

**UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA**

**ESCUELA UNIVERSITARIA DE INGENIEROS TÉCNICOS DE  
TELECOMUNICACIÓN**



**TRABAJO FIN DE CARRERA**

*CD-ROM con demostraciones interactivas sobre el  
tratamiento digital de imágenes*

*Fco. Javier Selvi Nácher*

**UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA**

**ESCUELA UNIVERSITARIA DE INGENIEROS TECNICOS DE  
TELECOMUNICACION**



**TRABAJO FIN DE CARRERA**

*CD-ROM con demostraciones interactivas sobre el  
tratamiento digital de imágenes*

**ESPECIALIDAD**

***Imagen y Sonido***

**AUTOR**

***Fco. Javier Selvi Nacher***

**TUTORA**

***Fátima Casado Miraz***

**PRESIDENTE**

**SECRETARIO**

**VOCAL**

*Este trabajo está dedicado con todo cariño*

*a mis padres y a mi wely.*

## *Introducción al Trabajo Fin de Carrera*

El presente Trabajo se ha desarrollado principalmente con la finalidad de servir de apoyo en las prácticas de Tratamiento Digital de Imágenes, tema que actualmente se imparte en la asignatura de Televisión II. Para ello se han empleado todas las herramientas disponibles, explotando principalmente las posibilidades multimedia del equipo informático de que dispone el laboratorio.

La imagen digital, integrada en un sistema multimedia, será, junto al sonido, el formato principal en las telecomunicaciones de un futuro cercano. En el primer capítulo se comentan sus fundamentos y las aplicaciones en las que ya se ha hecho necesario el tratamiento digital de la imagen. En el terreno de las videoproducciones, hilo conductor de la asignatura de TVII, resulta fundamental el conocimiento de estas técnicas.

Se han preparado una serie de demostraciones con una herramienta interactiva, el Action, a modo de documentales, sobre las técnicas relacionadas con imágenes digitales. La duración de toda la serie ronda los 40 minutos. Estas demostraciones se han editado digitalmente, buscando ante todo una presentación adecuada en un entorno multimedia, donde se funden imagen y sonido para ofrecer un producto didáctico y ameno. La interactividad constante con el usuario dota a estas demostraciones de mayor dinamismo e incrementa las posibilidades de que el alumno acabe asimilando completamente los conceptos básicos.

Para que el alumno, además de ver las posibilidades en el campo de la imagen digital, pueda experimentar con ellas, se ha programado un software de entrenamiento que permite realizar todos los procesados que se van viendo en los documentales. El programa, bautizado como TIRMA, se ha creado con una potente herramienta matemática, que es el Matlab. A diferencia del resto de programas de tratamiento de imágenes, TIRMA muestra al usuario los fundamentos de cada una de las operaciones que lleva a cabo, permitiendo un análisis numérico en todo momento de las matrices que constituyen la imagen y sus colores. Además está interrelacionado con los documentales y por ello, este programa será la base de la práctica, ya que puede mostrar las demostraciones de cada tema, permite al alumno practicar sobre otras imágenes diferentes de las del documental, indagando en el proceso que se ha realizado, y por último ofrece una completa ayuda en la que se dispone de explicaciones más detalladas para cualquier consulta de estos procesos. Se ha dotado a este software de un entorno amigable, para que el interface con el usuario resulte sencillo de manejar y de agradable funcionalidad.

También se adjuntan otros programas de distribución shareware, que completan la visión de las aplicaciones de la imagen digital. Son programas de visualización, conversión de formatos y captura de imágenes, creación de morfismos y transiciones, generación de estéreogramas y demostración de entornos virtuales mediante la generación de imágenes con objetos en 3 dimensiones.

Una recopilación de imágenes digitales de diferentes estilos, formatos y resoluciones conforma una librería con la que se puede experimentar en todos estos programas.

Como una aplicación concreta y resumen del tratamiento digital de sonido e imagen, se presenta un programa de generación y análisis de espectrogramas. Es una herramienta desarrollada bajo Matlab que combina el análisis digital de sonido y una vez obtenido el espectrograma, el análisis de esta imagen. Una de las aplicaciones con vistas a la cual se desarrolla este software, es el análisis de los silbos herreños, aunque su utilidad es general en el estudio de las representaciones tiempo-frecuencia de señales de audio.

Todo el trabajo, así como la presente memoria, se compactan en un CD-ROM y quedan disponibles en un sólo disco. Como se comenta en el capítulo 9, el CD-ROM es el soporte ideal para este trabajo. Las razones básicas por que se elige el CD-ROM son la disponibilidad en el laboratorio de una unidad reproductora y la capacidad necesaria para albergar todo el volumen de información. La reciente aparición de este formato hace que la oferta de títulos sea muy genérica, y la grabación de un trabajo específico con fines didácticos, constituye una innovación importante, así como la entrega de trabajos fin de carrera en este soporte.

## Índice

1. Introducción al Tratamiento Digital de Imágenes.	1
2. TIRMA: software para el tratamiento digital de imágenes.	6
3. Listado del programa TIRMA y accesorios.	42
4. Action como herramienta de presentación interactiva.	64
5. Explicación de las Demostraciones.	72
6. Software adicional incluido.	77
7. Librería de Imágenes.	88
8. Aplicación a una imagen digital concreta: Espectrograma.	93
9. El CD-ROM como medio ideal de almacenamiento para el Trabajo.	107
10. Características del equipo y posibles problemas	116
11. Bibliografía.	119
12. Agradecimientos.	121

## 1.Introducción al tratamiento digital de imágenes:

El procesado digital es una técnica de reciente aparición que se está imponiendo en muchos campos, especialmente en aquellos donde se trabaja con señales que por cualquier motivo deben ser modificadas. En la rama de telecomunicaciones hay un campo donde el procesado digital está causando una autentica revolución y es en la imagen y el sonido. Esta técnica se ha consolidado antes para las señales de audio ya que los requerimientos hardware son menores por la sencillez de la señal frente a la de vídeo.

Actualmente le está llegando el turno a la imagen ya que los equipos digitales empiezan a conseguir calidades similares a la de los equipos analógicos. En la ingeniería del sonido se han superado con creces todos aquellos sistemas de almacenamiento analógicos tras la llegada del Compact Disc, ya no sólo en la calidad del sonido grabado sino en las posibilidades de procesado que aparecen al disponer del sonido digitalizado. Del mismo modo, los sistemas de imagen tanto de televisión como de fotografía están siendo igualados en calidad por los sistemas digitales. Esto no quiere decir que los sistemas digitales vayan a retirar a los que se emplean actualmente, pero van a suponer un serio competidor a muchos de ellos.

A pesar de que ya se puede rozar prácticamente la calidad de una fotografía, las cámaras CCD no van a relegar el uso de las cámaras fotográficas aunque sí aportarán nuevas técnicas de procesado como se mostrará en este trabajo.

Estamos tratando pues de una nueva tecnología que va a complementar a los sistemas actuales generando nuevas posibilidades de procesado y aportando mejores herramientas a los usuarios.



Los campos de aplicación del tratamiento digital de imágenes se extienden con rapidez. En el campo de la televisión se está desarrollando fuertemente esta nueva filosofía con la llegada de la televisión digital. El problema que presenta el tratamiento de imágenes de televisión es que hay una gran cantidad de información que transcurre en tiempo real y el hardware debe ser muy potente y rápido ante todo.

En la fotografía ya está implantado a nivel comercial y se pueden encontrar cámaras digitales que graban las imágenes directamente en disco; posteriormente estas imágenes se pueden visualizar en un monitor e incluso ser modificadas por el propio usuario. Una de las más importantes aplicaciones es en el campo de la medicina, donde se están proporcionando grandes ventajas a los médicos a la hora de conocer en profundidad el estado de los pacientes, al extraer mayor cantidad de información de imágenes como radiografías, ecografías, resonancias magnéticas,...; hay otros muchos campos donde estos avances están resultando primordiales como la astronomía con la incorporación de cámaras CCD a los telescopios, en visión artificial, reconocimiento de objetos, etc...

Una vez comentada la importancia del Tratamiento Digital de Imágenes veamos en qué consiste. Podemos empezar diciendo que se trata de la manipulación por medios digitales de imágenes previamente digitalizadas. Vamos a suponer que estamos tratando una imagen fija, estática en el tiempo; por tanto la única variación que tendrá será en el espacio, en los ejes X e Y. Estamos pues ante una señal bidimensional, ya que hay dependencia de 2 variables. Esto diferencia su tratamiento digital en algunos aspectos del que se lleva a cabo con el sonido, ya que en éste es una señal unidimensional por depender tan solo del tiempo.

Sabemos que cualquier color se puede obtener como una combinación lineal de los tres colores básicos Rojo, Verde y Azul, y de este modo cada imagen en color se puede considerar como la superposición de tres imágenes con la luminancia de los colores. Así el tratamiento de una imagen en color se puede descomponer en un triple proceso de tratamiento sobre imágenes que sólo contienen información de luminancia, como son las imágenes en escala de grises. Por lo tanto, en lo que sigue, nos referiremos a imágenes en escala de grises y que no varían con el tiempo.

El primer paso para poder procesar la imagen es digitalizarla y dicho procedimiento implica una discretización de 2 tipos:

Espacial: hay que discretizar el número de elementos que van a constituir la imagen tomando una rejilla cuyas celdas representarán cada valor binario de la señal digital. Esta retícula de forma normalmente rectangular, definirá la división de la imagen en elementos básicos de forma también rectangular llamados Pixels. A cada pixel se le asocia un valor proporcional a la intensidad luminosa de su celdilla (puede ser un promedio de la luminancia de toda la celdilla o el valor de su centro).

Hay un concepto adicional sobre los pixels que es su relación de aspecto y que se calcula como el cociente entre su anchura y altura. Para que la imagen global tenga el mismo aspecto que cuando se digitalizó, los pixels de pantalla deben tener la misma relación de aspecto que con la que se digitalizó.

De amplitud: cuando ya se ha obtenido un conjunto discreto de puntos hay que discretizar las intensidades luminosas. Se ha comprobado que el ojo humano no es capaz de distinguir más de 64 niveles de grises diferentes, por lo que podemos

obtener una buena calidad con 6 bits para cuantificar cada pixel. De todos modos, los ordenadores trabajan de un modo más eficiente cuando lo hacen con números potencias de 2, por lo que se acostumbra a usar 8 bits/pixel obteniendo 256 niveles de grises. Se suele asignar por convenio el 0 al color negro y el 255 al blanco.

Una vez la imagen ha sido digitalizada se le pueden aplicar las técnicas de procesado digital que se van a estudiar en este trabajo.

## 2.TIRMA:

### software para el tratamiento digital de imágenes.

El programa TIRMA de Tratamiento de Imágenes de Resolución Medianamente Aceptable, se ha desarrollado usando plenamente las facultades del Matlab para Windows. Se trata de un software de entrenamiento en el procesado de imágenes digitales, programado al estilo de las herramientas de Matlab, para mantenerse dentro de este entorno, especialmente en el Toolbox de imágenes.

No alcanza a procesar las imágenes de gran resolución que sí pueden procesarse con otros programas disponibles en el equipo, como Corel PhotoPaint o PhotoStyler. Por contra, ofrece al usuario un control permanente de lo que está haciendo, y la posibilidad de consultar y modificar las imágenes a nivel numérico, cosa que el resto de programas que procesan imágenes, no permite. Además incorpora algunos tratamientos que no se encuentran en el resto de programas disponibles.

La finalidad de este programa es que el alumno pueda practicar lo que ve en las demostraciones, sabiendo exactamente qué le ocurre a las imágenes y que no le resulte transparente. La limitación que impone el equipo actual se encuentra algo por encima de imágenes de 320x200 pixels de 256 colores, pero esta resolución resulta más que suficiente para apreciar el efecto de todas las opciones que el programa puede aplicar a una imagen.



El programa se instala con su conjunto de menús en la parte superior de la pantalla, dejando visible la ventana de comandos de Matlab, que se empleará para señalar distintos mensajes y permitirá la entrada de datos precisos en algunas operaciones. En la parte inferior de la pantalla se instalarán los iconos de las aplicaciones minimizadas de Windows.

Incluye un amplio abanico de menús, con 80 opciones que serán detalladas a continuación.:

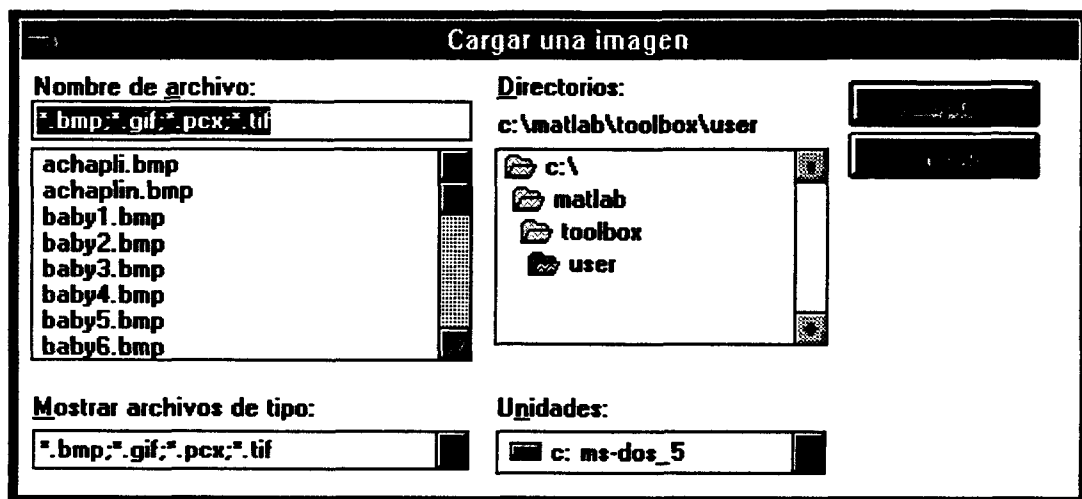
### *Ayuda del menú Ficheros*

Las opciones disponibles son: Abrir, Guardar, Borrar y Salir. Cada una de ellas puede ser seleccionada con el ratón o bien con la combinación de teclas Alt+inicial de cada opción.

#### Abrir:

Esta opción es la que habilita el resto de los menús desactivados en un principio. Permite cargar una imagen grabada de los formatos bmp, gif, pcx y tif.

