Capítulo 4

Detección de transiciones graduales: Fundidos y Cortinillas

4.1 INTRODUCCIÓN

A lo largo del capítulo anterior, hemos desarrollado un método para la detección de transiciones abruptas o cortes. Por los resultados obtenidos, podemos decir que se trata de un método efectivo y eficiente.

Pero, el tema de las transiciones graduales es bastante diferente. Aunque si hay algunos métodos propuestos en la literatura, ninguno obtiene resultados mucho mejores que el resto.

El principal problema se encuentra en poder distinguir una transición gradual (ya sea un fundido o una cortinilla) de una zona con mucho movimiento (ya sea de la cámara o de los objetos de la imagen). Hemos mostrado en el capítulo anterior, que si el efecto se produce en muy pocos fotogramas (uno o dos intermedios), su localización puede llevarse a cabo mediante $d_{dc}(X,Y)$ y $d_{\rho}(X,Y)$. Sin embargo, la mayoría de los fundidos y cortinillas, duran entre medio y varios segundos. Así pues, se deberán buscar procedimientos específicos para estos efectos.

En este capítulo, vamos a desarrollar nuestros métodos para la localización de fundidos y cortinillas. Para cada uno de los dos casos, seguiremos el mismo proceso: definiremos qué estamos buscando, y a partir de conocer nuestro objetivo, intentaremos establecer una técnica que nos lleve a él. Después mostraremos como está implementado en el PDIWIN32 y su efectividad en algunas secuencias de prueba.

4.2 LOCALIZACIÓN DE FUNDIDOS

4.2.1 ¿Qué es un fundido?

Se trata del efecto de transición gradual por excelencia. Consiste en la mezcla, siguiendo una ley lineal, de dos imágenes de partida, a través de varias imágenes intermedias: Si A y B son las imágenes inicial y final, y el fundido se realiza con cuatro imágenes intermedias, $C_1, \ldots, C_4$, cada una de éstas se obtendrá:

$$C_i = \frac{5-i}{5} \times A + \frac{i}{5} \times B$$

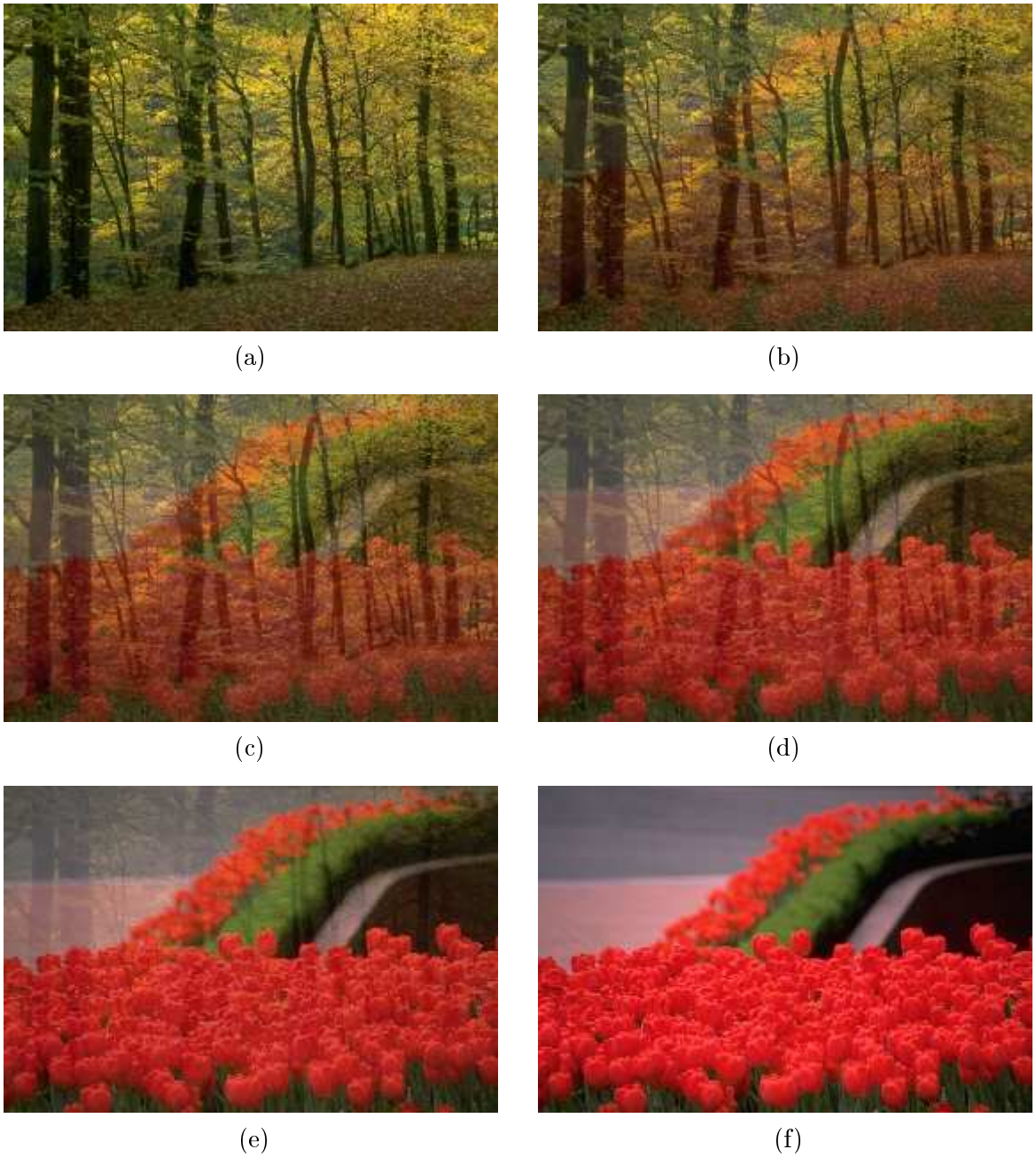
En la figura 4.1 aparece un fundido como el descrito, y que se ha realizado de forma manual mediante la anterior expresión [1].

El proceso de creación de un fundido, puede verse en la figura 4.2.

La opacidad de los filtros sucesivos sigue un proceso lineal, y al sumar el resultado de filtrar A y B se obtiene su combinación lineal. Las mesas de edición y montaje de vídeo permiten configurar la duración del fundido, generando de forma electrónica al valor de ponderación para cada imagen (mediante una rampa de tensión) y después haciendo la suma con amplificadores operacionales.

De igual forma, los programas de tratamiento digital de secuencias de vídeo también permiten mezclar secuencias diferentes, añadiendo todo tipo de efectos de transición entre ellas, como los fundidos. El método utilizado por estos programas se parece más al utilizado por nosotros para construir el ejemplo de la figura 4.1.

[1] El PDIWIN32 dispone de una función para la combinación lineal de dos imágenes.



(a) (b)

(c) (d)

(e) (f)

Figura 4.1: Ejemplo de fundido con cuatro fotogramas intermedios: (a)Fotograma inicial, A. (b)Fotograma intermedio, $C_1 = \frac{4}{5} \times A + \frac{1}{5} \times B$. (c)Fotograma intermedio, $C_2 = \frac{3}{5} \times A + \frac{2}{5} \times B$. (d)Fotograma intermedio, $C_3 = \frac{2}{5} \times A + \frac{3}{5} \times B$. (e)Fotograma intermedio, $C_4 = \frac{1}{5} \times A + \frac{4}{5} \times B$. (f)Fotograma final.

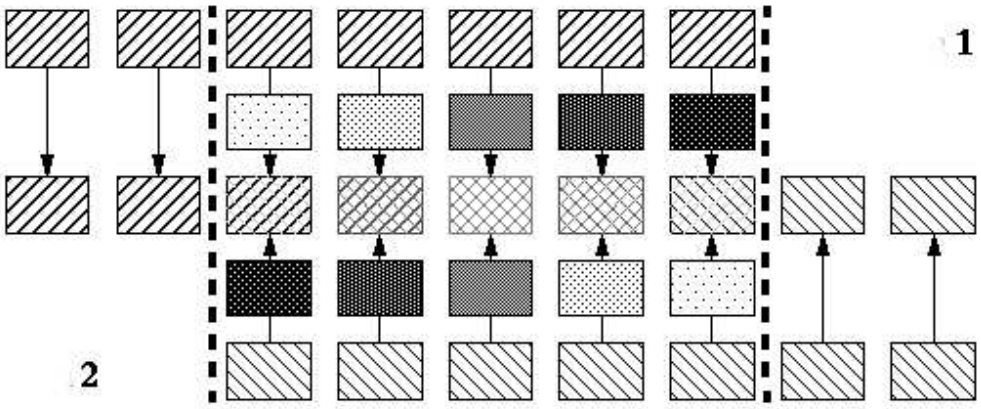


Figura 4.2: Proceso de obtención del fundido de dos fuentes vídeo 1 y 2.

4.2.2 Método de detección de fundidos

Sería interesante conocer alguna característica que tengan los pixels de los fotogramas de un fundido, y que nos permita localizarlos.

En primer lugar, nos basaremos en la aproximación de Alattar, descrita en [19], y que usa la varianza como la indicación para detectar fundidos.

Asumamos que $f_1(t)$ y $f_2(t)$ son dos secuencias ergódicas con una varianza de intensidad σ_1^2 y σ_2^2, respectivamente. Tenemos que la intensidad de $f_1(t)$ decrece linealmente a 0 (*fade-out*), y $f_2(t)$ se incrementa linealmente de 0 a un valor "normal" (*fade-in*). La región del fundido es la suma de $f_1(t)$ y $f_2(t)$ durante el fundido ($t_1 \leq t \leq t_2$, donde t_1 es el fotograma de comienzo y t_2 es el de fin). Con todo esto, una expresión para los fotogramas del fundido sería:

$$f(t) = f_1(t)[1 - \alpha(t)] + f_2(t)\alpha(t)$$

donde $\alpha(t) = \frac{t-t_1}{t_2-t_1}$, se incrementa de 0 a 100%. La varianza de $f(t)$ en la región del fundido es:

$$\sigma^2(t) = (\sigma_1^2 + \sigma_1^2)\alpha^2(t) - 2\sigma_1^2\alpha(t) + \sigma_1^2.$$

Este es el caso ideal, cuando $f_1(t)$ y $f_2(t)$ son ergódicos con σ_1^2 y σ_2^2. De forma gráfica, tal y como se puede ver en la figura 4.3, la varianza tendrá una forma parabólica. En secuencias reales con movimientos, la varianza de $f_1(t)$ y $f_2(t)$ puede no ser constante, pero se puede demostrar que en la región del fundido la forma de la varianza se aproxima al caso ideal.

