

Apéndice B

Las secuencias *VAL* y las listas de ficheros

B.1 INTRODUCCIÓN

Cómo ya hemos comentado al inicio del proyecto, el formato utilizado en las secuencias de trabajo ha sido el propio del PDIWIN32: ficheros **.VAL*. Se trata de ficheros que contienen secuencias de imágenes en su interior.

El origen de este formato se encuentra en el propio origen del programa (diseñado inicialmente por la profesora Valery Naranjo [23]), y su característica más destacable es que los ficheros *VAL* almacenan las imágenes sin comprimir. Se trata, por tanto, de un formato *transparente* similar a los mapas de bits, *BMP*, utilizados habitualmente en Windows. La ventaja más inmediata frente a los ficheros *AVI* o *MPG* es que nos permite extraer los fotogramas uno a uno (y de forma aleatoria) para su procesado, sin necesidad de conocer los drivers de compresión/descompresión.

Al mismo tiempo nos encontramos con varios inconvenientes. El más importante a nivel local es el del espacio de memoria ocupado. Cuando se trabajan con secuencias largas (del orden de minutos) la memoria necesaria para almacenarlas se dispara (del orden de Gigabytes). Por otra parte, tenemos el problema de la compatibilidad de formatos: el formato *VAL* sólo es posible utilizarlo en nuestro programa. Las Tarjetas Capturadoras de Video (para pasar de formato de video analógico a secuencia de frames digitales) pueden programarse, de forma que lo generen directamente o bien utilizar algunos de los kits de desarrollo de aplicaciones multimedia para crear programas que pasen de los formatos estándares al formato *VAL* y viceversa. Nosotros hemos elegido otra opción: trabajar con ficheros imagen (a ser posible comprimidos), uno para cada frame, en lugar de ficheros secuencia. Una secuencia podrá ser un fichero *lista* conteniendo los nombres de todos los ficheros imagen.

La principal ventaja de las listas de ficheros es su facilidad de edición. Por ejemplo,

seleccionar una subsecuencia simplemente consistirá en copiar los nombres de los ficheros que interesen a otro fichero *lista*. De igual forma, para el resto de posibilidades de edición de la secuencia a nivel de fotograma, solo se deberán añadir o eliminar nombres, añadiendo o eliminando fotogramas donde interese.

A continuación y para terminar esta sección comentaremos la estructura de un fichero *VAL* y la forma como se visualizan en el programa. En la sección siguiente entraremos más en profundidad a comentar el tema de los ficheros *lista*.

B.1.1 Estructura de un fichero *VAL*

Un fichero *VAL* está formado por una cabecera y un cuerpo de fichero o datos, como puede verse en la figura B.1. La cabecera contiene la siguiente información:

- **Ancho**(*SHORT*, 16 bits): Ancho de la imagen en pixels. Si es negativo, indica que el primer campo es el de las líneas pares, en el caso de ficheros entrelazados, y la anchura es el módulo
- **Alto**(*SHORT*, 16 bits): Altura de la imagen en pixels. Si es negativo, indica que la imagen es RGB y la altura es el módulo.
- **No. de fotogramas**(*SHORT*, 16 bits): Número de fotogramas del fichero.

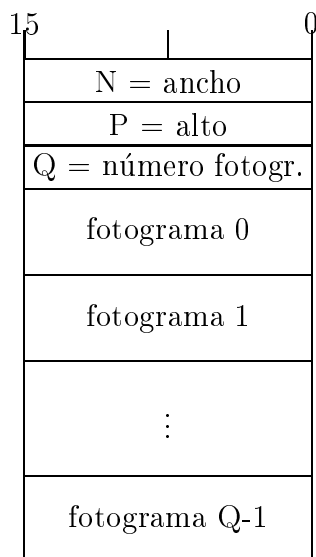


Figura B.1: Formato del fichero *VAL*.

