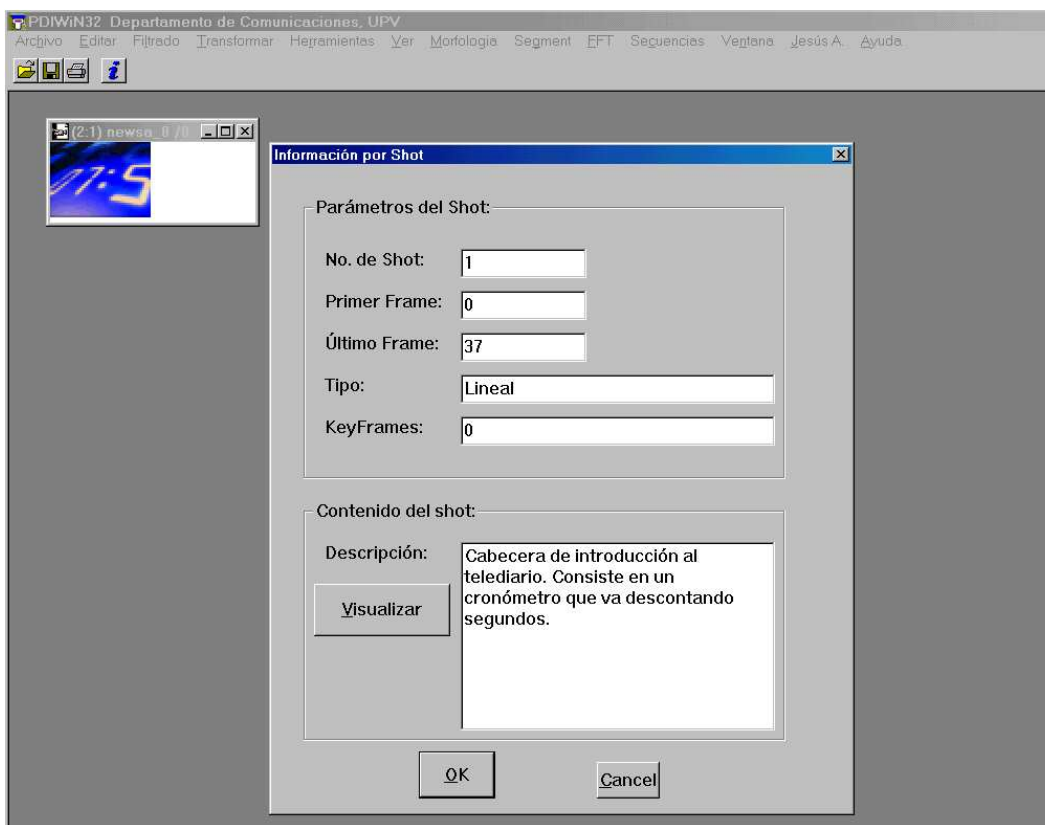


Figura 5.13: Cuadro de diálogo para describir el contenido de cada plano de la secuencia.

información necesaria del plano (número, primer y último fotograma, tipo de plano y fotogramas clave), al mismo tiempo situará en la ventana de la secuencia, la imagen del primer fotograma del plano. Este cuadro de diálogo dispone de un botón: “Visualizar”, y al pulsarlo, nos visualizará el plano actual. Se puede elegir tantas veces como se desee, y de esta manera será más fácil poder describir brevemente su contenido. Cuando se ha finalizado el proceso de describir el plano, se pulsa el botón “OK” y aparecerá el mismo cuadro pero con los datos del plano siguiente (antes se guardará en el fichero la información definitiva del plano anterior). El proceso se repetirá hasta completar todos los planos de la secuencia; y al terminar nos mostrará el mensaje para preguntarnos si se desea ver el fichero “.txt” resultado, al igual que hemos comentado anteriormente.

A continuación incluimos como ejemplo final, el fragmento del contenido de un fichero ".txt" (corresponde a *news\_8.txt*):

```
=====
SEGMENTACIÓN TEMPORAL DE SECUENCIAS '.val', USANDO PDIWin32.
=====
```

```
TÍTULO: {Telediario 2 de Televisión Española.
Emitido el 6/8/1998.}
CÓDIGO: {CDEPH001}
FECHA: {13/10/1999}
```

```
-----
-----
INFORMACIÓN GENERAL.
```

```
-----
-----
- NOMBRE FICHERO: {news_8.val}
- SECUENCIA DIEZMADA: {Si}
- CARACTERÍSTICAS IMAGEN
+ No DE BITS: {24}
+ ANCHO: {48}
+ ALTO: {36}
- No FRAMES: {1907}
```

```
-----
-----
INFORMACIÓN PARA CADA SHOT.
```

```
-----
-----
- SHOT {1}:
+ TIPO DE SHOT: {Lineal}

+ PRIMER FRAME: {0}
+ ÚLTIMO FRAME: {37}
+ KEYFRAMES: {0}

+ DESCRIPCIÓN: {Cabecera de introducción al telediario.
Consiste en un cronómetro que va descontando segundos.}
```

```
-----
:
```

---

- SHOT {27}:  
+ TIPO DE SHOT: {Lineal}

+ PRIMER FRAME: {1666}  
+ ÚLTIMO FRAME: {1906}  
+ KEYFRAMES: {1906}

+ DESCRIPCIÓN: {Rueda de prensa de Jose María Aznar en la Moncloa.}

---

---

P.F.C de Jesús Angulo López, E.T.S.I.Teleco - U.P. Valencia

---