Al cargar una imagen se abre una ventana con la imagen y su nombre. Se permite tener hasta 8 ventanas abiertas con diferentes imágenes. Para seleccionar la imagen aparece una ventana de diálogo como la que se muestra:



### Guardar:

Cuando se ha procesado una imagen y se desea almacenar el resultado se emplea esta opción. Se graba la imagen de la última figura activada, pasando como nombre inicial el de la figura, aunque puede ser modificado. Se puede acceder a cualquier unidad y a cualquier subdirectorio.

Si el nombre con el que se pretende grabar la imagen ya pertenece a un fichero existente en el subdirectorio actual, aparece un mensaje indicándolo y preguntando si se desea emplear ese nombre, perdiendo el fichero anterior.

## Borrar:

Permite borrar cualquier fichero seleccionándolo con una ventana de diálogo similar a las anteriores opciones.

## Salir:

Esta opción abandona el programa TIRMA cerrando previamente las ventanas abiertas durante su ejecución y borrando todas las variables empleadas por el programa.

## *Ayuda de Operaciones*

Este es el menú mayor del programa y permite realizar hasta 39 operaciones diferentes sobre la imagen de la última figura activada. Se puede acceder a cada submenú mediante teclado pulsando la tecla Alt más la inicial del submenú.

Los submenús se han distribuido de la siguiente forma:

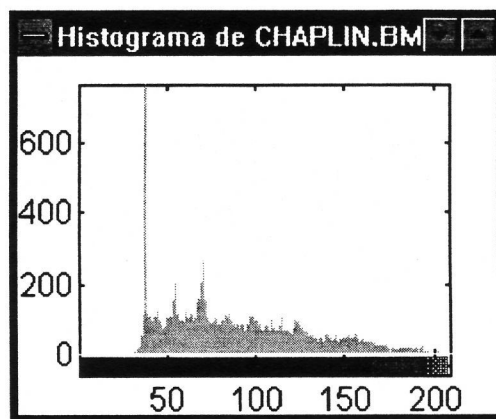
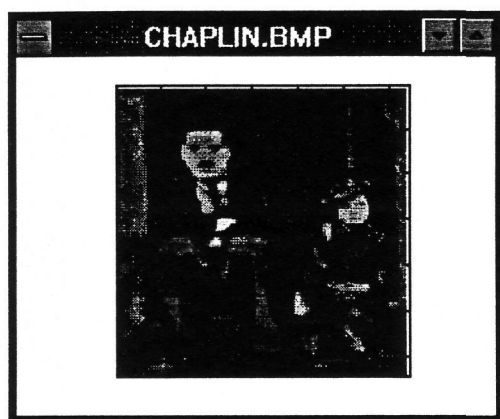
Operaciones Puntuales, Locales, Globales y Geométricas.

## OPERACIONES PUNTUALES:

El valor de cada pixel será modificado sólo en función de su valor inicial. No influirá el resto de la imagen. Las operaciones puntuales que se han dispuesto son las siguientes:

### Histograma:

Crea una nueva ventana con el histograma de la imagen que se está usando, dejando intacta la imagen original. El nombre de la nueva ventana es 'Histograma de ...', el nombre de la imagen.

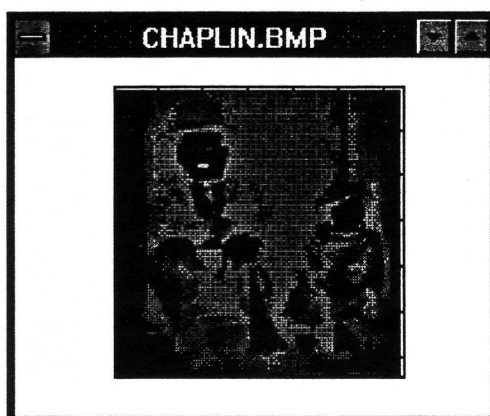


El histograma consiste en una gráfica que muestra la distribución de colores en la imagen, es decir, indica cuántos pixels son de cada uno del total de colores que aparezcan en la imagen. En el eje de abscisas se encuentran los colores de la imagen y en el de ordenadas el número de veces que aparece cada color en la imagen. En una imagen en escala de grises da idea de brillo y contraste.



## Negativo:

Calcula el complementario de cada pixel, esto es, el valor máximo menos cada uno de los valores de cada pixel. Si se trata de una imagen en escala de grises con 256 niveles, se restará a 256 la intensidad de cada pixel, resultando así que los pixels claros (valores cercanos a 256) pasarán a ser oscuros (valores cercanos a 1) y viceversa. El resultado es igual a un negativo fotográfico. Si la imagen es en color y la intensidad de su paleta de colores no ha sido previamente ordenada en luminancia, el resultado será que los colores de la imagen cambiarán de unos pixels a otros.



## Posterizar:

En la ventana de comandos se solicita el número de niveles para posterizar, es decir, a cuantos colores será reducida la imagen. Para obtener buenos resultados se aconseja ordenar previamente la paleta de colores por luminancia, sobretodo si la imagen es en color. Si se desconoce el número de colores existentes en la imagen se puede consultar en Estadística\Resolución.

El número de niveles solicitado para posterizar no debe ser superior al número de colores de la imagen.

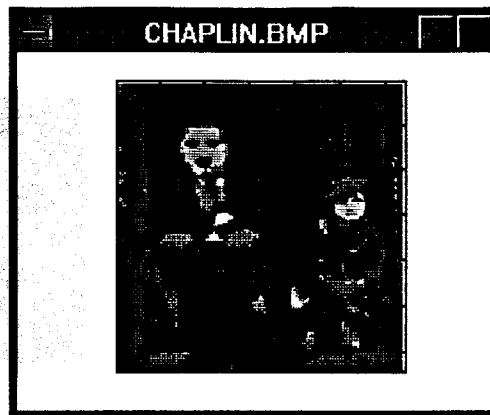
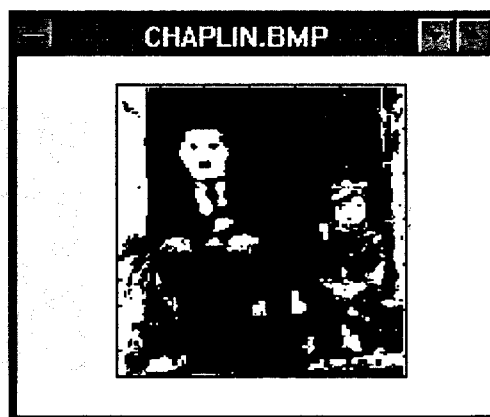


Imagen posterizada con 4 niveles

### Binarizar:

Esta operación transforma la imagen a blanco y negro. Es similar a una posterización de 2 niveles, pero aquí se puede establecer el umbral entre los pixels que serán considerados negros y los que pasarán a ser blancos. Se aconseja emplear un nivel de binarización intermedio considerando el número de colores de la imagen. En el ejemplo se ha binarizado con un nivel de 120.



Nota: otra forma de calcular rápidamente el número de colores de la imagen es usar la expresión 'length(map)' en la ventana de comandos.

## Umbralizar:

Emplea el rango dinámico disponible por la paleta de colores en un rango de la imagen menor del inicial y delimitado por el nivel de umbralización. Los pixels de intensidad menor que la indicada pasarán a tener el primer color del mapa de colores que suele ser negro, y el resto abarcarán todo el rango de colores. Para una mejor comprensión de esta operación se propone acudir a la demostración del Histograma, donde hay un ejemplo de umbralización.

## Clipping:

Es similar a la umbralización pero llevada a cabo por la zona alta y baja de intensidades de la imagen. Para hacer un clipping hay que establecer un nivel inferior y otro superior, menores que el número de colores de la imagen, niveles que son solicitados en la ventana de comandos. Los pixels de valor menor al nivel inferior pasarán a ser del primer color de la paleta, y los pixels de valor superior al nivel superior pasarán a ser del último color de la paleta (blanco, generalmente). Los pixels comprendidos entre ambos niveles extenderán su rango dinámico a toda la paleta excepto los dos colores ya empleados. Se logra un gran aumento de contraste en la zona intermedia de tonos grises, sacrificando los colores muy oscuros y los muy claros (si la luminancia está ordenada).

## Slicing:

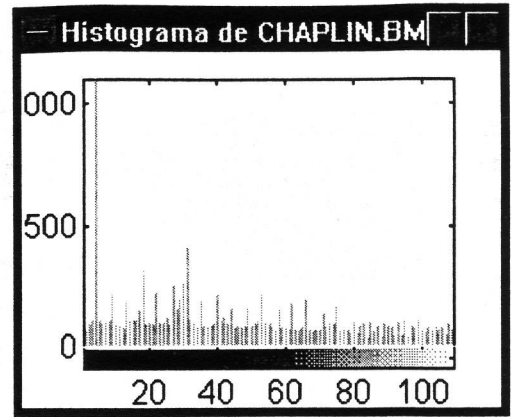
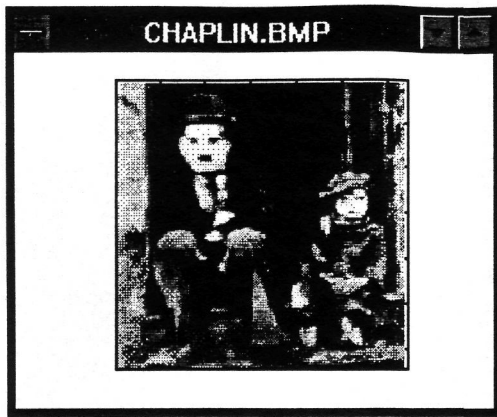
Esta operación se emplea para aislar una parte de la imagen con sus colores originales y pasar el resto a un mismo valor, que suele ser el primero de la paleta de colores. Puede ser empleado para realizar una segmentación.

Tras esta operación pueden quedar colores sin emplear por lo que se recomienda reducir la paleta de colores con la opción Colores\Reducir el nº colores.

## Ecualizar histograma:

Distribuye los colores de la imagen para que ésta tenga un reparto de colores uniforme. Se busca que el histograma de la luminancia de la imagen se aplane lo más posible. Ello permite distinguir colores que antes podían ser muy próximos. Generalmente el resultado suele ser beneficioso para la inteligibilidad de la imagen, aunque hay casos en que se pierde contraste en zonas importantes (se recomienda acudir a la demostración Histograma).

Al ecualizar el histograma, cambia la distribución de colores, pudiendo aparecer colores repetidos, por lo que se aconseja reducir el nº de colores.



### Añadir ruido:

Esta opción puede ser útil para comprobar el efecto de determinados filtros sobre imágenes contaminadas con ruido, o también como aplicación artística envejeciendo la imagen. La imagen pasa automáticamente a escala de grises, aunque fuese en color.

Hay 3 tipos de ruido que podemos emplear:

**GAUSIANO:** ruido blanco del que podemos establecer su media y su varianza en la ventana de comandos. Los valores entre paréntesis son orientativos.

**SAL Y PIMIENTA:** se trata de ruido impulsivo aleatorio que toma valores extremos como son el blanco y el negro en una imagen de grises. Se puede establecer su densidad con un valor orientativo de 0.05

**SPECKLE:** añade ruido multiplicativo de varianza sintonizable en la ventana de comandos. El valor orientativo para empezar a probar es 0.04

## OPERACIONES LOCALES

Estas operaciones se extienden a toda la imagen pero realizando un procesado por áreas. El resultado de cada pixel estará determinado por su vecindad, es decir, por el valor de sus pixels contiguos. En todas las operaciones se ha empleado una vecindad de 3x3 puntos ya que las imágenes con las que se trabaja son de baja resolución y así el resultado no se ve influido por gran parte de la imagen sino una pequeña zona. Si se desea trabajar con áreas mayores se puede hacer con una máscara de usuario. La clasificación de operaciones locales efectuada en el menú es la siguiente:

### Suavizar:

Haciendo uso de una matriz de coeficientes que representan los valores de un filtro espacial, se efectúa una convolución bidimensional con la imagen, dando como resultado la imagen filtrada. Para suavizar la imagen, los filtros son Paso-Bajo ya que se trata de eliminar los cambios bruscos de intensidad que representan altas frecuencias espaciales. Se proponen 3 efectos distintos de suavizado (soften):

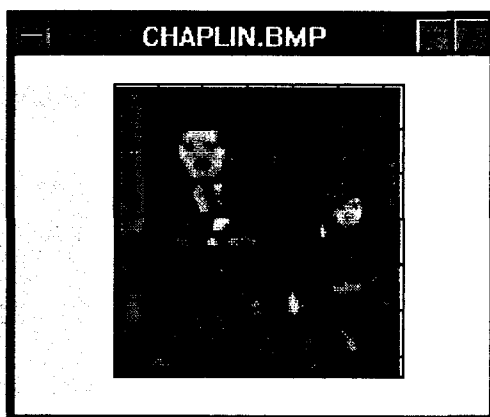
IMAGEN:	[1/9 1/9 1/9	Esta máscara calcula la media de la
	1/9 1/9 1/9	vecindad. Reduce ruido en la imagen.
	1/9 1/9 1/9]	

CONTORNOS: [1/8 1/8 1/8  
1/8 0 1/8  
1/8 1/8 1/8]

Este filtro sólo considera a los pixels contiguos y no emplea el valor original del pixel que se está calculando.

DESENFOCAR: [1/14 2/14 1/14  
2/14 2/14 2/14  
1/14 2/14 1/14]

Ponderando más a los más cercanos se logra un efecto de desenfoque, aunque hay que decir que la vista prefiere una imagen ruidosa a una desenfocada, ya que tratamos inútilmente enfocarla.



### Realzar:

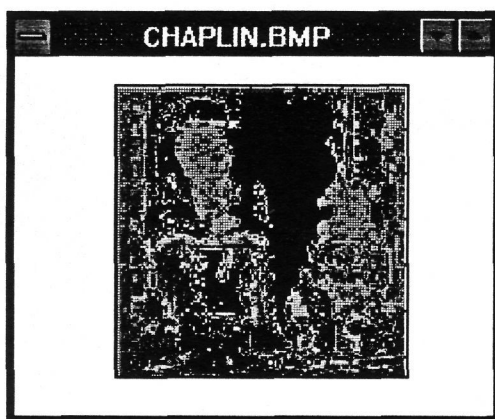
Es la operación contraria a la anterior ya que se trata de filtros Paso-Alto. Las operaciones de realce dejan pasar los saltos bruscos de intensidad y atenúan las transiciones suaves. Este efecto puede ser empleado en la detección de detalles inapreciables en principio en la imagen. Se dispone de 2 máscaras con efecto sobre la imagen o sobre los contornos:

IMAGEN:             $\begin{bmatrix} -1/4 & -1/4 & -1/4 \\ -1/4 & 13/4 & -1/4 \\ -1/4 & -1/4 & -1/4 \end{bmatrix}$             Esta máscara afila (Sharp) los detalles finos de la imagen en general.