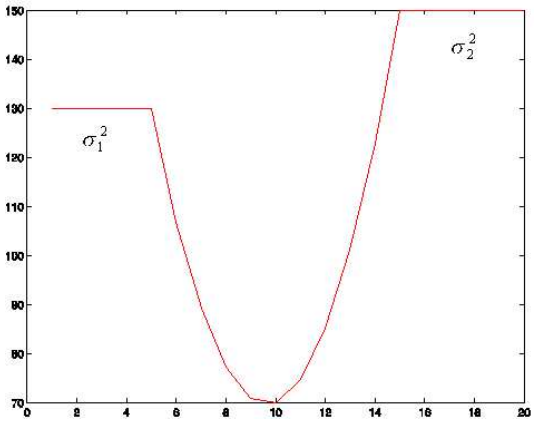


Figura 4.3: Curva de la varianza en un fundido.

Sin embargo, la varianza es una v.a. y existe la posibilidad de que se produzcan variaciones de tipo parábolico cuando no hay un fundido. En la figura 4.4, aparece la varianza para la secuencia *newsb_8.val*. Esta secuencia incluye un fundido que se ha

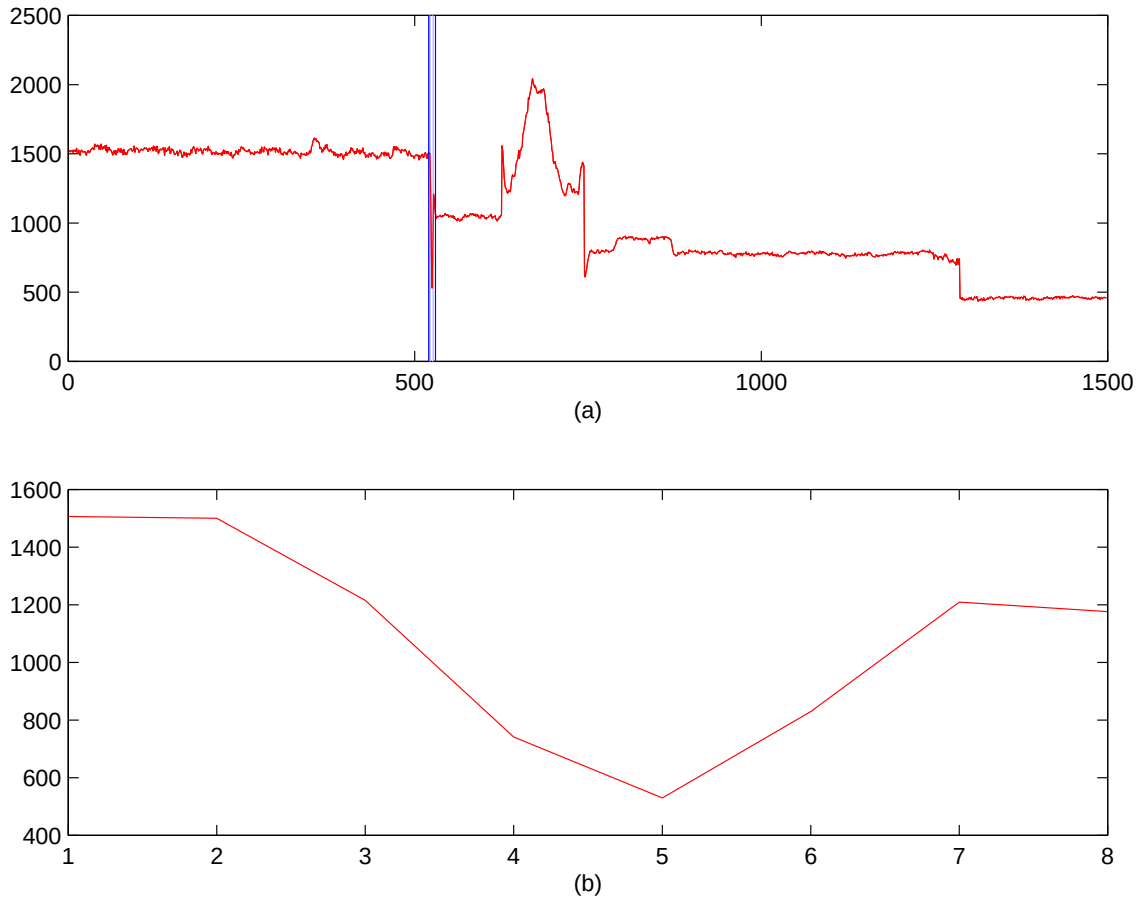


Figura 4.4: Varianza de la secuencia *newsb_8.val*: (a)Varianza para todos los fotogramas. El fundido esta localizado entre las líneas verticales. (b)Varianza en la región del fundido.

marcado y ampliado para poder observar su forma. También se puede observar que existen regiones, sin fundido, donde la variación se parece a una parábola.

Añadamos a esto la dificultad de localizar zonas de tipo parábola en una señal unidimensional, sin tener otra referencia. Por eso, se hace necesario encontrar una métrica que al menos nos localice las regiones candidatas a fundido, y después comprobar en ellas si la varianza sigue la ley parabólica.

La métrica que hemos definido, $d_\rho^{fundidos}$, se parece bastante a la "correlación" utilizada en la detección de cortes, y parte de la siguiente hipótesis:

> *"La luminosidad de los pixels, en los fotogramas de un fundido, seguirá una variación lineal y monótona".*

Veamos en un ejemplo qué queremos decir con esto. Ya hemos usado anteriormente la secuencia *newsb_8.val*, que tiene un fundido (de 5 fotogramas). Comenzaremos con tomar una subsecuencia de 50 fotogramas, que incluya el fundido en su zona central. En la figura 4.5 se incluyen los cortes ortogonales de esta subsecuencia, tomados para un pixel del centro de la imagen. Estos cortes, consisten en una imagen construida con la fila (para el corte horizontal) o la columna (para el vertical) del pixel seleccionado, para cada una de las imágenes de la secuencia.

(a) (b)

Figura 4.5: Cortes ortogonales de una secuencia, para el pixel (i, j): (a)Corte Horizontal: fila i - tiempo. (b)Corte Vertical: tiempo - columna j.

Si después realizamos los cortes ortogonales a una de estás imágenes, por ejemplo a la imagen del corte horizontal (para un pixel de las primeras imágenes y centrado horizontalmente), se obtendría el resultado de la figura 4.6.

Aparecen representados los cortes para la componente R, ya que los de las otras dos son muy semejantes. Centrandonos en el corte vertical, se observa la variación lineal de la luminosidad del pixel en los fotogramas del fundido, tal y como suponíamos en

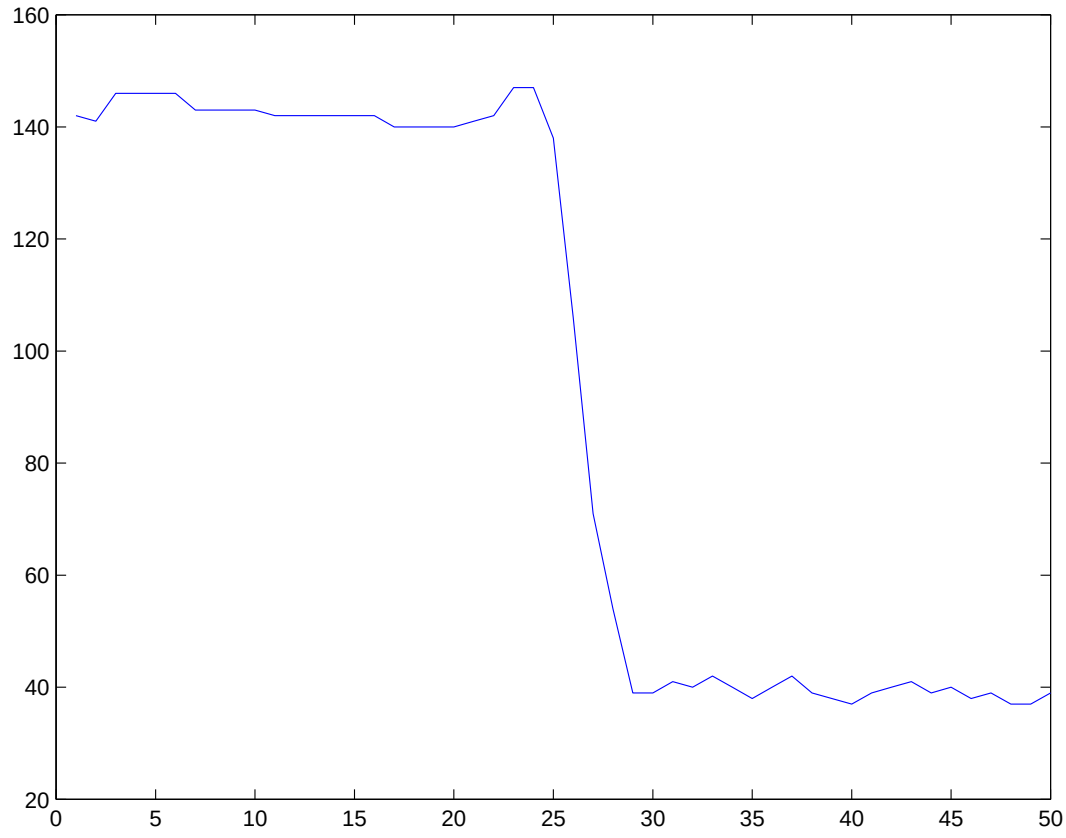


Figura 4.6: Corte vertical (para la componente R) de la imagen corte horizontal de una secuencia.

la hipótesis de partida. Se puede comprobar, que este efecto se produce para todos los pixels de la imagen.

Una vez verificada esta tendencia lineal, podemos definir la métrica para la localización de fundidos que se basará en ella. Si x_{ij}^{n-1}, x_{ij}^{n} y x_{ij}^{n+1} son respectivamente el valor del pixel (i,j) de la imágenes $X_{n-1} = \{x_{ij}^{n-1}\}$, $X_n = \{x_{ij}^{n}\}$, y $X_{n+1} = \{x_{ij}^{n+1}\}$ (que a su vez, se corresponden al fotograma $n-1$, n y $n+1$), y se definen la diferencias

$$d_{1_{ij}}^{n} = (x_{ij}^{n} - x_{ij}^{n-1})$$

$$d_{2_{ij}}^{n} = (x_{ij}^{n+1} - x_{ij}^{n})$$

se considerará que

$$\rho_{i,j}^{fundidos} = \begin{cases} 1 & \text{Si } |d_{1_{ij}}^{n}| > 2 \text{ y } |d_{2_{ij}}^{n}| > 2, \text{ y además } sign(d_{1_{ij}}^{n}) = sign(d_{2_{ij}}^{n}) \\ -1 & \text{Si } |d_{1_{ij}}^{n}| > 2 \text{ y } |d_{2_{ij}}^{n}| > 2, \text{ y además } sign(d_{1_{ij}}^{n}) \neq sign(d_{2_{ij}}^{n}) \\ 0 & \text{Otro caso} \end{cases}$$

y a partir de esta medida por pixel, se define la siguiente métrica para cada fotograma:

$$d_{\rho}^{fundidos}(X_n) = \frac{\sum_{i,j} \rho_{i,j}^{fundidos}}{anch \times alt} \tag{4.1}$$

Si tenemos una secuencia de N fotogramas ($n = 0, 1, \ldots, N-1$), el proceso se realizará a partir del segundo y hasta el penúltimo, definiendose que $d_{\rho}^{fundidos}(X_0) = d_{\rho}^{fundidos}(X_{N-1}) = 0$.