El cuerpo del fichero lleva todos los fotogramas, encapsulados uno tras otro. Si las imágenes son de:

- **Grises:** Se almacenan por filas, empezando por la de arriba. El número de bytes de cada fila coincide con Ancho.
- **RGB:** Se almacenan por filas empezando por la de arriba. El número de bytes de cada fila es 3*Ancho, y los datos se escriben BGR-BGR-BGR-..., es decir, las tres componentes de un pixel ocupan posiciones consecutivas.

Para escribir o leer fotogramas de ficheros *VAL* se dispone de dos funciones que facilitan notablemente la tarea:

GetValFrame

- **Sinopsis:** Tomar un fotograma de una secuencia *VAL*.
- **Formato:** `GetValFrame(FILE *fich,int numfot,imgdes *desimg);`
 - `FILE *fich:` Puntero al fichero *VAL* que contiene a la secuencia. Debe estar abierto con permiso de lectura.
 - `int numfot:` Número del fotograma a extraer.
 - `imgdes *desimg:` Puntero al descriptor de la imagen donde copiar el fotograma extraído.

AddFrametoVal

- **Sinopsis:** Añadir un fotograma de una secuencia *VAL*.
- **Formato:** `AddFrametoVal(imgdes *srcimg,FILE *fich);`
 - `imgdes *srcimg:` Puntero al descriptor de la imagen que añadir a la secuencia como un nuevo fotograma.
 - `FILE *fich:` Puntero al fichero *VAL* que contiene a la secuencia. Debe estar abierto con permiso de escritura.

B.1.2 Visualización de secuencias *VAL* en pantalla

A nivel de programación interna, decir que la clase de ventana destinada a albergar una secuencia *VAL* es la clase `TMyMDIChild`, la misma clase encargada del resto de tipo de imágenes en el `PDIWIN32`. La razón se encuentra en que los fotogramas de una secuencia de este tipo son interpretados como imágenes `VICTOR` (librería de funciones en C, para el procesamiento de imágenes, linkada al programa y utilizada con profusión).

Sin embargo, por tratarse de una secuencia, es necesario que pueda realizarse la reproducción continua de la misma, el retroceso y avance fotograma a fotograma, etc ..., tal como se hace por ejemplo, con las secuencias *AVI* en otros programas multimedia.

En la tabla B.1 podemos ver su descripción:

Como curiosidad, decir que el `PDIWIN32` también permite la reproducción de ficheros *AVI* y las teclas utilizadas son las mismas.

TECLA	FUNCIÓN
Return	Reproducción continua de la secuencia, desde el fotograma actual hasta el último.
Pausa	Pausa en la reproducción.
Inicio	Traslada la secuencia al primer fotograma.
Fin	Traslada la secuencia al último fotograma.
Página Anterior	Retrocede un fotograma en la secuencia.
Página Siguiente	Avanza un fotograma en la secuencia.
Barra Espaciadora	Despliega un cuadro de diálogo permitiendo especificar a que fotograma acceder.
Escape	Captura el fotograma actual y crea una ventana de imagen con él.

Tabla B.1: Teclas y acciones asociadas.

B.2 LISTAS DE FICHEROS IMAGEN

En la introducción ya hemos comentado los fundamentos de las listas de ficheros. A continuación entraremos más en profundidad, desarrollando el proceso típico utilizado para la obtención de una secuencia *VAL* a partir de una señal analógica de video, y utilizando las listas de ficheros como formato intermedio.

Comencemos observando el esquema de la figura B.2:

La fuente de información de video que aparece en el esquema, es una fuente de video analógico. En el caso del proyecto realizado se trata de un video semiprofesional (tipo Super VHS), cuya salida de video esta conectada a la tarjeta digitalizadora (capturadora de frames) del PC de trabajo. En este PC se dispone del software comercial ADOBE PREMIER ©5.0 que junto con los *drivers* adecuados, permite capturar señales de TV analógica convencional (PAL o SECAM) y almacenarlas en el disco duro sin perdida de ningún *frame* (sí bien se trata de un disco duro de alta velocidad, tipo SCSI). El resultado de este primer paso es una secuencia de video comprimido, fichero tipo *AVI*. Utilizando este mismo programa, es posible después exportar el *AVI* como un conjunto de ficheros imagen (uno por *frame*), que por facilidad elegimos sean de tipo *BMP*. Todos estos ficheros (en una secuencia grande pueden ser miles) ocupan mucha memoria, ya que el formato *BMP* no utiliza ninguna compresión.