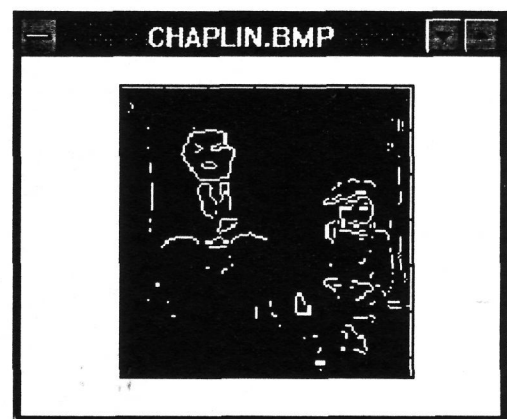
CONTORNO:         $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$             Filtro empleado para realzar contornos como se muestra en la demostración de Filtrados.

Detectar contornos:

Hay que distinguir entre realzar contornos y detectarlos. Aunque enfatizamos los detalles realzando los contornos, la imagen mantiene un parecido con los colores originales, pero si realizamos una detección de contornos, pasamos a una imagen de prácticamente 2 colores (blanco y negro), uno para los contornos y el otro el fondo. Estas técnicas se emplean principalmente en visión artificial y robótica (detección de blancos).



Realzar contornos



Detectar contornos



Hay diversos métodos para realizar esta operación y además se pueden aplicar sobre direcciones particulares; por ello se ha clasificado por la dirección:

**TODAS DIRECCIONES:** hay que seleccionar en la ventana de comandos el método que se va a emplear. En la ventana se comentan las diferencias entre los 4 métodos propuestos.

**HORIZONTAL o VERTICAL:** también hay que elegir el método como en el caso anterior. La imagen resultante es una matriz  $x$  cuyos únicos valores son 0 y 1. Si aparece algún problema con este rango de valores se puede modificar mediante la ventana de comandos haciendo ' $x=x+1;imshow(x,map)'$ .

**DIRECCIONES DIAGONALES:** realizan un filtrado por máscara de coeficientes de tamaño  $3 \times 3$  dependiendo de la dirección:

<u>Abajo-Izquierda</u>	<u>Abajo-Derecha</u>	<u>Arriba-Izquierda</u>	<u>Arriba-Derecha</u>
$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$

El problema de estos filtros es que la suma de sus coeficientes es nula, por lo que  $H(0,0)=0$  y la componente continua de la imagen original se anula. El resultado tras el filtrado tiene media nula, es decir, oscilará entre  $+M$  y  $-M$ . Por

eso, se corrige el rango llevándolo de 1 a +M, mediante la función limi.

## Filtrados Especiales:

En los procesos por áreas se pueden reducir efectos indeseados en la imagen, de carácter muy particular. Por eso se ofertan este grupo de filtrados especiales, que tienen por finalidad reducir tipos concretos de ruidos:

**MEDIANA:** el valor de un pixel se calcula como la mediana de toda la vecindad, por lo que si hay ruido impulsivo, equivalente a pixels aislados con valores extremos, este valor no podrá ser la mediana del pixel resultante. Ejemplo:

Imagen original:	Imagen con ruido impulsivo:	
23 27 24 25 30	23 27 24 25 30	Al pasar un filtro de mediana
24 28 25 24 26	24 28 138 24 26	los 2 pixels erróneos tomarán
23 28 28 27 31...	215 28 28 27 31...	valores cercanos al original.

El pixel central de valor 138 pasará a ser 27 tras calcular la mediana de la vecindad 3x3 (27,24,25,28,138,24,28,28,27), valor muy aproximado al original que era 25.

**GAUSIANO:** se genera una máscara correspondiente a un filtro gaussiano simétrico rotacional de tamaño 3x3 y con una anchura de lóbulo principal por defecto de 0.5. Tras filtrar la imagen con este filtro se reduce el ruido blanco gaussiano.

**LAPLACIANO:** se genera una máscara correspondiente a un filtro laplaciano 3x3 con el que se filtra la imagen de la figura activa.

**WIENER:** calcula la media y la desviación estándar de una vecindad de 3x3 así como estima la potencia de ruido aditivo que tiene la imagen. Con estos parámetros filtra por el método Wiener, liberando a la imagen de parte del ruido aditivo de potencia constante.

### Máscara de usuario:

Esta opción permite establecer cualquier máscara y de cualquier dimensión. Para ello hay que introducir la matriz en la ventana de comandos de la siguiente forma:

Máscara de coeficientes: [1 1 1;1 1 1;1 1 1]/9

Una vez familiarizado con los diferentes filtrados se puede empezar con pequeñas variaciones sobre las máscaras que se han comentado, para observar el efecto producido. También es interesante emplear máscaras de 5x5 y hasta de 7x7. Recordar que para no perder la componente continua de la imagen, la suma de los coeficientes deberá ser 1, para que así  $H(0,0)=1$  y al multiplicar por el valor continuo de la señal no afecte.

Nota: en el fichero filtros.txt se incluyen una gran cantidad de máscaras de diferentes tamaños y con una breve explicación del efecto que generan.

## *OPERACIONES GLOBALES*

Son operaciones donde se procesan todos los pixels de la imagen conjuntamente para obtener alguna información de la imagen. La operación global más importante es la Transformada Rápida de Fourier bidimensional (FFT2) de la que se puede extraer la información espectral de la imagen (periodicidades, ruidos,...).

### FFT Bidimensional:

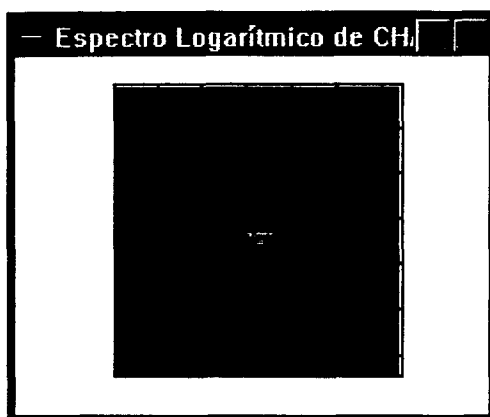
Se calcula con la función de Matlab `fft2`, pero el resultado queda algo confuso si no se desplaza para situar el origen de frecuencias en el centro de la imagen. Eso se lleva a cabo con la función `fftshift`.

La representación de la FFT se puede ver en diferentes formas:

**MÓDULO LINEAL:** el módulo de la FFT con el que estamos acostumbrados a trabajar en una dimensión con señales temporales, será el que nos muestre el espectro de la señal y sus características en frecuencia. La escala lineal puede quedarse algo pequeña para ver una dinámica tan grande como ofrecen estos espectros, en cuyo caso pasaremos a la siguiente opción.

**MÓDULO LOGARÍTMICO:** al pasar el módulo de la FFT a escala logarítmica podremos ver con mayor detalle el espectro general de la imagen. En la ventana de comandos se pregunta el rango dinámico que se empleará en la

visualización y puede ser de algunas decenas de dBs pero se recomiendan 100 dBs, como se ha realizado en el siguiente ejemplo:



**FASE ESPECTRAL:** la fase no muestra una información clara del espectro, pero en cambio contiene mucha más información de la imagen que el módulo. De hecho se puede recuperar una imagen a partir de su fase, prescindiendo del módulo (normalizándolo a uno), mediante una transformada inversa de Fourier bidimensional, pero no se puede hacer del modo contrario, ya que al prescindir de la fase toda la información del módulo se acumula en el origen (acudir a la demostración Ojetivos\Restauración).

## *OPERACIONES GEOMETRICAS*

Las transformaciones geométricas no tienen por objetivo modificar la información de la imagen como venimos haciendo, sino simplemente su aspecto visual.

### Zoom:

Habilita una opción de zoom permanente en la última ventana activa. La forma de hacer un zoom, o acercamiento a la imagen, es pulsar el botón izquierdo del ratón en el punto central donde se desea aproximar y se visualizará la mitad de la imagen anterior con centro donde se indicó. Para alejarse y volver al estado anterior hay que pulsar el botón derecho del ratón.

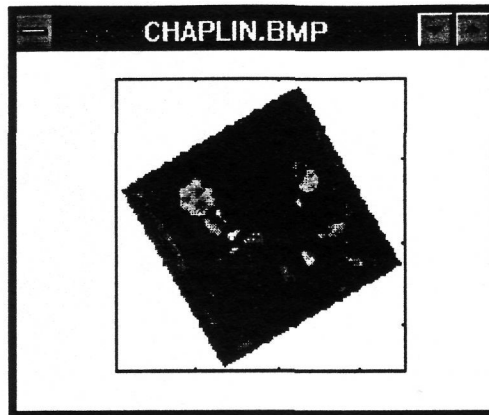
Cuando se activa el zoom se queda activo permanentemente en esa ventana.

Hay una forma de hacer un zoom sobre un área determinada de la imagen, sujetando el botón izquierdo del ratón desde una esquina de la zona y desplazándolo a la otra esquina; al soltar se verá un zoom ajustado a esa zona.

### Girar:

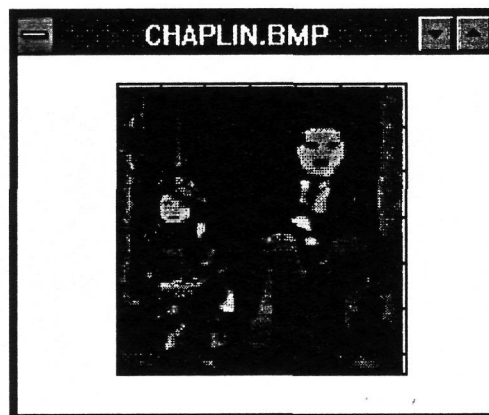
Se indica el ángulo en grados que se desea girar y la imagen aparecerá rotada y con el consiguiente aumento de tamaño. Para el cálculo de los nuevos

pixels se emplea una interpolación bilineal. En el ejemplo, la imagen ha sido rotada 30 grados:



### Espejo:

Se crea una imagen simétrica a la original tomando como eje de simetría un eje vertical. Las columnas de la matriz permutan de izquierda a derecha. Repitiendo la operación, la imagen vuelve a quedar igual.



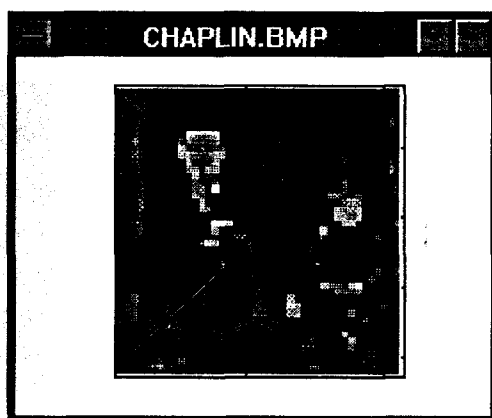
## Tamaño:

Permite modificar el tamaño de la imagen indicando el nuevo valor del número de filas y columnas. De este modo, se puede deformar la imagen y modificar su relación de aspecto ( $n^{\circ}$  filas/ $n^{\circ}$  columnas). Para interpolar los nuevos pixels se emplea la técnica bilineal.

## Diezmado:

Es un caso particular del cambio de tamaño que consiste en desechar filas o columnas, diezmado así la imagen digital. En la ventana de comandos se solicita el factor de diezmado en filas y columnas, referente a cada cuántos pixels nos vamos a quedar con uno de ellos. Un factor de diezmado 2 en cada dirección reduce a la mitad la resolución de la imagen. Se puede modificar la relación de aspecto en valores enteros.

En la foto de ejemplo se ha diezmado por 3 veces tanto filas como columnas:





## Recortar:

Con el ratón sobre la imagen hay que señalar los dos puntos que delimiten el área de la zona que queremos recortar. La información exterior a esa zona se pierde.

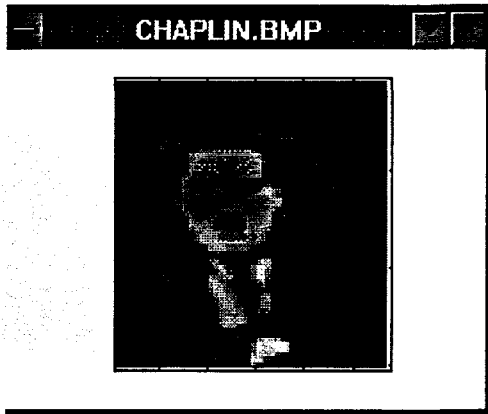


Imagen recortada

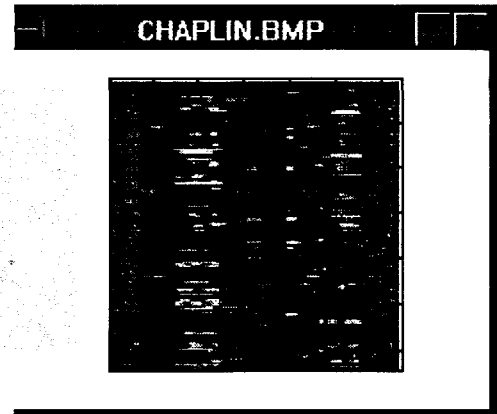


Imagen codificada

## Codificar:

Es un pequeño ejemplo particular de la utilidad de la codificación de imágenes. Al elegir esta opción la imagen desordena la posición de sus filas aleatoriamente, tal y como se hace en una televisión privada, guardando la llave de la codificación realizada en una variable 'orden' que puede ser consultada si más que teclear su nombre en la ventana de comandos.

## Decodificar:

La operación inversa a la anterior, que se encarga de deshacer la reordenación de filas gracias a la variable orden. Si se modifica el valor de la variable (se pierde la llave de la codificación), ya no resulta posible volver a abonarse.

## Ayuda del menú Colores

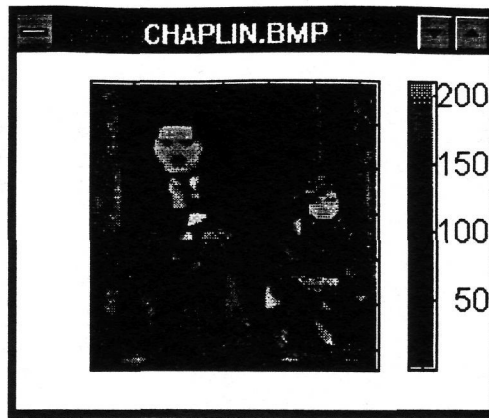
Las opciones de este menú hacen referencia a la paleta de colores que acompaña a toda imagen digital. No modifican en ningún caso los valores de la matriz 'x' de la imagen, excepto cuando se modifica la paleta y se desea mantener intacta la imagen.