Lo que hace esta métrica para cada fotograma, n, es compararlo con el anterior, $n-1$, y el siguiente, $n+1$. Si la variación entre n y $n-1$, y $n+1$ y n tiene el mismo signo para un pixel, nos indicará una variación lineal en la luminancia en ese pixel. Cuando este fenómeno se produce para la mayoría de los pixel de una imagen, obtendremos un valor elevado de $d_{\rho}^{fundidos}(X_n)$. Durante todos los fotogramas de un fundido se deberán obtener valores altos de $d_{\rho}^{fundidos}(X_n)$.

Al igual que cuando definimos $d_{\rho}(X,Y)$, aquí también se fija un umbral de 2 para no tener en cuenta las variaciones aleatorias (en forma de ruido).

En la figura 4.7, se muestra el resultado de calcular $d_{\rho}^{fundidos}(X_n)$ para la secuencia *newsb_8.val*. Se observa claramente el pico producido por el fundido (fotogramas $522-525$). Fijando un umbral, $u_{\rho_{fundidos}} = 0.3$, capturaríamos este fundido, pero también una falsa alarma (fotograma 627).

Las falsas alarmas producidas por $d_{\rho}^{fundidos}$ se deben al movimiento de objetos grandes (ocupan muchos pixels de la imagen) en imágenes sucesivas. El método que hemos creado para la localización de fundidos combinará la información obtenida con $d_{\rho}^{fundidos}$ y la información de la varianza que describimos anteriormente. Los pasos que seguiremos serán:

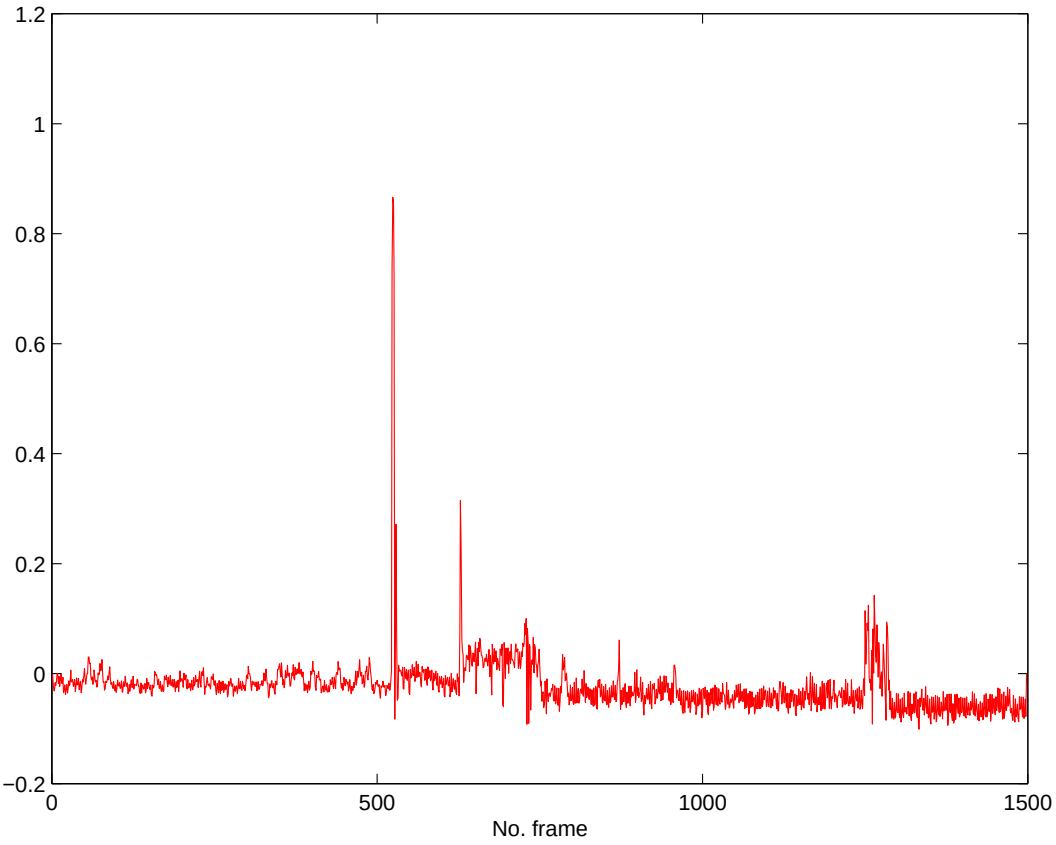


Figura 4.7: Resultado de calcular $d_\rho^{fundidos}(X_n)$ para la secuencia *newsb_8.val*.

1. Calcular $d_\rho^{fundidos}$ y σ^2 para cada fotograma de la secuencia.
2. Fijado un umbral, $u_{\rho_{fundidos}}$, todos los fotogramas que lo superen serán candidatos a pertenecer a un fundido.
3. Se tomará la varianza en estos candidatos a fundido y se comparará con la que debería obtenerse de forma ideal (según la expresión descrita anteriomente). Si la varianza real y la estimada difieren menos de un valor, que denominaremos umbral de varianza, $u_{\sigma^2_{fundidos}}$, se concluirá que el candidato a fundido lo es realmente.

En la sección siguiente, desarrollaremos cómo se ha programado la busqueda de fundidos en el PDIWIN32 utilizando el procedimiento que acabamos de describir.

4.2.3 Localización de los fundidos con el PDIWin32

En primer lugar y como información básica para la localización de los fundidos, deberemos calcular $d_\rho^{fundidos}$ para cada fotograma de la secuencia. La entrada de menú

que se encarga de ello puede verse en la figura 4.8. En el mismo "POPUP" hay otra

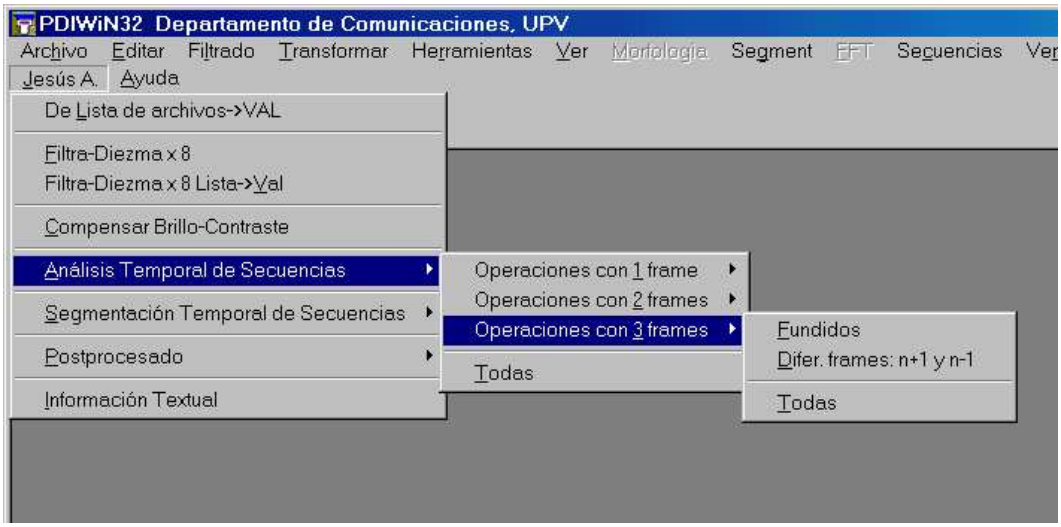


Figura 4.8: Entrada de menú que realiza el cálculo de $d_\rho^{fundidos}$ para una secuencia.

entrada de menú: **Difer. frames: n+1 y n-1**, para calcular el módulo de la diferencia para los fotogramas $n+1$ y $n-1$. Aunque no se ha utilizado en este proyecto, nos ha parecido interesante incluirla para posteriores aplicaciones de Análisis Temporal de Secuencias.

Cuando la ventana activa sea una secuencia *VAL* [2] y seleccionemos **Fundidos**, nos calculará el valor de $d_\rho^{fundidos}$ para cada fotograma.

De igual forma que para las otras funciones del **Análisis Temporal de Secuencias**, el proceso puede cancelarse antes de finalizar. Para ello, mientras procesa la secuencia, aparece una ventana que indica el número de fotogramas que tiene la secuencia, y cual se esta procesando en ese instante. Al terminar, si todo ha ido correctamente, aparece una ventana que nos mostrará el nombre del fichero de texto donde se han guardado los valores de $d_\rho^{fundidos}$: *fundido.txt*.

Para la localización de los fundidos se ha situado una función, **Detección de fundidos**, tal y como se ve en la figura 4.9.

Los datos necesarios para los fundidos se encuentran en los ficheros *fundido.txt* y *varianza.txt*. Por eso, lo primero que hace esta función es comprobar que se encuentran en el directorio de trabajo. Si no se encuentra alguno de los dos, nos mostrará un mensaje de error.

Si tenemos los dos ficheros necesarios, lo siguiente será mostrarnos la ventana de configuración de parámetros, figura 4.10. Podremos de esta forma fijar los umbrales: $u_{\rho_{fundidos}}$ y $u_{\sigma^2_{fundidos}}$.

[2] En caso contrario, y como ya comentamos, nos mostrará el error correspondiente.

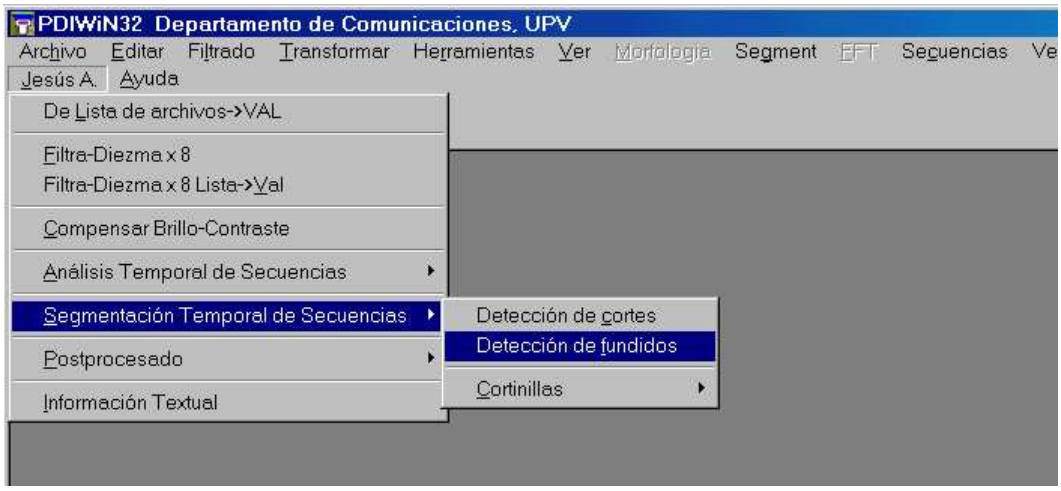


Figura 4.9: Entrada de menú para detectar los fundidos de una secuencia.