La solución que hemos elegido consiste en convertir, mediante el software adecuado (hay incluso programas gratuitos en Internet), los ficheros *BMP* en ficheros *JPG* con compresión. Ahora, debido a la compresión, el ahorro de memoria es notable. Y son los ficheros *JPG* los que se conservan (lo más habitual es realizar copias de seguridad de ellos en CD-ROM) en el disco duro, borrando los ficheros intermedios: *AVI* y *BMP*. Cómo curiosidad decir que, el nombre de estos ficheros imagen suele ser el nombre de

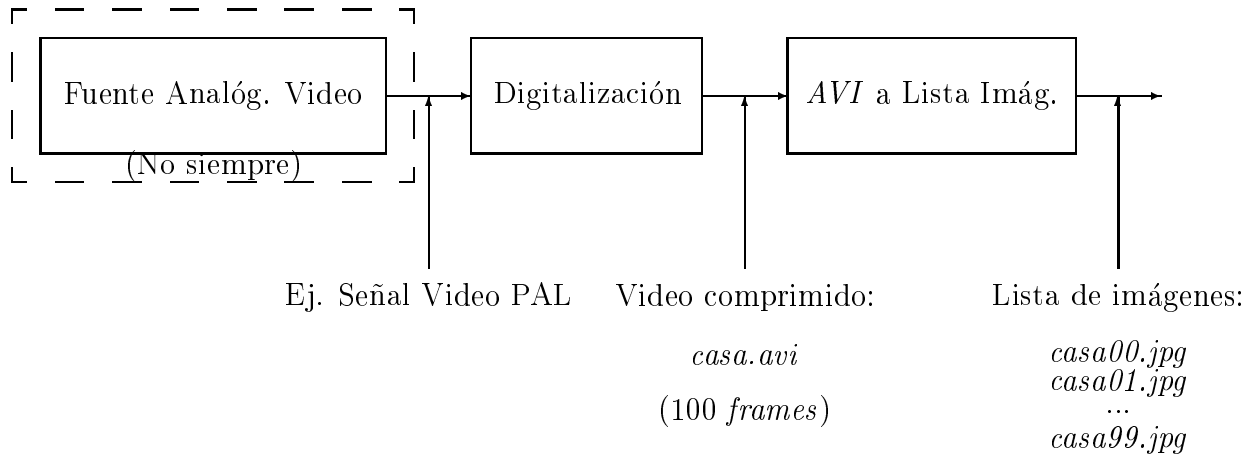


Figura B.2: Proceso de digitalización de una secuencia de video.

la secuencia *AVI*, añadiendo a cada uno de ellos un número correspondiente al número del *frame* que contiene. Como se verá más adelante, esto facilita la tarea de ordenarlos.

Pero como aparece en el esquema, no siempre partimos de una fuente analógica de video. En el proyecto también hemos trabajado con secuencias ya disponibles en formato digital comprimido, más concretamente secuencias *MPEG*. Resulta evidente que no será necesario pasar a formato analógico y luego realizar el proceso descrito. El programa *ADOBE PREMIER 5.0* también nos permite, partiendo de un fichero *MPG*, exportar a una lista de imágenes y después seguir los mismos pasos descritos anteriormente.

En adelante describiremos cómo a partir de este conjunto de imágenes comprimidas, generamos una secuencia *VAL* que podamos procesar después.

B.2.1 Los ficheros lista, *LST*

Como continuación del esquema de la figura B.2 fijémonos ahora en la figura B.3, que nos muestra los pasos hasta el final del proceso.