El Windows 3.1. sobre el que funciona el matlab, presenta el problema de que con la tarjeta actual de vídeo, sólo soporta los colores correctos de una sólo ventana por lo que el resto de ventanas que no sean la que esté activa, mostrarán una paleta de colores aproximada y muy lejana de la realidad en algunos casos. Para ver bien una imagen hay que activar su ventana sin más que pulsar con el ratón en su interior.

Las opciones que incorpora el menú de colores son:

### Ver Paleta:

Muestra la paleta de la imagen de la última ventana activada junto a la propia imagen, indicando numéricamente los colores empleados. Esta opción es muy útil para redibujar la imagen si el programa ha dado algún error en algún momento, por falta de memoria, por ejemplo, dejando la ventana vacía.



Nota: si por cualquier motivo, tras una operación, el programa quedase colgado con el puntero del ratón en forma de reloj de arena y el ordenador hubiese dejado de acceder al disco duro, se puede pulsar Ctrl+C para abortar la operación, y entonces se puede ir a la opción de Ver Paleta para redibujar la ventana activa, o bien indicar en la ventana de comandos 'imshow(x,map)'.

### Cargar paleta predefinida:

El matlab dispone de un set de paletas predefinidas que pueden ser aplicadas a cualquier imagen con esta opción. El conjunto de paletas y su descripción es:

GRAY: escala de grises.

HSV: todos los colores con máxima saturación y luminancia, variando el tinte.

HOT: mapa cálido de colores.

PINK: mapa rosado de colores.

COOL: mapa sombreado con colores cyan y magenta.

BONE: mapa azulado de colores.

JET: variante del HSV.

COPPER: mapa de tonos de color cobre.

FLAG: colores rojo, blanco, azul y negro alternadamente.

PRISM: colores rojo, naranja, amarillo, verde, azul y violeta repetidamente.

### Cargar paleta de otra imagen:

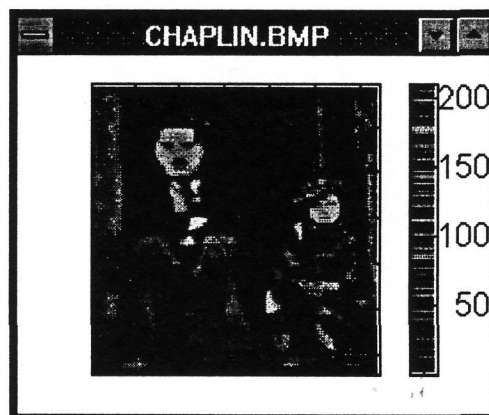
Aplica a la imagen actual el mapa de colores de una imagen grabada. Para ello hay que seleccionar la imagen de la que se tomarán los colores, mediante una ventana de diálogo. Puede ocurrir que el número de colores de ambas imágenes sea distinto y el resultado no sea el esperado. Para consultar cuántos colores tiene la imagen activa se puede acudir a Estadística\Resolución.

## Ordenar paleta:

A menudo los colores de la imagen se encuentran en un orden cualquiera y eso puede traer efectos indeseados a la hora de realizar otras operaciones sobre la imagen. Se dispone de 2 métodos para reordenar la paleta:

*POR LUMINANCIA:* aconsejable si se va a procesar la imagen. Al modificar el orden de los colores, nos vemos obligados a modificar también los valores de la matriz  $x$  para que el resultado visual sea el mismo. Para realizar la ordenación se pasan los valores RGB de la paleta a valores YIQ y tomamos como patrón de ordenación los valores de la primera columna referidos a la luminancia de cada color.

*ALEATORIAMENTE:* se reordenan los colores de la imagen de modo aleatorio e igualmente se modifican los valores de la matriz  $x$  para que no varíe el aspecto de la imagen.



## Reducir el nº de colores:

Muchos mapas de colores tienen colores repetidos aumentando innecesariamente la memoria que requiere la imagen. Es aconsejable ejecutar esta opción para ver su gran utilidad. También se eliminan los colores no empleados por la matriz.

## Cambiar un color:

Esta opción permite seleccionar un punto de la imagen con el ratón y cambiar todos los colores iguales al seleccionado. Hay que advertir que cuando existen colores repetidos no se cambian todos, sino aquellos cuyo valor en la matriz sea el mismo. Para que no ocurra ésto se debe reducir previamente el nº de colores con la opción anterior.

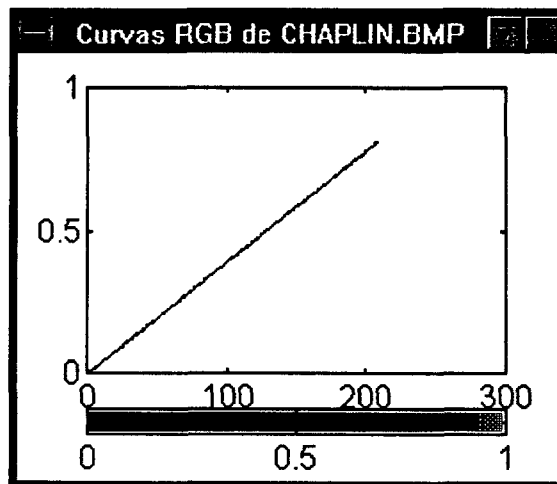
Para seleccionar el color que substituirá al elegido en la imagen, aparece una ventana de diálogo con unos colores básicos y la opción de definir un color personalizado en una carta colorimétrica o mediante valores hsv o rgb numéricamente.

## Destacar desde un color:

Esta opción llama a una aplicación externa que permite seleccionar un color de la imagen y substituye a partir de él, a todos los que estén por encima en la paleta, por un color elegido en una ventana de diálogo de colores igual a la de la opción anterior.

## Curvas RGB:

Crea una nueva ventana con las gráficas de la variación de la intensidad de los colores Rojo, Verde y Azul que va desde 0 hasta 1. Incluye en el eje de abcisas la paleta de colores para ver su equivalencia.



En este caso se ve que las 3 curvas RGB coinciden en una sóla, ya que se trata de una imagen en escala de grises. Se observa que el valor de la luminancia de los colores va de 0 a 1 y que existen algo más de 200 colores en la imagen.



## Ayuda de Estadística

En este apartado se ofrece información de la imagen que puede resultar de interés para realizar algún procesado de la imagen. Si se desea calcular algún otro parámetro estadístico de la matriz o de la paleta de colores, se puede realizar a través de la ventana de comandos del matlab sabiendo que la matriz toma siempre el nombre de 'x' y la paleta de colores 'map'. La información básica que puede necesitarse es:

### Resolución:

Es el número de pixels que tiene la imagen en filas y columnas o el tamaño de la matriz 'x'. En realidad, la resolución debería ser un factor de densidad de muestreo, como son los puntos por pulgada (dpi), pero normalmente se emplea simplemente para el número de pixels. El programa funciona bien con resoluciones inferiores o iguales a 320x200. Un tamaño mayor conllevaría mucho tiempo de cálculo y posiblemente fallos en alguna de las operaciones.

También se muestra en la ventana de comandos, además del número de filas y columnas, el número de colores que se está usando.

En el caso de la imagen con la que se está trabajando en esta ayuda, CHAPLIN.BMP, la resolución es de 128 filas y 128 columnas con 209 colores.

## Valor medio:

Por lo general, las imágenes tienen media no nula, ya que los valores de la matriz  $x$  suelen ser de 1 a 256, para facilitar la equivalencia con el mapa de colores y la media suele rondar en este caso un valor de 120.

En el caso de la imagen del ejemplo, la media es de 80, ya que tiene 209 colores.

## Desviación estándar:

Se calcula con la función de matlab `STD2` ya que se trata de una matriz. El resultado aparece indicado en la ventana de comandos. Este parámetro estadístico no se emplea mucho.

La desviación estándar bidimensional de la imagen de ejemplo `CHAPLIN.BMP` es de 42.

## *Ayuda de Convertir a*

Por lo general, la imagen que se carga es en colores y puede ser interesante convertirla a escala de grises, o bien mostrar sus 3 componentes RGB, HSV o YIQ. Estas son las opciones que presenta este menú de sencillo manejo, ya que no requiere ningún parámetro y los resultados son bastante demostrativos.

### Grises:

El mapa de colores es una matriz de 3 columnas, donde cada una indica la luminancia del rojo, verde y azul, respectivamente. Si multiplicamos a todos los valores de la primera columna por 0.3, los de la segunda por 0.59 y los de la tercera por 0.11 y sumamos las 3 columnas se obtiene una sola columna con la luminancia de cada color que aparece en la imagen. Para que el nuevo mapa de colores sea gris hay que crear una nueva matriz 'map' con 3 columnas iguales a la calculada. Para establecer el mapa de colores se ejecuta 'colormap(map)'.

### RGB:

A partir del mapa de colores de la imagen se crean 3 nuevas imágenes, manteniendo la original, donde cada una de las 3 representa la luminancia del color Rojo, Verde y Azul respectivamente. Para ello en la imagen roja se anulan la segunda y tercera columnas de map, en la imagen verde se anulan la primera y la

tercera columnas y en la imagen azul se hacen cero todos los valores de la primera y la segunda columnas. El nombre de las 3 nuevas figuras es el de la imagen original precedido por la letra R, G y B respectivamente.

## HSV:

El efecto es similar al anterior, ya que se generan 3 nuevas imágenes, pero éstas contienen respectivamente el Tinte de cada uno de los colores (Hue), la Saturación (Saturance) y la Luminancia (Value). La saturación y la luminancia aparecen en grises ya que se establecen 2 nuevas paletas de color donde sus 3 columnas tienen los mismos valores. Pero la figura del tinte aparece con los colores detectados respetando el tinte, pero poniendo la saturación y la luminancia a su valor máximo, es decir, 1.

## YIQ:

Genera 3 imágenes conteniendo la Luminancia (Y), y la crominancia codificada según el sistema americano de TV, NTSC, en componentes I y Q. Las 3 ventanas aparecen en tonos grises y mantienen el nombre de la imagen original añadiendo tan solo las iniciales Y, I y Q respectivamente.

## Ayuda de Demos

Se aconseja pasar primeramente a ver las demostraciones y tras cada una de ellas ir haciendo pruebas con las imágenes que se adjuntan. Para ver las demos sólo hay que seleccionarlas a través de este menú o bien a través de los iconos de Windows. El programa abre otra aplicación llamada *Action Player* sobre la que se pueden visualizar las presentaciones.

El manejo del Action Player es muy intuitivo y no presentará ningún problema. Hay que señalar que el arranque de las demostraciones puede tardar hasta unos 30 seg. ya que debe cargar en memoria la información de las escenas.

Si se desea detener la presentación se puede pulsar en cualquier instante la tecla ESC y se podrá reanudar pulsando el botón del ratón sobre el Play o pulsando Ctrl+Enter.

Para abandonar la presentación hay que ejecutar la opción Exit del menú File.

Hay un pequeño grupo de 6 teclas llamado Control Panel que se pueden pulsar con el ratón aunque también se puede ejecutar la misma orden con el teclado. La correspondencias de teclas y las funciones que desarrollan se encuentran en el menú Control y son:

REWIND	Ctrl+1	Acude al principio de la escena.
BACKSTEP	Ctrl+2	Retrocede medio segundo.
STOP	Ctrl+3	Detiene la presentación.
STEP	Ctrl+4	Avanza medio segundo.
PLAY	Ctrl+5	Continúa con la presentación.
FAST FORWARD	Ctrl+6	Acude al final de la escena actual.

En el menú Control también se puede reiniciar la presentación (Ctrl+Enter) y pasar a otra escena anterior o posterior si la hubiese.

Por último en el menú Window podemos activar o desactivar el control panel y pasar a pantalla completa (Full Screen) durante la visualización.

## *Ayuda de Ayuda*

Cada una de las ayudas que se ofrecen hacen referencia a los distintos menús que aparecen en el programa. Es conveniente consultarlas para conocer qué función está realizando cada una de las opciones de los menús.

Si se desea hacer una consulta más rápida y específica de algún tema concreto, se puede efectuar una búsqueda con el menú Búsqueda\buscar de la parte superior del Bloc de notas. Para aprender a manejar el Bloc de notas también se puede consultar la ayuda disponible de esta aplicación de windows.

En todo caso, las acciones más importantes para desplazarse por las ayudas son ejecutadas por las teclas RePág (subir una página), AvPág (bajar una página) y los cursores. Por último, cuando se desee abandonar la ayuda, acudir al menú Archivo\Salir.

También es importante saber manejar las ventanas. Las ventanas en Windows tienen 3 estados fundamentales: minimizada (se muestra como un pequeño icono), restaurada o normal (se llega pulsando 2 veces seguidas con el ratón sobre el icono), y maximizada que ocupa toda la pantalla. Es preferible trabajar en posición normal ya que así se puede modificar el tamaño y mover de posición fácilmente. Todas las ventanas en windows tienen un recuadro en la esquina superior izquierda que da acceso a un menú donde se puede mover, cambiar el tamaño, maximizar, minimizar o cerrar la ventana activa, así como pasar a otra ventana. Pulsando 2 veces seguidas rápidamente sobre este recuadro, la ventana se cierra. También se puede maximizar a toda la pantalla la ventana activa

pulsando 2 veces en la barra superior (que se ilumina cuando la ventana está activa). Por último, también se puede minimizar o maximizar la ventana con las flechas hacia abajo y arriba que se encuentran en la esquina superior derecha. Si la ventana está maximizada la flecha para maximizar se transforma en una doble flecha para restaurar.