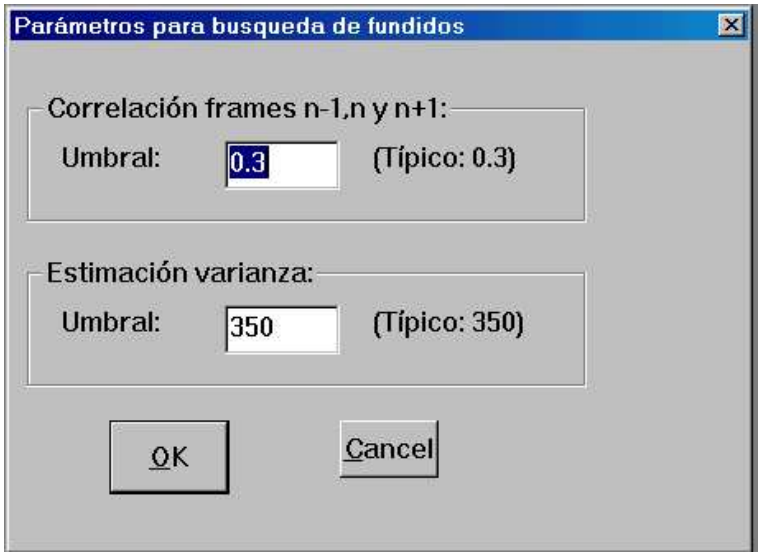


Figura 4.10: Ventana para configurar los parámetros del proceso de detección de fundidos.

Utilizando estos parámetros, realiza todo el proceso descrito en el apartado anterior. Cuando finaliza, nos muestra los ficheros donde guarda los resultados. Estos ficheros son: *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 ":".

El mismo proceso realizado en el PDIWIN32, también lo hemos programado para poder ser realizado en el programa MATLAB, partiendo de la información de los ficheros *fundido.txt* y *varianza.txt*.

4.2.4 Funciones programadas en el PDIWin32 para la localización de los fundidos

P8Correla3Frames

- □ **Sinopsis:** Para el cálculo de $d_\rho^{fundidos}$. Funcionará para imágenes de 8 o 24 bits.
- □ **Formato:** `P8Correla3Frames(imgdes *desimg1,imgdes *desimg2,imgdes *desimg3,float *corre_3,float *corre3num,int Frame,int numFrames);`
  - `imgdes *desimg1`: Puntero a la imagen del fotograma $n-1$.
  - `imgdes *desimg2`: Puntero a la imagen del fotograma n.
  - `imgdes *desimg3`: Puntero a la imagen del fotograma $n+1$.
  - `float *corre_3`: Variable donde se almacenará el resultado.
  - `float *corre3num`: Otro valor que no ha sido finalmente usado.
  - `int Frame`: Fotograma n que se esta procesando.
  - `int numFrames`: Número de fotogramas que tiene la secuencia. Estos dos últimos parámetros se utilizan solo para mostrar la ventana que indica, mientras se esta calculando, el punto donde se encuentra el proceso.

P8VarianzaFundidos

- □ **Sinopsis:** Sirve para calcular el valor teórico de la varianza del fotograma de un fundido [3] (para la aplicación del criterio de la varianza sobre los hipotéticos fundidos detectados con $d_\rho^{fundidos}$). El valor (en coma flotante) devuelto será $\sigma^2(t)$.

[3] *Implementa* la expresión: $\sigma^2(t) = (\sigma_1^2 + \sigma_1^2)\alpha^2(t) - 2\sigma_1^2\alpha(t) + \sigma_1^2$.

- **Formato:** `float P8VarianzaFundidos(int t1,int t2,int t,float vari1,float vari2);`
  - `int t1`: Número del primer fotograma del posible fundido.
  - `int t2`: Número del último fotograma del posible fundido.
  - `int t`: Número del fotograma del que calcular la varianza teórica.
  - `float vari1`: Varianza del primer fotograma del posible fundido.
  - `float vari2`: Varianza del último fotograma del posible fundido.

P8FundidosOK

- **Sinopsis:** Nos devolverá 1 cuando un candidato a fundido lo sea, y 0 en caso contrario (para la aplicación del criterio de la varianza sobre los hipotéticos fundidos detectados con $d_\rho^{fundidos}$).
- **Formato:** `int P8FundidosOK(int ind_min,int ind_max,float error,int numElementos);`
  - `int ind_min`: Número del fotograma de inicio del fundido.
  - `int ind_max`: Número del fotograma de fin del fundido
  - `float error`: Es el valor de $u_{\sigma^2_{fundidos}}$ fijado previamente.
  - `int numElementos`: Número de fotogramas que tiene la secuencia

4.3 LOCALIZACIÓN DE CORTINILLAS

4.3.1 ¿Qué es una cortinilla?

Una cortinilla [4] es un efecto de transición entre dos planos, que consiste en el desplazamiento espacial de una imagen sobre otra de forma gradual, a lo largo de varios fotogramas. De forma habitual, al igual que para los fundidos, el desplazamiento espacial se produce de una forma lineal.

En la figura 4.13, se muestra el método utilizado para realizar una cortinilla. En este caso, la cortinilla es vertical hacia la izquierda (la imagen se desplazará sobre toda la dimensión vertical, avanzando hacia la izquierda).

El proceso consta de las siguientes etapas:

- Se corta el final del plano que debe desaparecer y el principio del plano que debe aparecer.
- Se hace una tira negativa de los dos trozos de película obtenidos.
- Se hace una nueva tira positiva de los dos negativos de la misma película aplicando un filtro delante de las imágenes.
- Se une la película resultante entre los dos planos a unir.

Al igual que los fundidos, las cortinillas o efectos de incrustaciones, se utilizan para realizar transiciones graduales entre dos planos, utilizando fotogramas intermedios.

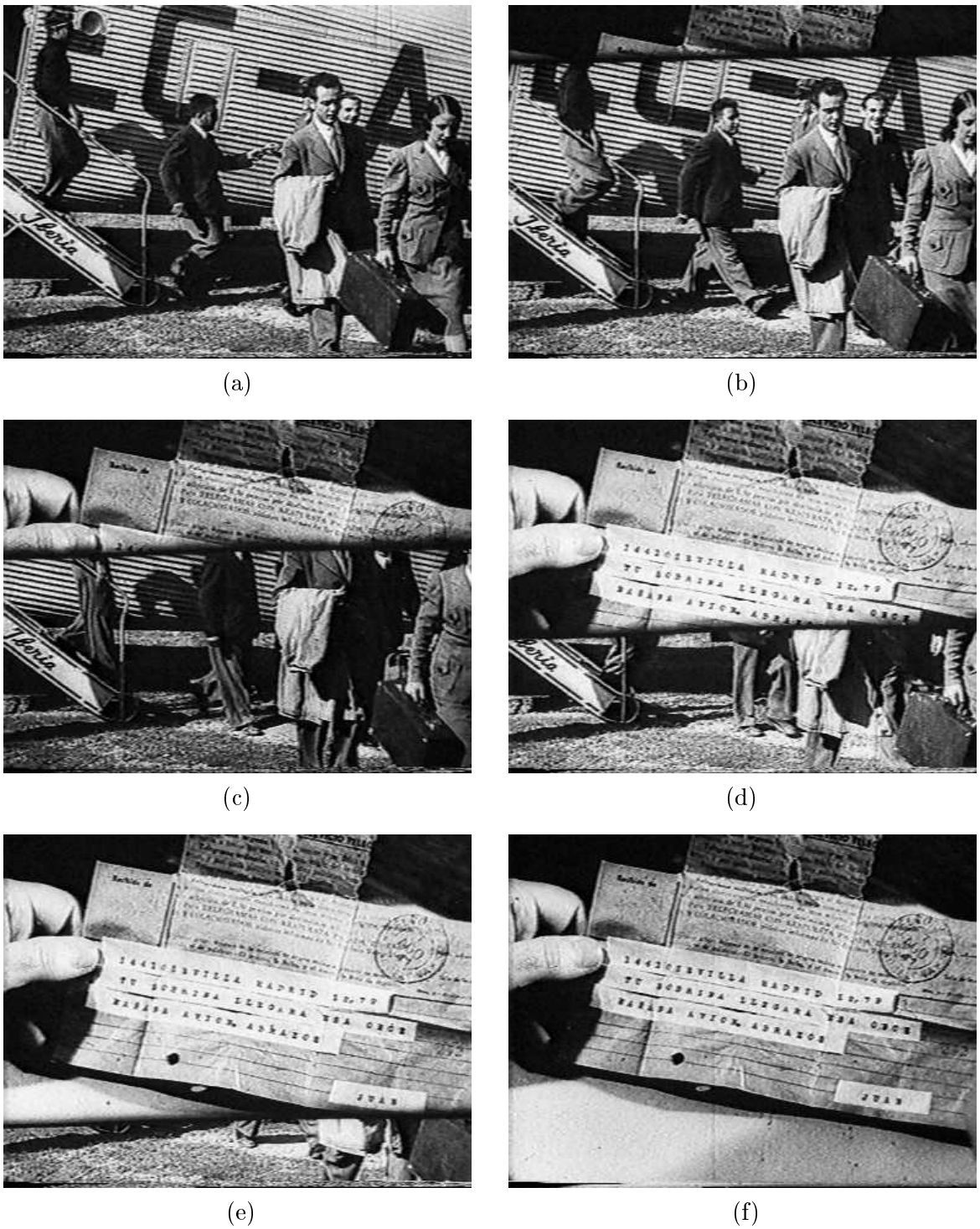
La región de opacidad en los filtros sucesivos aumenta o disminuye siguiendo un proceso lineal, y al sumar el resultado de filtrar A y B se obtiene el efecto de la cortinilla.

De igual forma, los programas de tratamiento digital de secuencias de vídeo también permiten mezclar secuencias diferentes, añadiendo todo tipo de efectos de transición entre ellas, como las cortinillas. Los métodos utilizado por estos programas se parecen mucho al que acabamos de describir (los filtros serán mascaras digitales).

[4] En francés, *volet*, y en inglés, *wipe*.

(a) (b) (c) (d) (e) (f)

Figura 4.11: Ejemplo de cortinilla vertical (desplazandose hacia la derecha), con cuatro fotogramas intermedios.



(a) (b)

(c) (d)

(e) (f)

Figura 4.12: Ejemplo de cortinilla horizontal (desplazandose hacia abajo), con cuatro fotogramas intermedios.