En primer lugar, es preciso generar un fichero de tipo *LST* (formato propio para el *PDIWIN32*). Se trata simplemente de un fichero ASCII con esta extensión, y cuyo contenido consiste en la lista (uno por línea) con el nombre de todos los ficheros imagen a incluir en la secuencia *VAL*. El nombre del fichero podrá ser relativo, es decir, estar formado sólo por el nombre propiamente dicho (incluyendo la extensión) o absoluto, indicando en este caso para cada fichero cuál es su ubicación (unidad y directorio donde

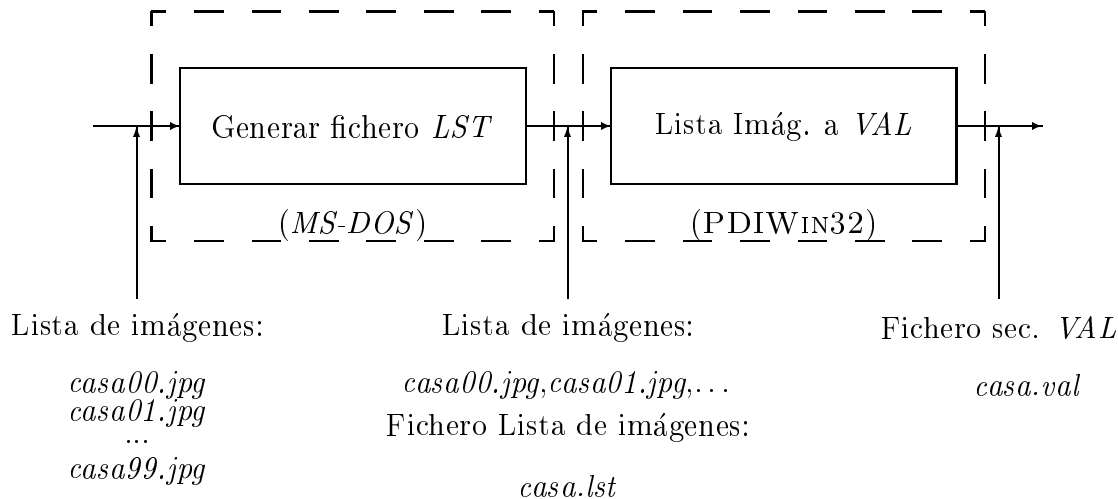


Figura B.3: Proceso de generar una secuencia VAL a partir de una lista de ficheros.

se encuentra), junto con el nombre. Posteriormente volveremos a este tema.

Un ejemplo del posible contenido de este fichero lo vemos en la figura B.4, donde los ficheros imágenes, de tipo *JPG*, están en el directorio `C:/SECUEN/CHAPLIN`.

La forma más fácil de generar este fichero “lista de ficheros” es utilizando una ventana al MS-DOS, y las opciones del comando `DIR`. A continuación mostramos un ejemplo de como generar el fichero lista *chaplin.lst* a partir de todos los ficheros imagen que van desde el *chapl000.jpg* al *chapl999.jpg* y estando dentro del directorio adecuado:

```
C:/SECUEN/CHAPLIN/ DIR chapl*.jpg /on /b > chaplin.lst
```

Simplemente se utilizan las opciones de ordenar por nombre y redireccionar la salida a un fichero.

B.2.2 Convertir una lista de ficheros imagen a un VAL

Como se puede observar en la figura B.3, el último paso se realiza dentro del PDIWIN32. Para ello, se ha programado una función que convierte una lista de ficheros imagen a un VAL: **Lista** → **VAL** y se encuentra en el menú **Archivo**.

En primer lugar, la función nos pedirá que introduzcamos el nombre del fichero *LST* que contiene la lista de ficheros imagen, y lo hará mediante un cuadro de dialogo cómo el de la figura B.5.

Como ya se indicó anteriormente, la lista puede contener bien nombres de fichero sin ruta (nombre relativo), bien nombres de ficheros con la ruta completa (nombre

```
C:/SECUEN/CHAPLIN/chapl000.jpg
C:/SECUEN/CHAPLIN/chapl001.jpg
C:/SECUEN/CHAPLIN/chapl002.jpg
C:/SECUEN/CHAPLIN/chapl003.jpg
      :
C:/SECUEN/CHAPLIN/chapl996.jpg
C:/SECUEN/CHAPLIN/chapl997.jpg
C:/SECUEN/CHAPLIN/chapl998.jpg
C:/SECUEN/CHAPLIN/chapl999.jpg
```

Figura B.4: Contenido del fichero *chaplin.lst*.

absoluto). En el primer caso se supondrá que los ficheros están en el directorio donde se encuentre el fichero de la lista.