### 3. Listado del programa TIRMA y accesorios.

A continuación se incluye el código fuente del fichero TIRMA.m que es el software de Tratamiento de imágenes programado bajo Matlab v 4.0 y los programas a los que acude y que también se incluyen :

BINARIZA.m

CODIFICA.m

DESCODIF.m

DESTACAR.m

DIEZMA.m

FONDGRAD.m

IMCARGAR.m

IMSALVAR.m

LIMI.m

```

%TIRMA Software de Tratamiento Digital de Imágenes
%           Javi Selvi Nâcher ©1994
%           Trabajo Fin de Carrera
%           EUITT de la ULPGC
%
%
clear all                                     % borra todas l
as variables existentes                       % cierra todas
close (get(0,'Children'));                   % cierra todas
las figuras abiertas

%=====
%           Figura principal
%=====
figure('Units','normalized','Position',[0 0 1 1],...
      'Name','T I R M A','NumberTitle','off','Menubar','none'
);
%=====

nf = 'length(get(0,'Children'))-1';          % número de imá
genes abiertas
abr = ['if eval(nf)<8 [x,map,fich]=imcargar;',...
      'if isstr(fich)',...
      '      set([m12 m2 m3 m4 m5],'Enable','on');',...
      '      figure('Units','normalized','Position',[.
7*rand .7*rand .3 .3],'Name',fich,'NumberTitle','off','M
enubar','none');',...
      '      imshow(x,map);end;end;'];       % abre hasta un
máximo de 8 figuras en posición aleatoria

gcf = 'uf=get(0,'Children');figure(uf(2));'; % coger
figura
gci = 'x=getimage;map=colormap;nom=get(gcf,'Name');';
      % coger imagen

col = [.8 1 1;.8 .8 1;.8 .8 .8;1 .8 .8;1 1 .8;1 1 1;.8 1 1];
orienta = 0;                                % orientación v
ertical
fondgrad;                                    % rellena la ve
ntana Tirma con un gradiente entre 5 colores suaves

axes('Units','normalized','Position',[.38 .6 .6 .4]);
text(0,0,'Tirma','FontName','caligula','FontSize',50,'Color',[0
0 1],'FontAngle','italic');

axis('off')
spinmap(1,10);                               % hace rotar lo
s colores durante 1 seg
set(gca,'Xtick',[],'YTick',[],'Visible','off'); % quita los eje
s
set(1,'Position',[0 .9 1 .1]);              % reduce la ven
tana principal
colormap(gray)

```

```

%=====
%
%                               FICHEROS
%=====

```

```

m1 = uimenu(1,'label','Ficheros','Accelerator','F');

m11 = uimenu(m1,'label','Abrir...',...
    'callback',abr,...
    'Accelerator','A');

m12 = uimenu(m1,'label','Guardar...',...
    'callback',[cogf cogi 'imsalvar(x,map,nom);'],...
    'Accelerator','G');
m13 = uimenu(m1,'label','Borrar...',...
    'callback',[fich,cam]=uigetfile('*.*','Borrar un f
    ichero');delete [cam fich]'],...
    'Accelerator','B');
m14 = uimenu(m1,'Separator','on');
m15 = uimenu(m1,'label','Salir',...
    'callback','close(get(0,'Children')));clear all',...
    'Accelerator','S');

```

```

%=====
%
%                               OPERACIONES PUNTUALES
%=====

```

```

m2 = uimenu(1,'label','Operaciones','Accelerator','O');

m21 = uimenu(m2,'label','Puntuales','Accelerator','P');

uimenu(m21,'label','Histograma',...
    'Accelerator','H',...
    'callback',[cogf cogi 'figure(''Units'', 'normalized'',
    'Position',[.7*rand .7*rand .3 .3],'Name',[ 'Histograma de
    ' nom],'NumberTitle','off','Menubar','none');imhist(x,
    map);]);

uimenu(m21,'label','Negativo',...
    'Accelerator','N',...
    'callback',[cogf cogi 'x=max(x(:))-x+1;imshow(x,map);']
);

uimenu(m21,'label','Posterizar...',...
    'Accelerator','P',...
    'callback',[cogf cogi 'home;nnp=input(''Número de nivel
    es para posterizar: '');'],...
    'x=round(limi(x,1,max(x(:)),1,nnp));map(1:nnp,:)=map(li
    nspace(1,length(map),nnp),:);imshow(x,map);']);

```

```

uimenu(m21,'label','Binarizar...','...
    'Accelerator','B',...
    'callback',[cogf cogi 'home;nii=input('Nivel de binari
zación: ');x=1+(x>nii);map=[0 0 0;1 1 1];imshow(x,map);']);

uimenu(m21,'label','Umbralizar...','...
    'Accelerator','U',...
    'callback',[cogf cogi 'home;nii=input('Nivel de umbral
ización: ');...
    'map=imadjust(map,[map(nii,:);map(length(map),:)],[map(
1,:);map(length(map),:)]);colormap(map);']);

uimenu(m21,'label','Clipping...','...
    'Accelerator','C',...
    'callback',[cogf cogi 'home;nii=input('Nivel inferior:
');nis=input('Nivel superior: ');...
    'map=imadjust(map,[map(nii,:);map(nis,:)],[map(1,:);map
(length(map),:)]);colormap(map);']);

uimenu(m21,'label','Slicing...','...
    'Accelerator','S',...
    'callback',[cogf cogi 'home;nii=input('Nivel inferior:
');nis=input('Nivel superior: ');...
    'map(1:nii,:)=zeros(nii,3);map(nis:length(map),:)=zeros
(length(map)-nis+1,3);colormap(map)']);

uimenu(m21,'label','Ecuallar histograma','...
    'Accelerator','E',...
    'callback',[cogf cogi 'map=histeq(x,map);colormap(map) '
]);

m211 = uimenu(m21,'label','Añadir ruido','...
    'Accelerator','A');
uimenu(m211,'label','Gausiano','...
    'Accelerator','G',...
    'callback',[cogf cogi 'md=input('Media del ruido(0): '
');vr=input('Varianza del ruido(0.01): ');x=ind2gray(x,map);x
=imnoise(x,'gaussian',md,vr);imshow(x,256)']);

uimenu(m211,'label','sal y Pimienta','...
    'Accelerator','P',...
    'callback',[cogf cogi 'md=input('Densidad del ruido(0.
05): ');x=ind2gray(x,map);x=imnoise(x,'salt&pepper',md);imsh
ow(x,256)']);

uimenu(m211,'label','Speckle','...
    'Accelerator','S',...
    'callback',[cogf cogi 'vr=input('Varianza del ruido mu
ltiplicativo(0.04): ');x=ind2gray(x,map);x=imnoise(x,'speckle
',vr);imshow(x,256)']);

```

```

%=====
%
% OPERACIONES LOCALES
%=====

```

```

m22 = uimenu(m2,'label','Locales','Accelerator','L');
m221 = uimenu(m22,'label','Suavizar',...
    'Accelerator','S');
uimenu(m221,'label','Imagen',...
    'Accelerator','I',...
    'callback',[cogf cogi 'h=[1 1 1;1 1 1;1 1 1];h=h/9;x=co
nv2(x,h,'same')];imshow(x,map)']);
uimenu(m221,'label','Contornos',...
    'Accelerator','C',...
    'callback',[cogf cogi 'h=[1 1 1;1 0 1;1 1 1];h=h/8;x=co
nv2(x,h,'same')];imshow(x,map)']);
uimenu(m221,'label','Desenfocar',...
    'Accelerator','D',...
    'callback',[cogf cogi 'h=[1 2 1;2 2 2;1 2 1];h=h/14;x=c
onv2(x,h,'same')];imshow(x,map)']);
m222 = uimenu(m22,'label','Realzar',...
    'Accelerator','R');
uimenu(m222,'label','Imagen',...
    'Accelerator','I',...
    'callback',[cogf cogi 'h=[-1 -1 -1;-1 13 -1;-1 -1 -1];h
=h/4;x=conv2(x,h,'same')];imshow(x,map)']);
uimenu(m222,'label','Contornos',...
    'Accelerator','C',...
    'callback',[cogf cogi 'h=[-1 -1 -1;-1 9 -1;-1 -1 -1];x=
conv2(x,h,'same')];imshow(x,map)']);
m223 = uimenu(m22,'label','Detectar contornos en ',...
    'Accelerator','D');
met = [ '           Elige un método para la detección del conto
rno:
    '
    '1.Sobel = busca extremos relativos en la primera deriv
ada de la intensidad.
    '2.Prewitt = igual pero emplea su propia aproximación a
la derivada.
    '3.Roberts = es similar pero más sensible a contornos d
iagonales.
    '4.Marr-hildreth = busca los cruces por cero del laplac
iano de la imagen tras aplicarle un filtro gaussiano.'];
meth = ['sobel  '; 'prewitt'; 'roberts'; 'marr  '];
uimenu(m223,'label','Todas las direcciones...',...
    'Accelerator','T',...
    'callback',[cogf cogi 'home;disp(met);nm=input('Selecc

```

```

iona 1,2,3 ó 4: '');edge(x,[],meth(nm,:));');
uimenu(m223,'label','Horizontal...',...
        'Accelerator','H',...
        'callback',[cogf cogi 'home;disp(met);nm=input('Selecc
iona 1,2,3 ó 4: '');edge(x,[],meth(nm,:),[0 1]);');
uimenu(m223,'label','Vertical...',...
        'Accelerator','V',...
        'callback',[cogf cogi 'home;disp(met);nm=input('Selecc
iona 1,2,3 ó 4: '');edge(x,[],meth(nm,:),[1 0]);');
uimenu(m223,'label','dirección Abajo-izquierda',...
        'Accelerator','A',...
        'callback',[cogf cogi 'h=[-1 0 0;0 0 0;0 0 1];x=conv2(x
,h,'same');x=round(limi(x,min(x(:)),max(x(:)),1,length(map(:,
1)))));imshow(x,map)']);
uimenu(m223,'label','dirección abajo-Derecha',...
        'Accelerator','D',...
        'callback',[cogf cogi 'h=[0 0 -1;0 0 0;1 0 0];x=conv2(x
,h,'same');x=round(limi(x,min(x(:)),max(x(:)),1,length(map(:,
1)))));imshow(x,map)']);
uimenu(m223,'label','dirección arriba-Izquierda',...
        'Accelerator','I',...
        'callback',[cogf cogi 'h=[0 0 1;0 0 0;-1 0 0];x=conv2(x
,h,'same');x=round(limi(x,min(x(:)),max(x(:)),1,length(map(:,
1)))));imshow(x,map)']);
uimenu(m223,'label','dirección arriba-deRecha',...
        'Accelerator','R',...
        'callback',[cogf cogi 'h=[1 0 0;0 0 0;0 0 -1];x=conv2(x
,h,'same');x=round(limi(x,min(x(:)),max(x(:)),1,length(map(:,
1)))));imshow(x,map)']);
m224 = uimenu(m22,'label','Filtrados especiales',...
        'Accelerator','F');
uimenu(m224,'label','Mediana',...
        'Accelerator','M',...
        'callback',[cogf cogi 'x=medfilt2(x);x=x*length(map(:,1
));imshow(x,map)']);
uimenu(m224,'label','Gausiano',...
        'Accelerator','G',...
        'callback',[cogf cogi 'h=fspecial(''gaussian'');x=conv2
(x,h,'same');imshow(x,map)']);
uimenu(m224,'label','Laplaciano',...
        'Accelerator','L',...
        'callback',[cogf cogi 'h=fspecial(''laplacian'');x=conv
2(x,h,'same');imshow(x,map)']);
uimenu(m224,'label','Wiener',...

```

```

        'Accelerator','W',...
        'callback',[cogf cogi 'x=wiener2(x);x=x*length(map(:,1)
);imshow(x,map)']]);

```

```

uimenu(m22,'label','Máscara de usuario...',...
        'Accelerator','M',...
        'callback',[cogf cogi 'h=input('Máscara de coeficiente
s: ');x=conv2(x,h,'same');imshow(x,map)']]);

```

```

%=====
%
%                                OPERACIONES GLOBALES
%=====

```

```

m23 = uimenu(m2,'label','Globales','Accelerator','G');

```

```

m231 = uimenu(m23,'label','FFT bidimensional',...
        'Accelerator','F');

```

```

uimenu(m231,'label','Módulo lineal',...
        'Accelerator','M',...
        'callback',[ cogf cogi 'x=abs(fftshift(fft2(x)));x=roun
d(limi(x,min(x(:)),max(x(:)),1,256));',...
        'figure(''Units'', 'normalized', 'Position',[
.7*rand .7*rand .3 .3], 'Name', ['Espectro Lineal de ' nom], '
NumberTitle', 'off', 'Menubar', 'none');',...
        'map=bone(256);imshow(x,map)']]);

```

```

uimenu(m231,'label','módulo Logarítmico...',...
        'Accelerator','L',...
        'callback',[ cogf cogi 'x=abs(fftshift(fft2(x)));md=inp
ut('Margen dinámico en dB (100): ');',...
        'x=x/max(x(:));x=20*log10(x+(10^(-md/20)));x=rou
nd(limi(x,min(x(:)),max(x(:)),1,256));',...
        'figure(''Units'', 'normalized', 'Position',[
.7*rand .7*rand .3 .3], 'Name', ['Espectro Logarítmico de ' n
om], 'NumberTitle', 'off', 'Menubar', 'none');',...
        'map=bone(256);imshow(x,map)']]);

```

```

uimenu(m231,'label','Fase',...
        'Accelerator','F',...
        'callback',[ cogf cogi 'x=angle(fftshift(fft2(x)));x=rou
nd(limi(x,min(x(:)),max(x(:)),1,256));',...
        'figure(''Units'', 'normalized', 'Position',[
.7*rand .7*rand .3 .3], 'Name', ['Fase espectral de ' nom], '
NumberTitle', 'off', 'Menubar', 'none');',...
        'map=bone(256);imshow(x,map)']]);

```

```

%=====
%
%                                OPERACIONES GEOMÉTRICAS
%=====

```

```

m24 = uimenu(m2,'label','geoMétricas','Accelerator','M');

uimenu(m24,'label','Zoom',...
       'Accelerator','Z',...
       'callback',[cogf 'imzoom on']);

uimenu(m24,'label','Girar...',...
       'Accelerator','G',...
       'callback',[cogf cogi 'an=input('Ángulo de rotación: '
');',...
                  'x=imrotate(x,an,'bilinear');imshow(x
,map)']);
uimenu(m24,'label','Espejo',...
       'Accelerator','E',...
       'callback',[cogf cogi 'x=fliplr(x);imshow(x,map)']);

uimenu(m24,'label','Tamaño...',...
       'Accelerator','T',...
       'callback',[cogf cogi 'home;nf=input('Nuevo número de
filas: ');nc=input('Nuevo número de columnas: ');x=imresize(
x,[nf nc],'bilinear');imshow(x,map)']);

uimenu(m24,'label','Diezmado...',...
       'Accelerator','D',...
       'callback',[cogf cogi 'home;nf=input('Factor de diezma
do en filas: ');nc=input('Factor de diezmado en columnas: ');
;x=diezma(x,nf,nc);imshow(x,map)']);

uimenu(m24,'label','Recortar...',...
       'Accelerator','R',...
       'callback',[cogf cogi '[p1 p2]=ginput(2);x=x(min(p2):ma
x(p2),min(p1):max(p1));imshow(x,map)']);

uimenu(m24,'label','Codificar',...
       'Accelerator','C',...
       'callback',[cogf cogi '[x,orden]=codifica(x);imshow(x,m
ap)']);

uimenu(m24,'label','dEcodificar',...
       'Accelerator','E',...
       'callback',[cogf cogi 'x=decodif(x,orden);imshow(x,map)
']);

```

```

%=====
%                                COLORES
%=====

```

```

m3 = uimenu(1,'label','Colores','Accelerator','C');

m31 = uimenu(m3,'label','Ver paleta',...