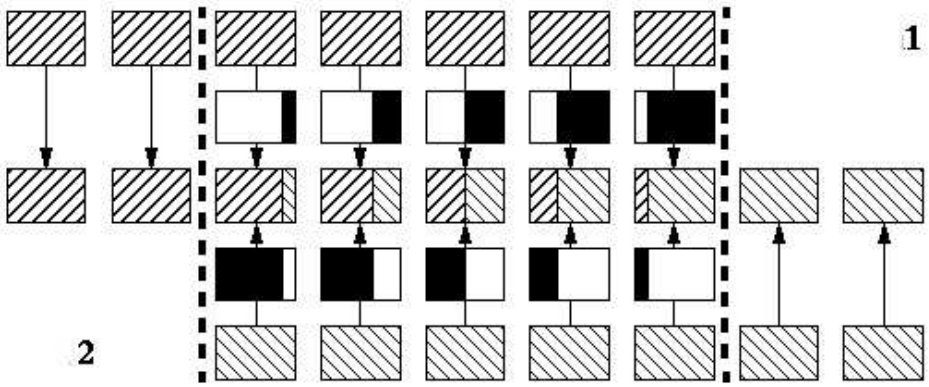
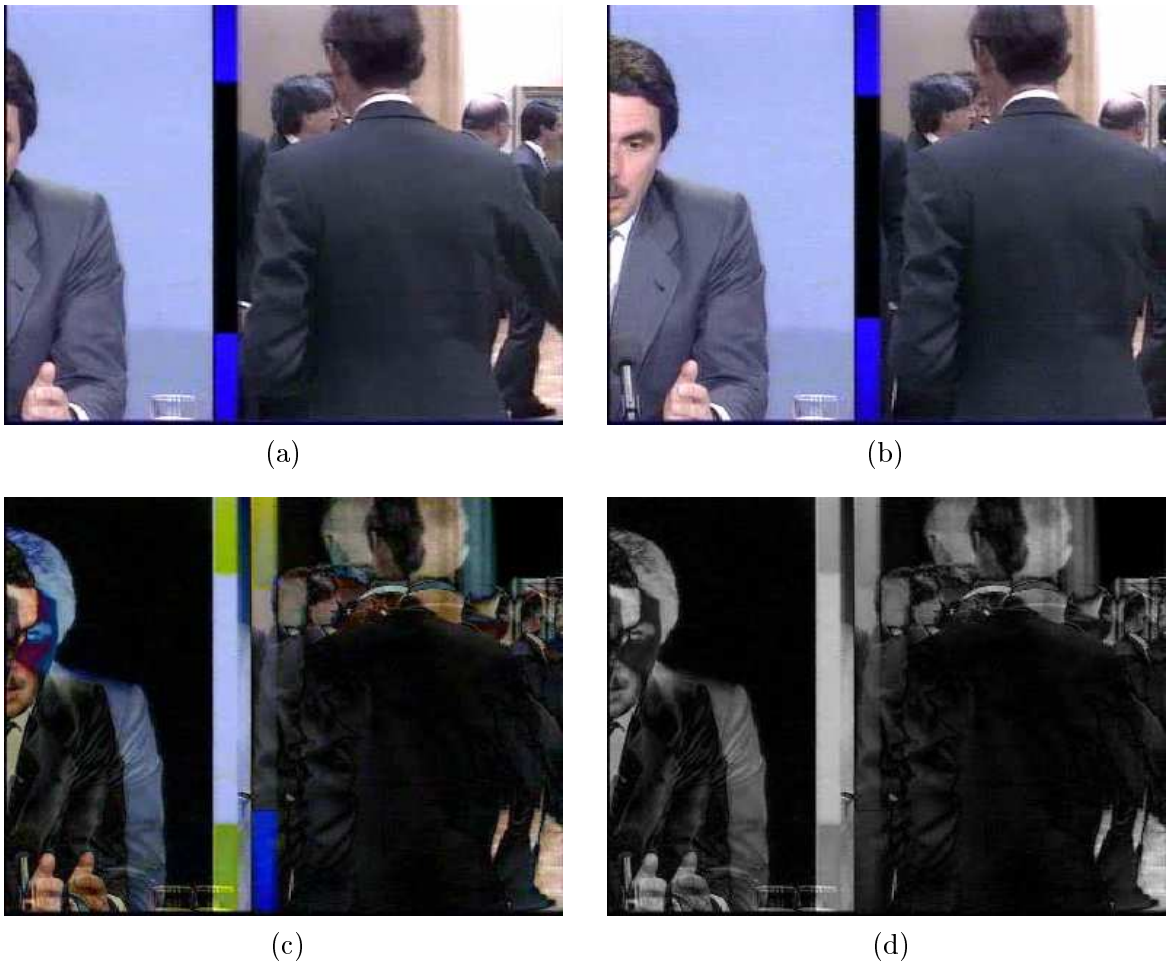


Figura 4.13: Proceso de realización de una cortinilla con dos fuentes vídeo 1 y 2.

Sería interesante conocer alguna característica que tengan los pixels de los fotogramas de una cortinilla, y que nos permita localizarlas.



(a) (b)

(c) (d)

Figura 4.14: Dos fotogramas de la cortinilla vertical de *newsa_8.val*: (a)Fotograma 1645. (b)Fotograma 1646. (c)Imagen diferencia de los fotogramas 1645 y 1646. (d)Imagen anterior convertida a grises.

Fijemonos en la figura 4.14: se muestran dos fotogramas consecutivos de una cortinilla vertical, y la imagen diferencia entre ellos. Al realizarse el desplazamiento espacial (anchura x pixels), de una imagen sobre la otra, la imagen diferencia tendrá una región de anchura x con valores altos (zonas claras) a lo largo de toda la altura, y esto se observa sobre todo en la imagen de grises. Esta cortinilla también tiene una carácterística especial: se produce un desplazamiento del contenido de la imagen según avanzan los fotogramas. Por esto último, el resto de la imagen diferencia también podrá tener otras regiones claras, si bien éstas no ocuparán toda la imagen en la dimensión altura. Esto último será el principio utilizado para la localización de las cortinillas.

Llamaremos $\phi[m,n]$ a la imagen digital diferencia, tal que $1 \leq m \leq anch$ (columnas) y $1 \leq n \leq alt$ (filas), siendo $anch$ y alt el número de pixels de ancho y alto respectivamente. Tomemos ahora esta imagen y realicemos su proyección por columnas normalizada. Para la columna j tendremos que:

$$proyec_{col}[j] = \frac{1}{alt} \sum_{i=0}^{alt} \phi[j,i] \tag{4.2}$$

Observemos ahora el valor de esta proyección para cada una de las columnas de la imagen, figura 4.15. Es fácil de identificar el pico máximo como parte de la región

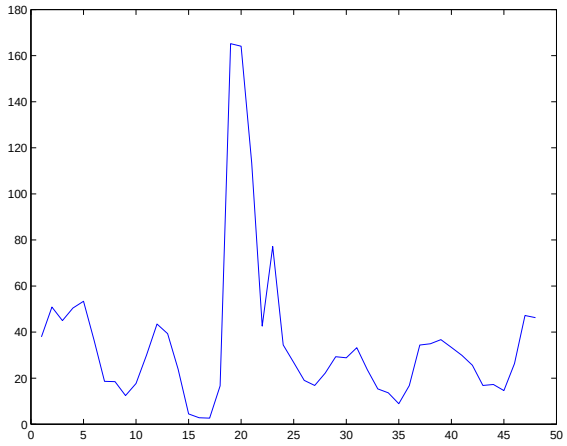


Figura 4.15: Proyección por columnas de la imagen diferencia entre dos fotogramas de una cortinilla vertical.

de anchura x que se ha desplazado la cortinilla al pasar de un fotograma a otro. Sin embargo, será más interesante quedarnos solo con este máximo absoluto, eliminando los máximos regionales, y esto se consigue con la *reconstrucción geodésica* [5]. Se trata de un operador morfológico que utilizando una señal, *marcador*, consistente en una delta unidad en la posición del máximo de la señal de partida, *referencia*, nos permitirá eliminar todos los máximos relativos, figura 4.16.

Tenemos por tanto un máximo, que nos indica la posición espacial de la cortinilla en la imagen diferencia. Al ir desplazandose la cortinilla del fotograma k al fotograma $k+1$, el máximo de la imagen diferencia entre $k+1$ y k también se irá desplazando espacialmente. En la figura 4.17, se muestra este desplazamiento para tres imágenes diferencia consecutivas.

Al realizar el proceso de proyección por columnas, sobre la imagen diferencia de dos fotogramas que no pertenecen a una cortinilla vertical, y realizar después la recons-

[5]Para conocer más en profundidad la definición de este operador, remitimos al lector al apéndice de Morfología Matemática.

trucción, los máximos que se obtiene serán mucho más pequeños que cuando si haya cortinilla (la cortinilla ocupa toda la columna).

Todo el proceso descrito hasta ahora para cortinillas verticales, es directamente trasladable a la cortinillas horizontales. En este caso será necesario realizar una proyección por filas. Si llamamos $\phi[m,n]$ a la imagen digital diferencia, tal que $1 \leq m \leq anch$ (columnas) y $1 \leq n \leq alt$ (filas), siendo $anch$ y alt el número de pixels de ancho y alto respectivamente. Su proyección por filas, para la fila j tendremos que:

$$proyec_{fil}[j] = \frac{1}{anch} \sum_{i=0}^{anch} \phi[i,j] \tag{4.3}$$

Si calculamos el valor de esta proyección para cada una de las filas de la imagen diferencia entre dos fotogramas de una cortinilla horizontal y desués realizamos la reconstrucción, el pico máximo se encontrará en la región de anchura y que se ha desplazado la cortinilla al pasar de un fotograma a otro. Al realizar el proceso de proyección por filas, sobre la imagen diferencia de dos fotogramas que no pertenecen a una cortinilla horizontal, y realizar después la reconstrucción, los máximos que se obtiene serán mucho más pequeños que cuando si haya cortinilla (la cortinilla ocupa toda la fila).

A continuación mostraremos la forma de utilizar estos máximos deslizantes, para poder localizar cortinillas.