Después, aparecera una ventana como la de la figura B.6. Nos permite seleccionar qué procesado previo se realizará antes de incluirlas en la secuencia. Las imágenes podrán incluirse completas o bien filtradas paso-bajo y diezmadas por 2, 4 u 8. La última opción nos ha resultado muy útil para nuestro proyecto, al trabajar siempre con secuencias diezmadas por 8.

Una vez elegida esta opción, nos presentará un cuadro de diálogo para dar nombre a la secuencia *VAL* que va ha crearse, figura B.7. A continuación comienza el proceso, mostrandonos una pantalla por si se quiere cancelar (indica el número del fotograma que se está insertando). Si todo finaliza bien, nos mostrará la secuencia creada en una nueva ventana de imagen.

Los ficheros de la lista serán de los tipos *TIF*, *BMP*, *GIF* o *JPEG*, pudiendo aparecer mezclados. El primer fichero de las lista fija el tipo (RGB o Grises) y el tamaño de la secuencia. Si este primer fichero no se encuentra, ni siquiera mostrará la ventana para elegir el diezmado. El resto de los ficheros, se incluirán únicamente si son del mismo tamaño. La función generará también un fichero con el mismo nombre que la secuencia y extensión *LOG*, con un resumen de las incidencias encontradas en el proceso de creación (ficheros origen, tamaño, imágenes no insertadas y razón, etc ...).

Si se está creando un fichero *VAL* en RGB, y una de las imágenes de la lista es de Grises, esta se convierte a color antes de ser incluida. Si se está creando un *VAL* de Grises y se encuentra un fichero en color, su luminancia es lo que se añade al *VAL*. Las

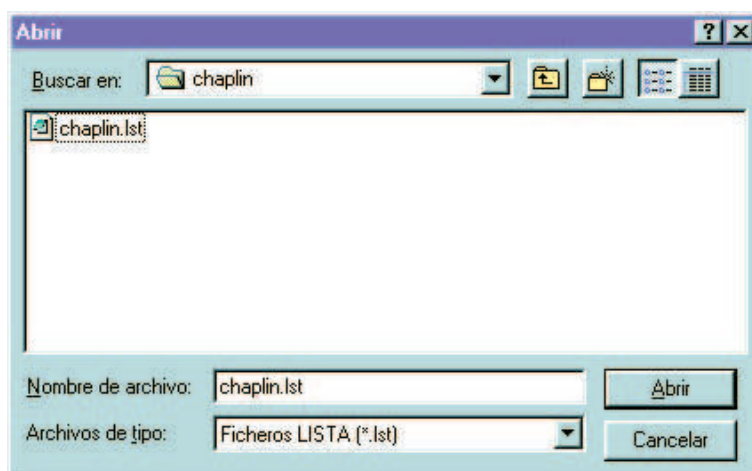


Figura B.5: Cuadro de diálogo para seleccionar el fichero *LST* a abrir

imágenes de paleta, se convierten a RGB o grises según sea su paleta antes de intentar ser incluidas.

B.2.3 Visualizar una lista de ficheros como si fuera un *VAL*

A continuación, vamos a comentar dos funciones relacionadas con las listas de ficheros imagen y con las secuencias *VAL*. Se incluye aquí su descripción por estar íntimamente relacionadas con el manejo de las listas, y haber sido de gran utilidad en la realización de este proyecto.

Se encuentran en el menú **Secuencias**, dentro del “POPUP” **Visualización especial**. La primera de estas funciones es **Visualizar Lista**, y consiste en la posibilidad de ver listas de ficheros imagen como si fuera un *VAL*. Cuando se selecciona esta función aparece un cuadro de diálogo para elegir el fichero *LST* a visualizar (similar al de la figura B.5), y a continuación muestra una ventana de imagen donde se visualizan cada uno de los ficheros imagen de la lista. Funciona con las mismas teclas de los *VAL*, véase tabla B.1.