```



```

        'Accelerator','V',...
        'callback',[cogf 'colorbar']]);

maps = str2mat('gray','hsv','hot','pink','cool','bone','jet','c
opper','flag','prism');

m32 = uimenu(m3,'label','cArgar paleta predefinida',...
    'Accelerator','A');

for i = 1:size(maps,1);
    uimenu(m32,'label',maps(i,:), 'callback',[cogf 'colormap(' ma
ps(i,:) ')']);
end

m33 = uimenu(m3,'label','cargar Paleta de otra imagen...',...
    'Accelerator','P',...
    'callback',[ '[x,map,fich]=imcargar;' cogf 'colormap(map
)']]);

m34 = uimenu(m3,'label','Ordenar paleta',...
    'Accelerator','O');

uimenu(m34,'label','por Luminancia',...
    'Accelerator','L',...
    'callback',[cogf cogi 'ntsc=rgb2ntsc(map);[dum,order]=s
ort(ntsc(:,1));',...
    '[x,map]=cmpermute(x,map,order);imshow(x,map)
;']]);

uimenu(m34,'label','Aleatoriamente',...
    'Accelerator','A',...
    'callback',[cogf cogi '[x,map]=cmpermute(x,map);imshow(
x,map);']]);

m35 = uimenu(m3,'label','Reducir el nº colores',...
    'Accelerator','R',...
    'callback',[cogf cogi '[x,map]=cmunique(x,map);imshow(x
,map);']]);

men1 = 'Señala un color seleccionando un punto de la imagen con
el ratón';

m36 = uimenu(m3,'label','Cambiar un color...',...
    'Accelerator','C',...
    'callback',[cogf cogi 'disp(men1);',...
    '[pu1,pu2]=ginput(1);x=limi(x,min(x(:)),max(x(:)),1,len
gth(map));map(x(pu2,pu1),:)=uisetcolor;',...
    'colormap(map)']]);

m37 = uimenu(m3,'label','Destacar desde un color...',...
    'Accelerator','D',...
    'callback',[cogf cogi 'disp(men1);[pu1,pu2]=ginput(1);x
=limi(x,min(x(:)),max(x(:)),1,length(map));destacar(x(pu2,pu1)
;']]);

```

```

m38 = uimenu(m3,'label','curvas Rgb',...
            'Accelerator','R',...
            'callback',[cogf 'map=colormap;figure(''Units'', ''norma
lized'', ''Position'', [.7*rand .7*rand .3 .3], ''Name'', [''Curvas
RGB de '' nom], ''NumberTitle'', ''off'', ''Menubar'', ''none'');'
...
            'colormap(map);rgbplot(map);colorbar('
'horiz'')' ]]);

```

```

%=====
%                               ESTADÍSTICA DE LA IMAGEN
%=====

```

```

m4 = uimenu(1,'label','Estadística','Accelerator','E');

```

```

uimenu(m4,'label','Resolución',...
        'Accelerator','R',...
        'callback',[cogf cogi '[a,b]=size(x);disp([''Resolución
de '',nom,': ',int2str(a),'' filas y ',int2str(b),'' column
as'']);disp([''Número de colores: ',int2str(length(map(:,1)))
''])];

```

```

uimenu(m4,'label','Valor medio',...
        'Accelerator','V',...
        'callback',[cogf cogi 'disp([''Valor medio de '',nom, ''
: ',int2str(mean2(x))''])];

```

```

uimenu(m4,'label','Desviación estándar',...
        'Accelerator','D',...
        'callback',[cogf cogi 'disp([''Desviación estándar bidi
mensional de '',nom,': ',int2str(std2(x))''])];

```

```

%=====
%                               CONVERTIR IMAGEN
%=====

```

```

m5 = uimenu(1,'label','conVertir a','Accelerator','V');

```

```

uimenu(m5,'label','Grisés',...
        'Accelerator','G',...
        'callback',[cogf cogi 'map=rgb2gray(map);colormap(map)'
]);

```

```

uimenu(m5,'label','RGB',...
        'Accelerator','R',...
        'callback',[cogf cogi 'figure(''Units'', ''normalized'',
''Position'', [.7*rand .7*rand .3 .3], ''Name'', [''R-' ' nom], ''Nu

```

```

mberTitle','off','Menubar','none');',...
    'map2=map;map2(:,2)=zeros(length(map),1);map2(:,
,3)=zeros(length(map),1);imshow(x,map2);',...
    'figure('Units','normalized','Position',[
.7*rand .7*rand .3 .3],'Name',['G-' nom],'NumberTitle','
off','Menubar','none');',...
    'map2=map;map2(:,1)=zeros(length(map),1);map2(:,
,3)=zeros(length(map),1);imshow(x,map2);',...
    'figure('Units','normalized','Position',[
.7*rand .7*rand .3 .3],'Name',['B-' nom],'NumberTitle','
off','Menubar','none');',...
    'map2=map;map2(:,1)=zeros(length(map),1);map2(:,
,2)=zeros(length(map),1);imshow(x,map2);clear map2']]);

uimenu(m5,'label','HSV',...
    'Accelerator','H',...
    'callback',[cogf cogi 'map=rgb2hsv(map)'];',...
    'figure('Units','normalized','Position',[
.7*rand .7*rand .3 .3],'Name',['Tinte ' nom],'NumberTitle'
','off','Menubar','none');',...
    'map2=map;map2(:,2)=ones(length(map),1);map2(:,
3)=ones(length(map),1);map2=hsv2rgb(map2);imshow(x,map2);',...
    'figure('Units','normalized','Position',[
.7*rand .7*rand .3 .3],'Name',['Saturación ' nom],'NumberT
itle','off','Menubar','none');',...
    'map2=map;map2(:,1)=map(:,2);map2(:,3)=map(:,2)
;imshow(x,map2);',...
    'figure('Units','normalized','Position',[
.7*rand .7*rand .3 .3],'Name',['Luminancia ' nom],'NumberT
itle','off','Menubar','none');',...
    'map2=map;map2(:,1)=map(:,3);map2(:,2)=map(:,3)
;imshow(x,map2);clear map2']]);

uimenu(m5,'label','YIQ',...
    'Accelerator','Y',...
    'callback',[cogf cogi 'map=rgb2ntsc(map)'];',...
    'figure('Units','normalized','Position',[
.7*rand .7*rand .3 .3],'Name',['Y-' nom],'NumberTitle','
off','Menubar','none');',...
    'map2=map;map2(:,2)=map(:,1);map2(:,3)=map(:,1)
;imshow(x,map2);',...
    'figure('Units','normalized','Position',[
.7*rand .7*rand .3 .3],'Name',['I-' nom],'NumberTitle','
off','Menubar','none');',...
    'map2=map;map2(:,1)=map(:,2);map2(:,3)=map(:,2)
;map2=limi(map2,min(map2),max(map2),0,1);imshow(x,map2);',...
    'figure('Units','normalized','Position',[
.7*rand .7*rand .3 .3],'Name',['Q-' nom],'NumberTitle','
off','Menubar','none');',...
    'map2=map;map2(:,1)=map(:,3);map2(:,2)=map(:,3)
;map2=limi(map2,min(map2),max(map2),0,1);imshow(x,map2);clear m
ap2']]);

```

```
%=====
%
%                                DEMOSTRACIONES
%=====
```

```
m6 = uimenu(1,'label','Demos','Accelerator','D');

uimenu(m6,'label','Introducción',...
       'Accelerator','I',...
       'callback','!c:\action25\playact c:\action25\intro1.act
');

uimenu(m6,'label','Objetivos',...
       'Accelerator','O',...
       'callback','!c:\action25\playact c:\action25\intro2.act
');

uimenu(m6,'label','Histograma',...
       'Accelerator','H',...
       'callback','!c:\action25\playact c:\action25\intro3.act
');

uimenu(m6,'label','Filtrado',...
       'Accelerator','F',...
       'callback','!c:\action25\playact c:\action25\intro4.act
');
```

```
%=====
%
%                                AYUDA
%=====
```

```
m7 = uimenu(1,'label','Ayuda','Accelerator','A');

uimenu(m7,'label','Ficheros',...
       'Accelerator','F',...
       'callback','!notepad tirma.a1');

uimenu(m7,'label','Operaciones',...
       'Accelerator','O',...
       'callback','!notepad tirma.a2');

uimenu(m7,'label','Colores',...
       'Accelerator','C',...
       'callback','!notepad tirma.a3');

uimenu(m7,'label','Estadística',...
       'Accelerator','E',...
       'callback','!notepad tirma.a4');

uimenu(m7,'label','conVertir a',...
       'Accelerator','V',...
       'callback','!notepad tirma.a5');
```

```

        'Accelerator','V',...
        'callback','!notepad tirma.a5');

uimenu(m7,'label','Demostraciones',...
        'Accelerator','D',...
        'callback','!notepad tirma.a6');

uimenu(m7,'label','Ayuda',...
        'Accelerator','A',...
        'callback','!notepad tirma.a7');

uimenu(m7,'label','Acerca de',...
        'Accelerator','A',...
        'Separator','on',...
        'callback','acerca2');

```

```

%=====

```

```

set([m12 m2 m3 m4 m5],'Enable','off');           % comienza con
ciertos menús desactivados
pause(0)

%end TIRMA

```

```

function y=binariza(X,map)
%BINARIZA es una función que permite crear una imagen en blanco
  y negro
%   a partir de un umbral en la paleta de colores de la ima
gen. Este
%   umbral se selecciona con el ratón sobre la misma paleta
  de colores.
%
%   Uso:   y=binariza(X,map), en y devuelve la imagen bina
ria
%

imshow(X,map)
colorbar;

[i,j] = ginput(1);
[a,b] = size(X);
n = j*size(map)/b;

for f=1:a
    for c=1:b
        y(f,c)=eval(X(f,c)<n);
    end
end
end;

```

```

function [y,orden] = codifica(X)
%CODIFICA toma la imagen de la matriz X y reordena sus
%   filas de manera aleatoria pero devolviendo en
%   el vector orden la nueva situación de las filas.
%
%   Uso:      [y,orden] = codifica(X)
%
[f,c] = size(X);
orden = randperm(f);
for fila = 1:f
    y(fila,:) = X(orden(fila),:);
end;

end

```

```

function x = decodif(y,orden)
%DECODIF reordena la matriz y según la distribución de filas
%      establecida en el vector orden y devuelve la imagen
%      decodificada como x.
%
%      Uso:      x = decodif(y,orden)
%

[f,c] = size(y);
for fila = 1:f
    x(orden(fila),:) = y(fila,:);
end;

end

```



```
function destacar(n)
%DESTACAR es una breve aplicación para resaltar los valores
% de una imagen que superen cierto valor n.
% Uso: destacar(n)

map = colormap;
c=uisetcolor('Escoge un color');
for i = n:length(map)
    map(i,:)=c;
end;
colormap(map)
end
```

```

function y = diezma(x,m,n)
%DIEZMA toma las muestras múltiplos de n en las filas,
%      y las múltiplos de m en las columnas dando lugar
%      a una reducción de la frecuencia de muestreo.
%
%      Uso:   y=diezma(x,m,n)

[a,b] = size(x);

for f=n:n:a
    for c=m:m:b
        y(f/n,c/m)=x(f,c);
    end
end;

end

```

```

% FONDGRAD rellena completamente el fondo de la ventana activa
% con un gradiente entre los colores que se especifiqu
en
% en la matriz col. Si no se inicializa col, el progra
ma
% permite seleccionar 2 colores para la transición de
% uno a otro. La orientación del gradiente se puede in
dicar
% con la variable orienta (1 es vertical y no 1 es hor
izantal)
% Si no se especifica el programa lo solicita por menú
.
%
%

if ~exist('orienta') orienta=menu('ORIENTACION','Vertical','Hor
izantal');
end

if ~exist('col') col(1,:)=uigetcolor('Selecciona el primer colo
r');
col(2,:)=uigetcolor('Selecciona el segundo col
or');
end

l = size(col);
l = l(1);

a(1:256,1) = interp1(linspace(1,256,l),(col(:,1))',1:256);
a(1:256,2) = interp1(linspace(1,256,l),(col(:,2))',1:256);
a(1:256,3) = interp1(linspace(1,256,l),(col(:,3))',1:256);

colormap(a);
b = [1:256];
if orienta == 1 % orientación vertical
    b = b';
end;
axes('Units','normalized',...
'Position',[0 0 1 1]);
image(b)

clear a b orienta col

end

```

```

function [X,map,fich] = imcargar
%IMCARGAR carga una imagen del disco en los formatos
%      bmp, gif, pcx, tiff, raw.
%      No necesita parámetros de entrada.

[fich,cam] = uigetfile('*.bmp;*.gif;*.pcx;*.tif','Cargar una im
agen');

if (isstr(cam))&(isstr(fich))

    ext = lower(fich(length(fich)-2:length(fich)));
    if ext == 'raw'
        [X,map] = imread([cam fich])
    else
        if ext=='tif' ext='tiff';end
        eval(['[X,map]=' ext 'read('' ' cam fich ' '');
'])
    end;
else
    disp('No se ha cargado ninguna imagen');
end;
end

```

```

function imsalvar(X,map,fi)
%IMSALVAR salva la imagen en los formatos
%      bmp, gif, pcx, tiff, raw.
%      No necesita parámetros de entrada.

if ~exist('fi') fi='*.bmp,*.gif,*.pcx,*.tif';end

[fich,cam] = uiputfile(fi,'Salvar una imagen');

if (isstr(cam))&(isstr(fich))

    ext = lower(fich(length(fich)-2:length(fich)));
    if ext == 'raw'
        imwrite(X,map,[cam fich])
    else
        if ext=='tif' ext='tiff';end
        eval([' ext 'write(X,map,' ' cam fich ' ');'])
    end;
else
    disp('No se ha salvado ninguna imagen');
end;
end

```

```
function y = limi(x,xmin,xmax,ymin,ymax)
%LIMI escala de forma lineal los valores de una matriz x
%      comprendidos entre xmin y xmax hacia nuevos valores
%      entre ymin y ymax.
%      Uso:      y=limi(x,xmin,xmax,ymin,ymax)

y = (ymax-ymin)*(x-xmax)/(xmax-xmin) + ymax;

end
```



## 4. ACTION v 2.5. como herramienta de presentación interactiva

### ¿Por qué ACTION para una presentación interactiva?

Se ha elegido este programa para realizar las presentaciones de los procesados realizados sobre las imágenes y para mostrar a modo de documental las posibilidades de estas nuevas técnicas, por la flexibilidad que ofrece en la edición y porque es capaz de combinar textos, imágenes, animaciones, sonidos y en un entorno interactivo.