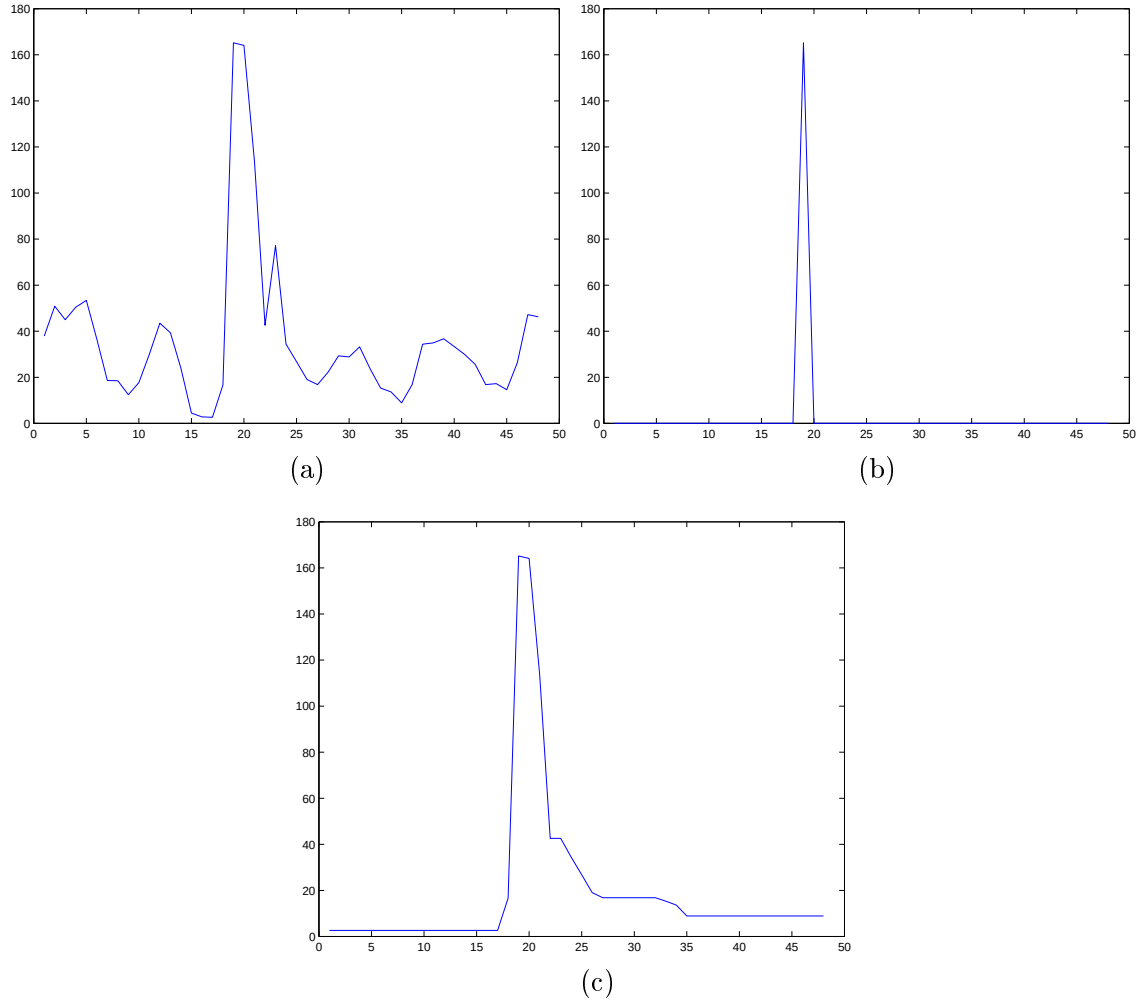


Figura 4.16: Reconstrucción geodésica: (a)Proyección por columnas de la imagen diferencia entre dos fotogramas de una cortinilla vertical. Es la señal de referencia. (b)Señal marcador. (c)Resultado de la reconstrucción.

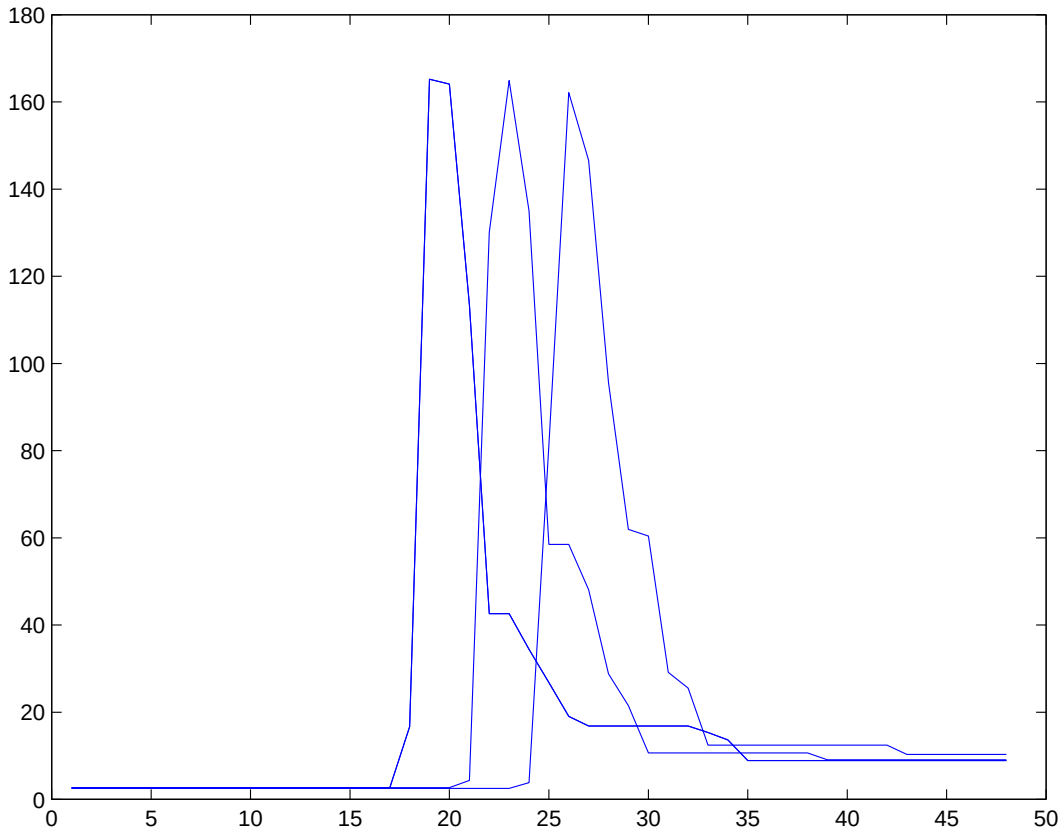


Figura 4.17: Proyecciones por columnas de tres imágenes diferencia entre fotogramas de una cortinilla vertical.

4.3.2 Imágenes "tira" para la localización de cortinillas

Se puede ir concretando algunos aspectos que ya deben quedar claros:

- Localización de cortinillas horizontales → Realizar la proyección por filas de la diferencia entre fotogramas consecutivos.
- Localización de cortinillas verticales → Realizar la proyección por columnas de la diferencia entre fotogramas consecutivos.

Después de tener la proyección de cada imagen diferencia, se debe realizar la reconstrucción geodésica utilizando el máximo como marcador.

Una forma de visualizar las sucesivas proyecciones es mediante una imagen de grises. Los valores de los pixels de cada fila de la imagen se corresponderán con los valores de la reconstrucción de la proyección para cada imagen diferencia (normalizados a 256 niveles de gris). La fila superior de la imagen se corresponderá con la primera proyección.

Si tenemos una secuencia de N fotogramas, y dimensión en pixels $anch$ y alt, la imagen de grises tendrá un tamaño de $N-1$ pixels de altura y $anch$ pixels de anchura para la localización de cortinillas verticales, y $N-1$ pixels de altura y alt pixels de anchura para la localización de cortinillas horizontales.

Ya que en este proyecto trabajamos con secuencias diezmadas (anchura y altura de imagen pequeña), y las secuencias tendrán cientos, o incluso miles de fotogramas, el aspecto de estas imágenes será parecido a una tiras oscuras tan largas como sea la secuencia. De ahí el nombre de imagen "tira".

Para poder observar un ejemplo, tomaremos una subsecuencia de *newsa_8.val* (del fotograma 1350 al 1700) que incluya la cortinilla vertical. Calcularemos las proyecciones por columnas, después la reconstrucción y contruiremos finalmente la imagen descrita, figura 4.18.

En (a) aparece la imagen de grises tal cual queda después de normalizar las reconstrucciones. En ella, se puede observar algunos fenómenos característicos:

- La cortinilla se corresponderá con una línea oblicua, cuya pendiente indicará la longitud temporal de la cortinilla. Esta línea recta se origina al desplazarse espacialmente el máximo de la reconstrucción, y estará formada por pequeñas líneas horizontales de tamaño: número de pixels que avanza la cortinilla en fotogramas sucesivos.
- Las transiciones abruptas o cortes, se corresponden con líneas horizontales. La explicación también es sencilla: la imagen diferencia de un corte tendrá valores muy altos en todos sus pixels y al realizar la reconstrucción, el máximo tendrá un valor elevado aunque su contraste respecto del resto de elementos de la proyección será pequeño.

- Las zonas con mucho movimiento originarán zonas grises de forma irregular. Cuanto más importante sea el movimiento, más clara será la zona correspondiente.

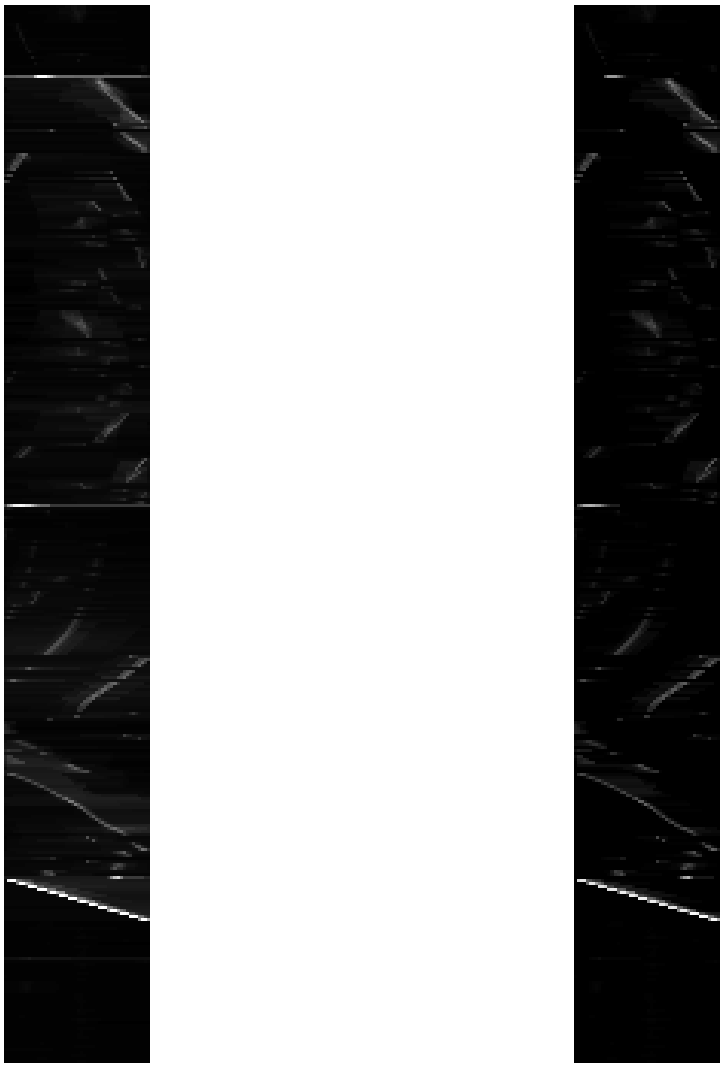
Nuestro objetivo, será eliminar todo aquello que no sean las lineas correspondientes a las cortinillas. El programa PDIWIN32 dispone de herramientas de tratamiento de imágenes que pueden ser de utilidad. En (b), aparece el resultado del residuo de la apertura horizontal con un elemento estructurante de tamaño 20. Si recordamos que una apertura [6] se queda con la zonas claras de una imagen, conseguiremos eliminar la líneas rectas originadas por los cortes. Para la elección del tamaño del elemento estructurante se deberá tener en cuenta el número de pixels que la cortinilla descubre en cada fotograma (desplazamiento de la cortinilla) ya que sino podrán eliminarse los trozos de la línea oblicua que indica la cortinilla.