Si existe un fichero, llamado de *puntos clave*, con la extensión *PTC* y el mismo nombre que el fichero *LST*, la reproducción se detendrá en los fotogramas indicados. Cuando se detiene, aparece una ventana, como la de la figura B.8, que es necesario aceptar para que continúe la reproducción. El contenido de este fichero *PTC* consiste simplemente en una lista de números enteros, que indican el número de *frame* donde debe detenerse. Sirve para ir haciendo pausas durante la visualización de la secuencia, en los puntos de transiciones, *keyframes*, etc. Esto ha sido muy utilizado en nuestro proyecto.

La otra función, dentro del mismo “POPUP”, se llama **Visualizar VAL** y sirve

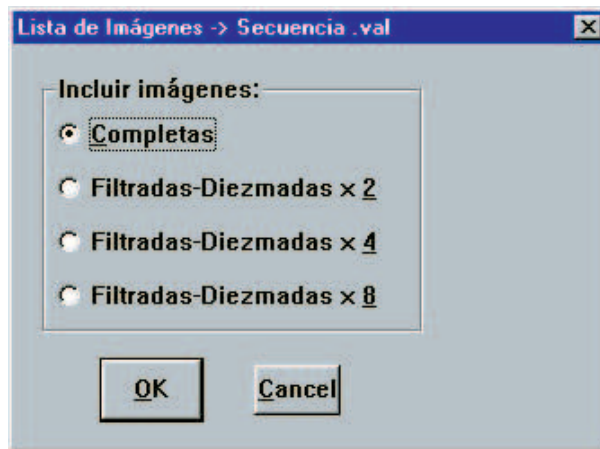


Figura B.6: Ventana para seleccionar cómo se han de incluir las imágenes

para visualizar una secuencia *VAL* utilizando la existencia de un fichero *PTC* con el mismo nombre. Será necesario tener activa la ventana con la secuencia, y la visualización la recorrerá desde el primer al último fotograma, parando en los *frames* “clave” (también mostrará la ventana de la figura B.8 para continuar).

B.2.4 Generación de una lista de ficheros a partir de un *VAL*

Para finalizar, y para una mayor integración en el uso las listas de imágenes, ha sido modificada la opción **Guardar como...** dentro del menú **Archivo** del programa PDIWIN32; de tal forma que cuando actúa sobre una secuencia *VAL* permite guardarla como un conjunto de ficheros imagen y además de forma automática, generará el fichero *LST* correspondiente.

Esto último, junto con la funciones comentadas en las secciones anteriores, permite una total interoperabilidad de las listas de imágenes y las secuencias *VAL* dentro del PDIWIN32.

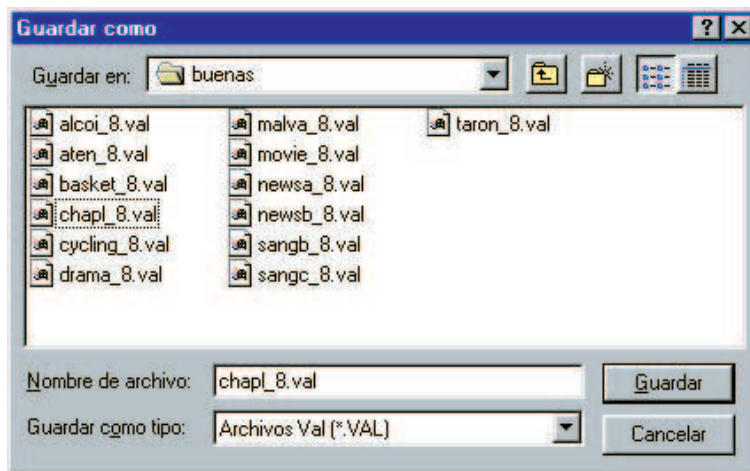


Figura B.7: Cuadro de diálogo para seleccionar el fichero *VAL* a crear



Figura B.8: Ventana para indicar que el fotograma actual es “clave”