Este software da pie a una nueva modalidad de producción, especialmente en el área de presentaciones, ya que permite la interactividad con la persona que recibe la información que se pretende mostrar. El espectador deja de ser pasivo y pasa a formar parte de lo que está viendo, ya que elige el camino por el que se desarrolla la presentación. Se puede volver a ver un módulo o salir de él para comprobar con las herramientas y con las imágenes que se adjuntan , el tipo de tratamiento que acaba de ver con una imagen de ejemplo. Puede saltar de un módulo a otro y buscar algún concepto de los que se entrelazan a lo largo del documental.

La forma de incorporar los diferentes objetos a escena es muy variada para darle más dinamismo a la presentación. Aunque en módulos similares se puede emplear las mismas formas de entrada de objetos para crear ciertas equivalencias entre ellos.

La presentación puede ser detenida para volver a ver las últimas secuencias o acudir al principio o al final de la escena, al igual que un magnetoscopio reproductor, de forma muy intuitiva para el usuario, con las teclas que a continuación se muestran.

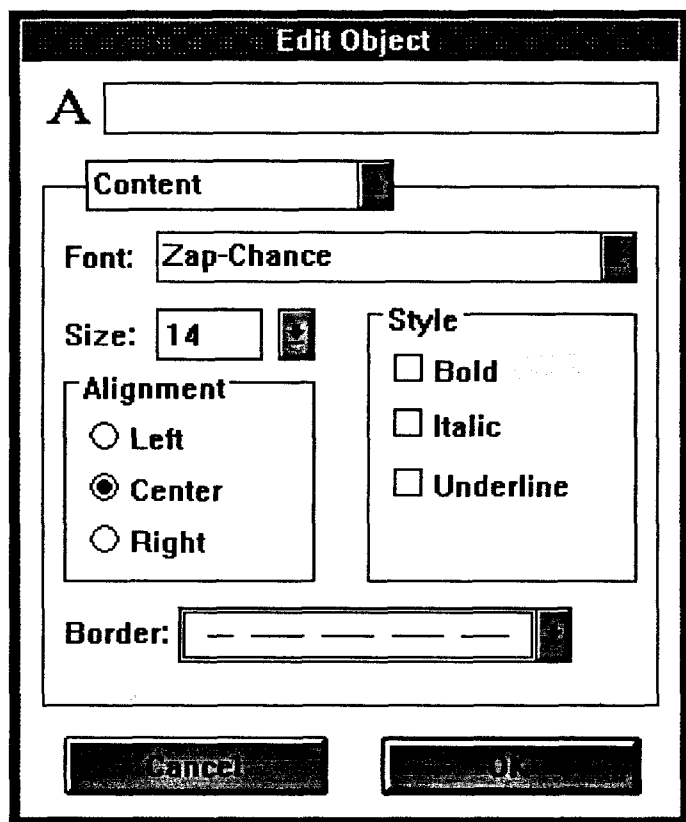


La conexión entre el programa y el usuario se realiza principalmente a través del ratón, con el que se pueden pulsar en cualquier momento estas teclas y también los botones que vayan apareciendo. Con el teclado también se pueden realizar todas estas funciones, como se indicará en el manual de manejo.

### Los objetos en ACTION:

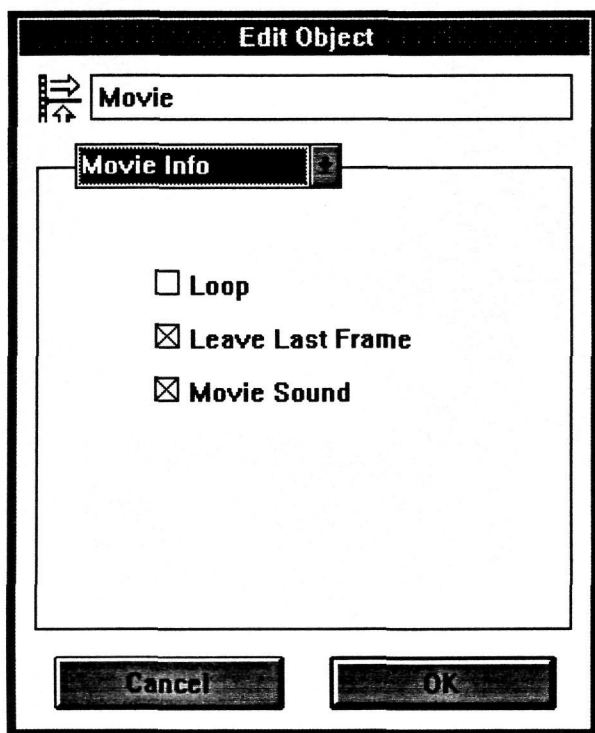
El Texto que aparece explicando los procesos que sufren las imágenes procura ser breve y conciso permitiendo al alumno al que va destinado el documental, leer con rapidez el mensaje fundamental y centrar la atención en

las imágenes, para comprobar que efectivamente sucede lo que se comenta. Action permite la aparición del texto en cualquier modalidad al igual que con el resto de objetos. Dispone de todos los tipos de letra que se encuentran instalados en Windows, en formato normal, cursiva, subrayado o negrita, y en tamaños escalables. Para disponer de tipos de letras





adecuados al documental que se destinan, se ha realizado en el ordenador del Laboratorio de TVII una instalación de un conjunto de True Type Fonts (TTF's) localizados en el CD-ROM que se adjunta como suplemento en el nº3 de la revista Tech Multimedia News.

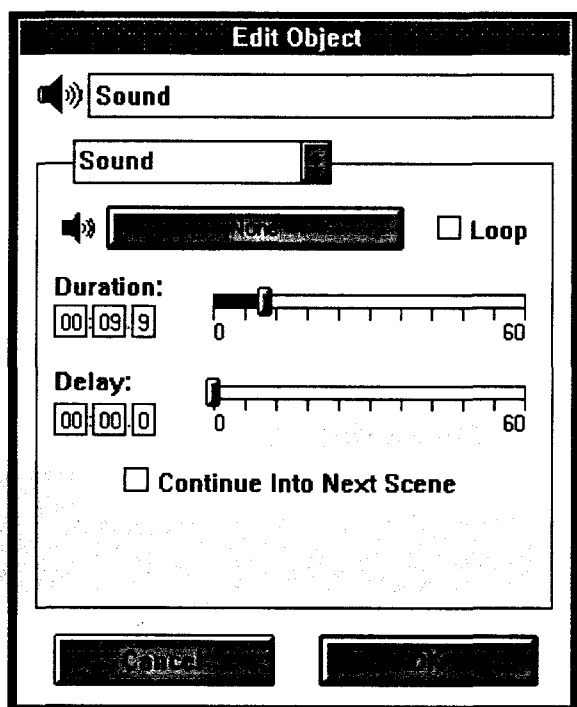


Las Imágenes que emplea Action son del tipo BMP o DIB. Se permite la posibilidad de redimensionarlas y situarlas en cualquier posición de la pantalla. El formato empleado para la inclusión de imágenes ha sido el BMP (Mapa de Bits), estándar en aplicaciones bajo Windows. Uno de los problemas que han surgido a la hora de exportar las imágenes

generadas con Matlab hacia Windows ha sido que la rutina encargada de grabar imágenes en este formato en Matlab crea un desplazamiento en dirección horizontal de cada línea de la imagen y resulta distorsionada. Se buscaron varias soluciones a este problema, como fueron: grabar la imagen en otro formato como PCX y convertirlas a BMP posteriormente con el programa Shareware incluido GWS; incluir la imagen en el portapapeles de Windows (clipboard) con la opción de Copy to Bitmap que incluyen todas las figuras generadas con Matlab y luego recuperarla (Paste) con algún programa de tratamiento de imágenes como Photo Styler o Corel Photo Paint, para grabarla directamente en formato BMP; pero finalmente la solución adoptada fue trabajar con un programa shareware, GrabIt Pro, que queda residente en memoria y es capaz de

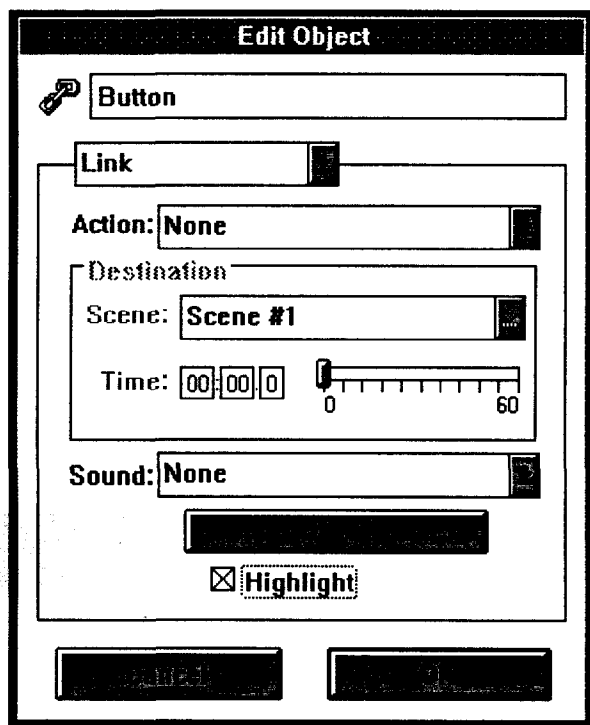
capturar pantallas bajo Windows, como las figuras que genera Matlab, y grabarlas luego directamente en formato BMP. También se incluyen ficheros Metafile que son imágenes vectoriales que dan muy buen resultado cuando se modifica su tamaño. Matlab permite la posibilidad de pegar en el portapapeles ficheros Metafile (Copy to Metafile) y se empleó esta licencia para exportar las gráficas.

Las Animaciones que se pueden mostrar en Action pueden ser de formatos AVI (Microsoft Video), MMM (MacroMind Director), MOV (QuickTime), FLI o FLC (Autodesk Animator). Se incluyen animaciones de diferentes ejemplos en formatos AVI y FLC; estas últimas no permiten una edición muy exacta a la hora de situar los puntos de entrada y salida, pero basta con dejar un pequeño margen de algunos segundos para evitar posibles problemas de solapamiento con objetos previos y posteriores.



Los Sonidos en Action pueden aparecer en cualquier instante o asociarse a algún evento mediante un enlace (Link) y aparecer junto a él. Los formatos de sonidos soportados en Action son WAV, MIDI o incluso se puede leer música grabada en el CD-ROM al igual que en un CD-Audio, pero se ha rechazado esta opción para evitar que la música desaparezca cuando

se precisa leer un fichero del CD-ROM, ya que la tecnología empleada para la grabación no es la moderna XA, ni el lector de CD-ROM disponible soporta esta doble lectura de datos y sonido simultánea. Se acompaña música digitalizada al documental, así como sonidos enlazados a la pulsación de algunas teclas. Hay ficheros MIDI que contienen las notas de las canciones, acompañadas del instrumento que se emplea, la duración de la nota y otras características; estos ficheros ocupan poco espacio en el disco pero tienen una calidad bastante baja además del limitado número de instrumentos (128 como máximo, que siempre suenan del mismo modo). Por contra los ficheros WAV tienen una calidad mucho mayor ya que contienen la música real digitalizada alcanzando la cota de calidad de un CD-Audio (44.1 Mhz de frecuencia de muestreo y 16 bits de cuantificación), aunque ocupando un elevado espacio en el disco. Están especialmente indicados para contener los sonidos de corta duración que se enlazan con determinados eventos como pulsaciones de teclas.



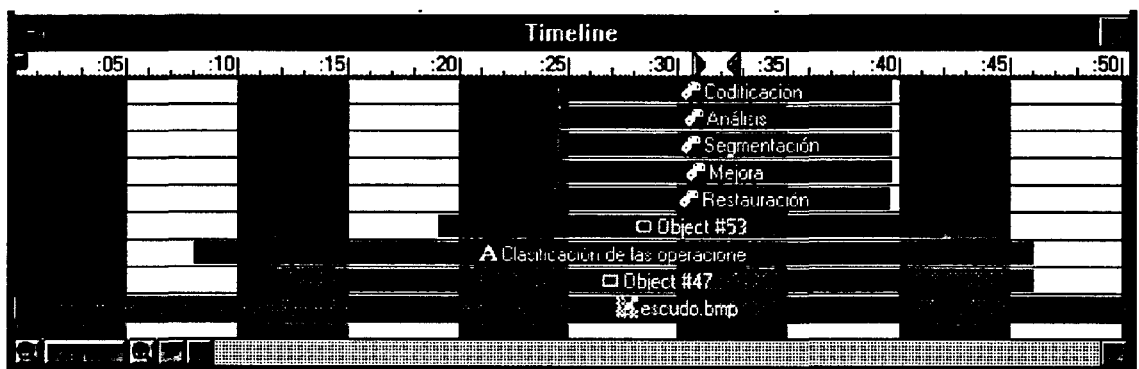
La Interactividad viene proporcionada por los botones que pueden desempeñar diversas funciones. Fundamentalmente se emplean para desplazarnos a algún punto concreto del documental, bien en el mismo módulo o en otro. Pueden tener asociado algún sonido como se ha llevado a cabo en las presentaciones.

Al situar un botón hay que

establecer un bucle para que en la reproducción el programa quede a la espera de la pulsación. Los bucles (loop) se generan con la posición de comienzo y la duración y dejan al programa repitiendo ese fragmento seleccionado hasta que se pulse en alguno de los objetos que tengan un enlace con otro punto de la escena o con otra escena.