Pero además será necesario eliminar el efecto producido por el movimiento. Una característica que tienen la líneas correspondientes a las cortinillas es que van de la columna primera hasta la última. Se pueden realizar dos reconstrucciones utilizando como marcador una imagen negra del mismo tamaño que la imagen "tira", que incluirá una zona blanca de varios pixels de ancha y todo lo alto de la imagen. En uno de los marcadores la región blanca estará a la izquierda de la imagen, y en el otro a la derecha.

La primera reconstrucción se hace tomando como referencia la imagen "tira" de partida y como marcador la imagen con la región blanca a la derecha. El resultado se utiliza como referencia en la otra reconstrucción y como marcador la imagen con la región negra de la derecha. De esta forma, finalmente solo nos quedamos con las líneas que van de un lado a otro de la imagen. En la figura 4.19, se muestra el resultado obtenido para la subsecuencia de *newsa_8.val*.

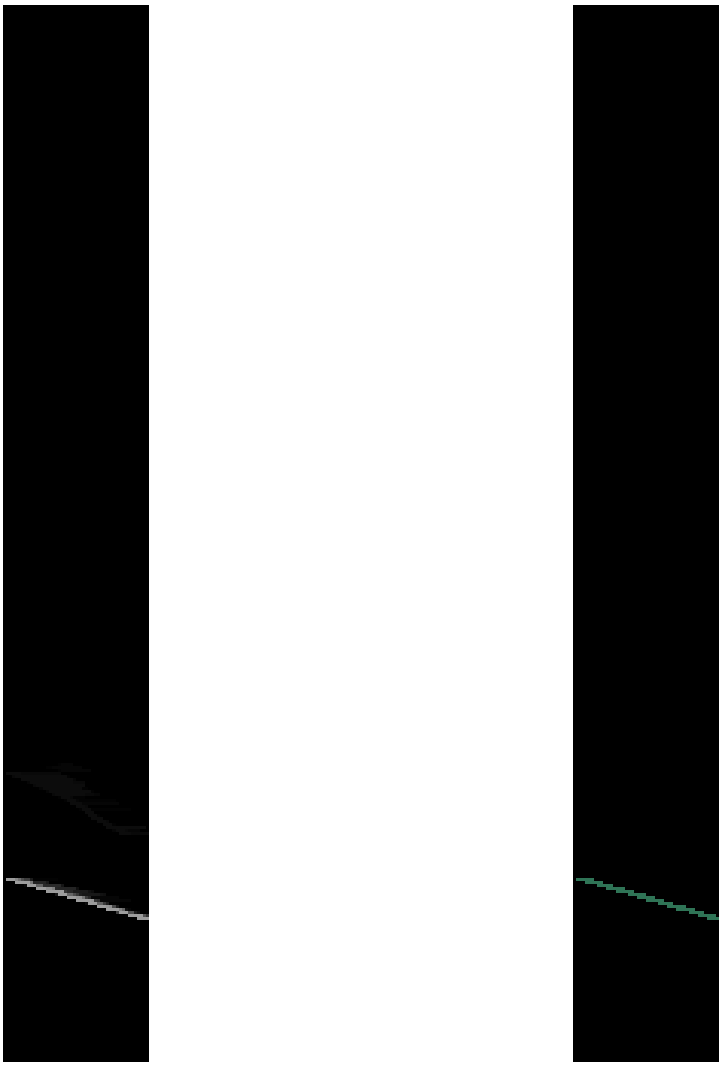
El siguiente paso será segmentar la imagen, quedándonos con cada una de las líneas oblicuas, que se corresponderán a cada una de las cortinillas que tiene la secuencia. Una posible alternativa de segementación será aplicar un umbral. En (b) se muestra el resultado de la umbralización.

[6] Consultar apéndice de Morfología Matemática.



(a) (b)

Figura 4.18: Resultado de busqueda de cortinillas verticales para una subsecuencia de *news_8.val*: (a)Imagen "tira" para proyecciones por columnas. (b)Resultado de realizar una apertura horizontal de tamaño 20.



(a) (b)

Figura 4.19: Localización de cortinillas verticales en la subsecuencia de *newsa_8.val*: (a)Imagen "tira" depués de procesado. (b)Segmentación mediante umbral de la imagen de (a).

4.3.3 Ejemplo de falso positivo en la localización de cortinillas

Con el ejemplo de la sección anterior, ha quedado demostrada la utilidad de proyecciones por filas o columnas para localizar las cortinillas. Sin embargo, como en la mayoría de las técnicas utilizadas para segmentar secuencias de vídeo, se pueden producir errores de detección si el contenido de las imágenes cumple algunas características muy específicas.

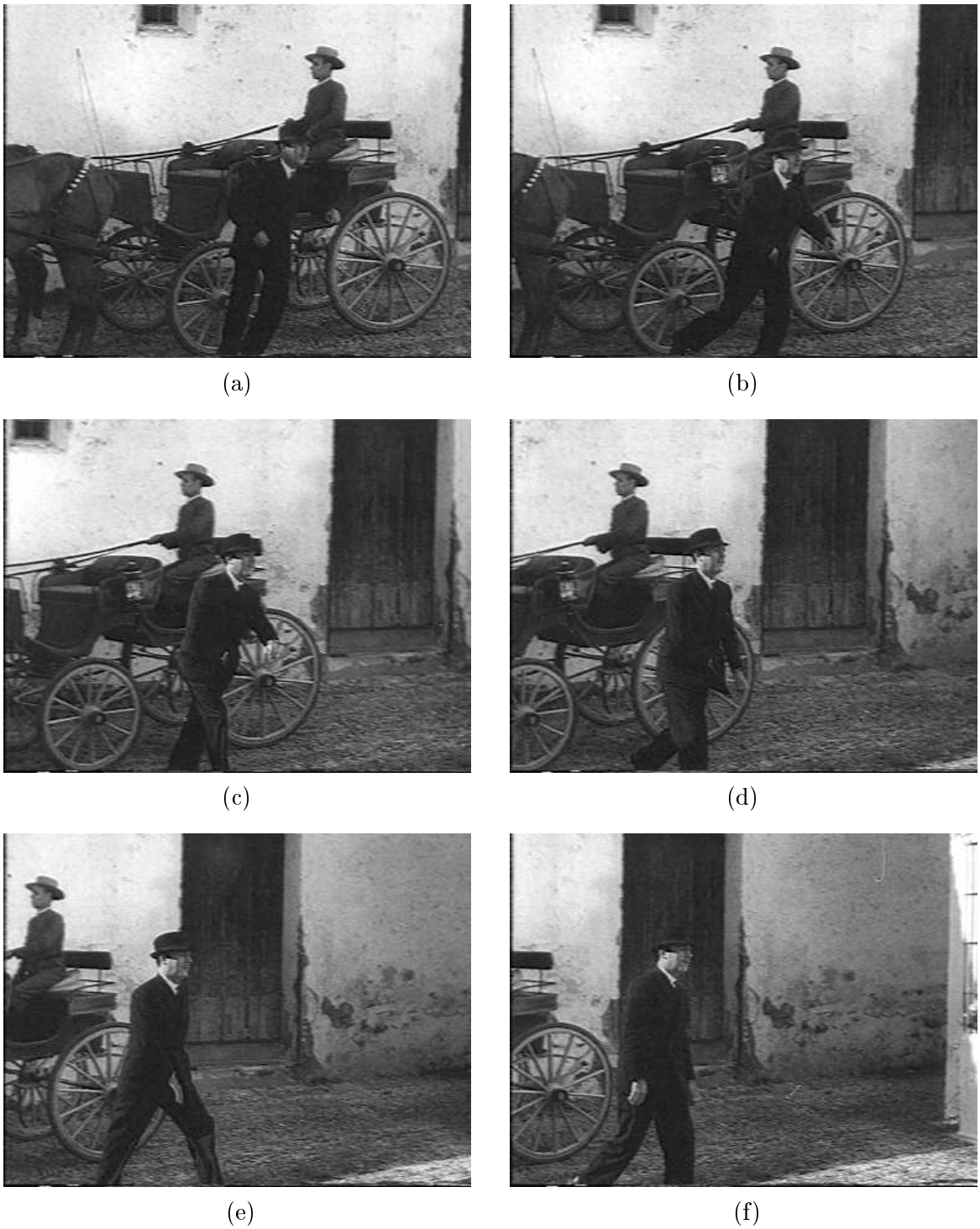
Centrándonos en las cortinillas, si en las imágenes de los fotogramas sucesivos de una secuencia, aparece un objeto limitado por líneas horizontales o verticales y la cámara realiza un movimiento panorámico en la dirección perpendicular a las líneas del objeto, se confudirá con una cortinilla. Veamoslo con un ejemplo.

En la figura 4.20 se muestran seis fotogramas consecutivos de una secuencia.

Si observamos, la línea que produce en la imagen la unión de la puerta (muy oscura) con la pared (muy clara) se desplaza en sentido vertical a lo largo de la imagen. El efecto que produce es el mismo que si se tratase de una cortinilla vertical.

Por eso, al realizar todo el proceso de localización de cortinillas verticales, se obtiene un falso positivo en los fotogramas correspondientes a la región donde se mueve la puerta, figura 4.21.

Aunque no se producirán con mucha frecuencia, no hemos encontrado ninguna forma de poder diferenciar estos Falsos Positivos de las auténticas cortinillas.

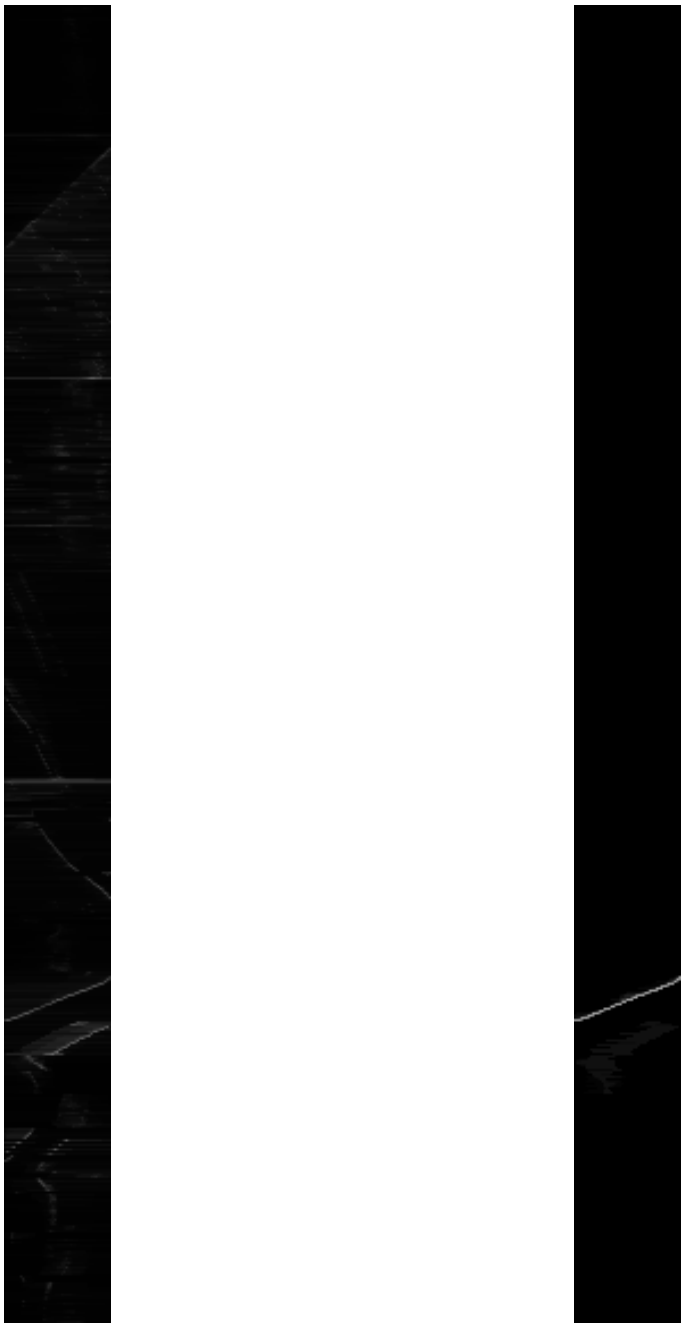


(a) (b)

(c) (d)

(e) (f)

Figura 4.20: Ejemplo de secuencia que produce un falso positivo al detectar cortinillas verticales.



(a) (b)

Figura 4.21: Localización de cortinillas verticales: (a)Imagen “tira” inicial. (b)Imagen “tira ” depués de todo el procesado.

4.3.4 Localización de las cortinillas con el PDIWin32

En primer lugar y como información básica para la localización de las cortinillas, se deberán calcular las proyecciones normalizadas por columnas (si se buscan cortinillas verticales) o por filas (si se buscan cortinillas verticales), de la imagen diferencia entre fotogramas consecutivos de la secuencia. Las entradas de menú que se encargan de ello, pueden verse en la figura 4.22.

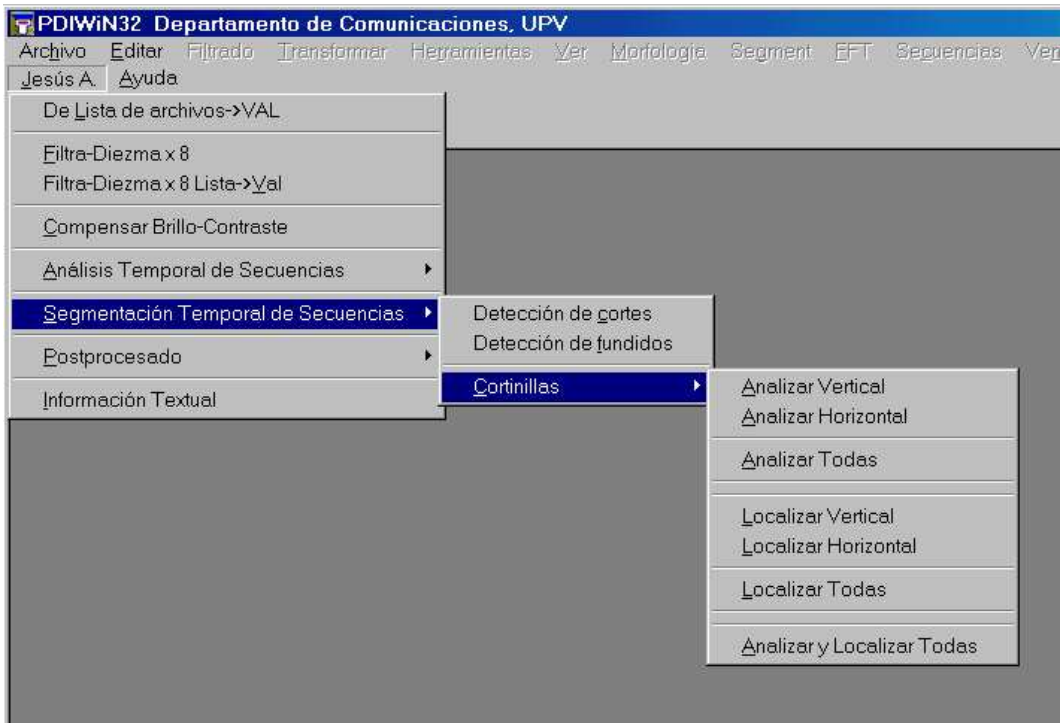


Figura 4.22: Entradas de menú para el procesado de las cortinillas.

Cuando la ventana activa sea una secuencia *VAL* [7] y seleccionemos la función **Analizar Vertical**, nos calculará las proyecciones por columnas. De igual forma que para las otras funciones del **Análisis Temporal de Secuencias**, el proceso puede cancelarse antes de finalizar. Al terminar, si todo ha ido correctamente, aparece una ventana que nos mostrará el nombre del fichero de texto donde se han guardado los valores de las proyecciones: *difercol.txt*. De igual forma, si se selecciona **Analizar Horizontal**, se realizará el proceso análogo por filas, para detectar cortinillas horizontales, y el resultado lo guardará en *diferfil.txt* [8]. Si se elige **Analizar Todas**, se realizarán ambos procesos de forma consecutiva.

[7] En caso contrario, y como ya comentamos, nos mostrará el error correspondiente.

[8] Para un conocimiento profundo del formato de estos dos ficheros, remitimos al apéndice D.

Una vez se dispone de estos dos ficheros, se puede proceder a la localización de las cortinillas, seleccionando algunas de las siguientes funciones: **Localizar Vertical**, **Localizar Horizontal** o **Localizar Todas**.

En cualquiera de ellas, lo primero que hace es comprobar que se encuentran los ficheros necesarios en el directorio de trabajo. Si no es así, nos mostrará un mensaje de error.

El proceso de localización es el descrito en los apartados anteriores y el resultado será, en primer lugar, dos imágenes "tira". La primera corresponderá al resultado de la reconstrucción geodésica de las proyecciones; y la otra imagen será el resultado definitivo, después de todo el procesado de segmentación y "limpieza", para quedarnos con las líneas oblicuas de las cortinillas. Además, se crearán dos ficheros de texto.

Estos ficheros son: *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.

El mismo proceso realizado en el PDIWIN32, también lo hemos programado para poder ser realizado en el programa MATLAB, partiendo de la información de los ficheros *difercol.txt* y *diferfil.txt* (será necesario eliminar las dos primeras líneas de cada fichero antes de abrirlos con MATLAB).

4.3.5 Funciones programadas en el PDIWin32 para la localización de las cortinillas

Ha sido necesario programar dos funciones para el cálculo de las proyecciones por filas y por columnas de la diferencia entre dos fotogramas consecutivos de una secuencia. Como hemos comentado anteriormente, se podrán utilizar con imágenes de grises e imágenes RGB.

P8SumapixelsColum

- **Sinopsis:** Para el cálculo de la proyección por columnas de una imagen (no se realiza ninguna normalización). Funcionará para imágenes de 8 o 24 bits, y en el caso de imágenes de color devolverá un único número resultado de la combinación del obtenido para R, G y B.
- **Formato:** `P8SumapixelsColum(imgdes *desimg,float *sum_pixels_col);`
  - `imgdes *desimg`: Puntero de la imagen a proyectar.

- `float *sum_pixels_col`: Puntero del vector de números en coma flotante donde almacenar los valores de la proyección para cada columna. Tendrá tantos elementos como pixels de ancho tenga la imagen.

P8SumapixelsFila

□ **Sinopsis:** Para el cálculo de la proyección por filas de una imagen (no se realiza ninguna normalización). Funcionará para imágenes de 8 o 24 bits, y en el caso de imágenes de color devolverá un único número resultado de la combinación del obtenido para R, G y B.

□ **Formato:** `P8SumapixelsFila(imgdes *desimg,float *sum_pixels_fil);`

- `imgdes *desimg`: Puntero de la imagen a proyectar.
- `float *sum_pixels_fil`: Puntero del vector de números en coma flotante donde almacenar los valores de la proyección para cada fila. Tendrá tantos elementos como pixels de alto tenga la imagen.

4.4 RESULTADOS OBTENIDOS EN LAS SECUENCIAS DE PRUEBA

A continuación, mostramos los resultados de la detección de fundidos y cortinillas obtenidos para las secuencias de prueba. En el caso de los cortes, incluíamos todas las secuencias, ya que todas tenían alguno. En este caso solo se incluyen en las tablas aquellas secuencias que tienen el efecto buscado.

La nomenclatura utilizada es la misma que para los cortes: se incluirá una columna con los fundidos (o cortinillas) que tiene la secuencia, otra columna para los detectados por el método, la siguente indicará cuantos de los detectados no son fundidos (o cortinillas) auténticos, y los denominaremos Falsos Positivos (FP), y en la última columna se incluye el número de Falsos Negativos (FN), se corresponden con los fundidos (o cortinillas) reales no detectados.

Resultados de la detección de fundidos

Se han utilizado los siguientes valores para los parámetros configurables:

- Umbral para la correlación fotogramas $n-1$, n y $n+1$, $u_{\rho_{fundidos}} = 0.3$.
- Umbral para el error en la estimación de la varianza, $u_{\sigma^2_{fundidos}} = 350$.

Secuencia	**Frames**	**Fundidos**	**Detectados**	N_{FP}	N_{FN}
"NEWSB_8.VAL"	1497	1	1	0	0
"NEWSA_8.VAL"	1907	1	1	0	0
"CYCLING_8.VAL"	1998	3	2	0	1
"ALCOI_8.VAL"	783	4	15	11	0
"MALVA_8.VAL"	1782	1	1	1	1
"CHAPL_8.VAL"	2713	2	100	100	2
"CHAPL_8C.VAL"	2713	2	20	20	2

Tabla 4.1: Resultados de la detección de fundidos.

Resultados de la detección de cortinillas

Secuencia	**Frames**	**Cortinillas**	**Detectadas**	N_{FP}	N_{FN}
"NEWSA_8.VAL"	1907	2	1	0	1
"MALVA_8.VAL"	1782	1	1	0	0

Tabla 4.2: Resultados de la detección de cortinillas.

Estos resultados, junto con los que se obtuvieron para los cortes, se comentarán más en profundidad en el capítulo de conclusiones.