### La edición de las escenas:

Para llevar a cabo la edición de una escena en Action se hace uso de la ventana de tiempos (TimeLine) donde cada objeto se representa como una caja donde su longitud es proporcional a su duración. Las cajas tienen 3 divisiones y se refieren respectivamente al tiempo de entrada (enter), al tiempo de permanencia (hold) y el tiempo de salida (exit).



Existe un código de colores para distinguir los diferentes objetos y además éstos van acompañados de un pequeño carácter delante identificativo del tipo de objeto (texto, imagen, sonido, animación o botón). En un eje temporal horizontal situado en la parte superior del Timeline se puede comprobar el momento de la entrada de cada objeto así como su duración de entrada, permanencia y salida.

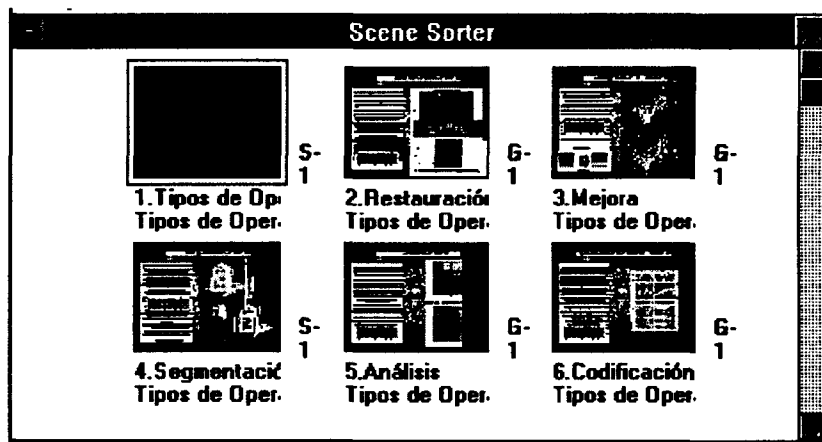
La situación de los objetos en el eje vertical indica la preferencia a la hora de superponerse en escena, de modo que al escudo.bmp se le pueden superponer todos los demás.

Hay que aclarar que esta parte del programa no es la que se emplea para visualizar las presentaciones, sino para diseñarlas. Es por esto que esta ventana sólo es accesible desde el programa action.exe y no desde el playact.exe encargado de la visualización exclusivamente.

Aunque en principio las características temporales de cada objeto se definen analíticamente en otra ventana (Edit Object) donde se indica la entrada, mantenimiento, salida, duración, contenido, color, tinta y sombras, sonido y enlace, estas pueden ser modificadas en la ventana Timeline observando su posición con respecto al resto de objetos. Así se consigue una edición bastante equilibrada en la estancia de los objetos en pantalla.

### El montaje de las escenas:

Una presentación consta de varias escenas que por lo general se suceden encadenadas. Pero se puede efectuar un montaje con las escenas confiriendo a la



presentación de interactividad con el usuario. Para ello se emplean los botones, que enlazan diferentes puntos entre escenas, enlaces

con el resto de objetos , de modo que puede establecerse un salto a un determinado punto de la escena al pulsar el ratón sobre una imagen o un texto, y por último en la finalización de cada escena se elige una modalidad para continuar la presentación que puede ser acudir a otra escena o repetir la misma escena.

En esta ventana (Scene Sorter) se muestra de modo aproximado las pantallas repletas de la información de cada una de las escenas y la transición entre ellas. Se aprecia que todas las escenas dependen de la primera Tipos de Operaciones y vuelven a ella al finalizar bien con un Goto (G-1) o un Skip (S-1).

## **5. Demostraciones interactivas.**

Se han preparado 4 demostraciones con una duración total de unos 40 minutos que muestran los conceptos básicos del tratamiento digital de imágenes. La herramienta para confeccionar estas presentaciones ha sido Action versión 2.5. comentada en el capítulo anterior. Para acceder a las demostraciones, se puede hacer desde el programa TIRMA o desde el windows, accediendo directamente sobre sus iconos.

El contenido y las particularidades de cada demostración son las siguientes:

### **1. Introduccion::**

Es la primera de las demostraciones. Se divide en 4 escenas que son:

1.Introducción: a partir de una imagen de barras de colores del generador del laboratorio de TVII, GEN9560 y capturada con la tarjeta de adquisición de imágenes Targa+, se estudiaron los componentes RGB de cada uno de los 8 colores, a partir de un cálculo estadístico efectuado con matlab. No quiere decir que estos sean los valores exactos de la luminancia en componentes RGB de los colores de las barras, ya que interviene la ganancia en cada componente que se establezca en la Targa+, pero es muy aproximado.El resultado fue la siguiente matriz de colores:

COLOR	Componente R	Componente G	Componente B
Blanco	1	1	1
Amarillo	0.8627	0.8627	0
Cyan	0	0.8627	0.8824
Verde	0	0.8627	0
Violeta	0.8824	0	0.8627
Rojo	0.8627	0	0
Azul	0	0	0.8627
Negro	0	0	0

Con estos valores se genera una matriz de colores en matlab que es visualizada como imagen tridimensional desde diferentes puntos de vista, construyendo así una animación en la que se intenta mostrar el carácter bidimensional de las imágenes digitales.



Se compara la señal analógica de barras de colores con una digital que tiene una rejilla superpuesta a modo de muestras.

2.Colores: se separa la imagen en color en sus 3 imágenes de Rojo, Verde y Azul, mostrando la actuación de la luminancia de cada color básico en los 8 colores de las barras.

3. Amplitud discreta: discretizar la amplitud en una imagen equivale a reducir el número de colores. A partir de una misma imagen se va reduciendo paulatinamente el número de niveles de cuantificación para mostrar el efecto de reducción de colores.

4.Discretización espacial: de modo análogo a la escena anterior, se muestra el efecto de reducir los niveles de cuantificación sobre una misma imagen.

La demostración termina con una pequeña animación en color de la última imagen empleada.

## 2.Operaciones:

Esta es la demostración más larga, con cerca de 14 minutos, y presenta mediante un menú inicial, una clasificación de las operaciones que se pueden realizar sobre las imágenes digitales según el objetivo que se persigue. Se divide en las siguientes escenas: Restauración, Mejora, Segmentación, Análisis y Codificación.

1. Restauración: se añade un ruido periódico a una imagen y se muestra el espectro para comprobar el camino que se sigue en las operaciones de restauración. Por último se restaura la imagen de una forma sencilla.

2. Mejora: las técnicas de mejora son muy variadas, por lo que se busca un ejemplo sencillo de modificación del contraste y del brillo.

3. Segmentación: se realiza una segmentación por luminancia y se emplea el resultado para hacer un cambio de color de una zona de la imagen.

4. Análisis: como muestra de un análisis se realiza una simulación de reconocimiento de caracteres.

5. Codificación: se ha programado una opción en el programa TIRMA especialmente pensada para esta demostración, en la que se simula la codificación de la imagen por un método similar al que emplea una de nuestras televisiones privadas (Canal+).

### 3. Histograma:

Esta es la demostración que goza de mayor interactividad. Su duración es de unos 9 minutos, pero depende del camino que se siga en la visualización. Está compactada en una sola escena.

Se presentan algunas de las más importantes operaciones puntuales.

Existe un botón en el título de la demostración que transporta inmediatamente al menú principal. Tras la presentación de las operaciones de Negativo, Ecuilización del histograma, Binarización, Posterización, Umbralización, Clipping, Slicing, Corrección gamma y Contraste, se propone un pequeño juego para averiguar si las definiciones han sido comprendidas.

#### **4. Filtrados:**

Aunque esta parte podría ser más larga, se ha preferido dar simplemente unas nociones de los filtrados más sencillos, para que el alumno, después, experimente por sí mismo los efectos de diferentes filtros en el programa TIRMA.

En esta demostración se muestran un filtro paso-alto de detección de contornos, uno paso-bajo y un filtro de mediana que elimina el ruido sal y pimienta simulado en una imagen.

## 6. Software adicional incluido.

Para completar la visión de las posibilidades que abre la imagen digital, se ha decidido incluir una serie de programas, con fines demostrativos que tratan o generan imágenes digitales. No constituyen, ni muchísimo menos, toda la oferta de software en este sentido, pero sirven como representación de aplicaciones distintas a lo que se ha visto hasta ahora el trabajo.

Hay que recalcar que este software se ha conseguido a través de una distribución *Shareware*, cuya finalidad es darle gran difusión a los programas, por ello, se pueden mostrar gratuitamente, ya que el uso que se les va a dar no será profesional. Algunos de estos programas permiten el uso durante un tiempo limitado (30 días), pero considerando que el presente trabajo será utilizado en alguna práctica docente de menos de 2 horas por cada grupo de prácticas, no se considera violada esta recomendación. Se espera que al incluir estos programas en el CD-ROM alcancen mayor difusión entre los alumnos de Imagen y Sonido de la escuela.

Los programas incluidos son los siguientes:

VPIC: Es un software de visualización y conversión de ficheros de imágenes que trabaja bajo el sistema operativo DOS (a partir de la versión 2.0). Se incluye junto al subdirectorío de las imágenes, para facilitar su visualización directamente desde DOS, sin necesidad de entrar en Windows.

Este programa ofrece unas ventajas que lo hacen propicio para visualizar imágenes. En primer lugar la rapidez con la que muestra las imágenes junto a la calidad que ofrece; alcanza hasta resoluciones de 1024x768 pixels con 256 colores, en el equipo del laboratorio, lo que permite visualizar completamente todas las imágenes incluidas en el trabajo (se adjunta una foto de San Francisco de esta resolución). Puede visualizar consecutivamente un conjunto de imágenes que se seleccionan con la barra de espacio, sin volver a la pantalla del programa (en la parte inferior de la pantalla se muestran los comandos y su correspondientes teclas). Cuando está visualizando una imagen, puede modificar las componentes RGB independientemente, el contraste y el brillo de la imagen con las teclas F1 a F11. Para consultar un menú de ayuda se puede pulsar la tecla '/' mientras se ve la imagen.

Además convierte la imagen a bastantes formatos diferentes, aunque esta opción se recomienda hacerla desde el siguiente programa que se comentará. Por último, el programa dispone de un set de drivers muy amplio para ser instalado con otras tarjetas de vídeo diferentes, por si el trabajo se instala en un equipo diferente y no ocupa mucha memoria.

Si se desea más información sobre el programa se pueden listar los ficheros VPIC.DOC y VPIC.TXT aunque se encuentran en inglés.

El programa se arranca tecleando VPIC desde su directorio ,simplemente.



## Graphic WorkShop v1.1:

Es una herramienta muy útil para visualizar y convertir los diferentes formatos de ficheros de imágenes. Funciona bajo Windows y aunque no es tan rápido como el programa anterior, ofrece un entorno más agradable, mostrando una primera aproximación en pequeño de todas las imágenes incluidas.

En la visualización se emplean los 256 colores que el Windows tiene configurados y muestra un máximo de 800x600 pixels. En el caso de que la imagen sea mayor de esta resolución aparecen unos cursores deslizantes para desplazar la parte visible de la imagen.

Por contra, dispone de gran versatilidad en la conversión de ficheros gráficos. Convierte entre los siguientes formatos: ART, BMP, CUT, DIB, EXE, GIF, HRZ, IFF, IMG, JPG, LBM, MAC, MSP, PIC, PCX, RAS, RLE, TGA, TIF, TXT y WPG. Estos formatos constituyen la casi totalidad de ficheros gráficos empleados bajo PC.

El manejo del programa es muy sencillo ya que opera dentro del entorno Windows, y todas las operaciones pueden realizarse con ayuda del ratón. Para arrancar el programa basta con pulsar dos veces seguidas sobre su icono.





## GrabIt Pro:

Este programa es un capturador de pantallas bajo Windows. Permite capturar un área determinada, una ventana, o la pantalla completa. El formato con el que puede grabar las imágenes capturadas es BMP y RLE.

GrabItPro se empleó durante la realización del trabajo para pasar imágenes de una aplicación a otra, siempre bajo el entorno de Windows. Se ha considerado importante el incluirlo en el CD-ROM para ofrecer esta misma ventaja a los futuros usuarios de todas las nuevas aplicaciones que se incluyen.

## Winmorph:

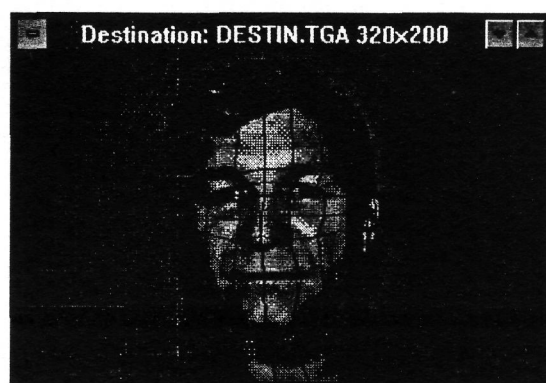


Es el primero de los dos programas de morfismos que se incluyen. Es una de las más modernas aplicaciones de la imagen digital y consiste en calcular la transición de una imagen a otra. En la última demostración se muestra algún morfismo realizado con este mismo programa.

La técnica de morfismo se ha aplicado en videoproducción, tanto cinematográfica como para vídeo-clips musicales y anuncios publicitarios. Las más conocidas son las transformaciones que aparecen en el vídeo-clip de 'Black or White' de Michael Jackson, donde aparecen diferentes caras de chicos y chicas que se van transformando de unas a otras. También están teniendo gran incidencia en

anuncios de TV.

Este programa funciona bajo Windows y es bastante sencillo de manejar. Emplea la técnica de mallas para crear la transición entre la imagen de origen y la de destino. Cada pequeña unidad de la malla tiene su correspondencia en las 2 imágenes y se crea una interpolación entre ellas para llegar de una a otra en el número de cambios fijado. Además del cambio de forma de estos pequeños polígonos, también se realiza un cambio de lugar, por lo que el proceso matemático de interpolación de forma y camino puede llegar a complicarse, hasta el punto que el cálculo de un morfismo lleve horas de proceso.



En este ejemplo que se incluye junto al programa, se ve la malla de 16x12 que se aplica a cada una de las 2 imágenes, y se puede ver que hay una correspondencia entre los elementos de la malla. Hay ciertas líneas guía que hay que incluir para obtener un buen resultado, como son la línea de los ojos, la de la nariz, el contorno de la cara, etc... El morfismo podrá contener tantas imágenes (frames) como se indique y cuántos más tenga, mayor calidad tendrá el cambio, aunque más tardará el proceso.



El resultado son un conjunto de imágenes en formato TGA con numeración correlativa, que pueden conformar una animación con la utilidad TGAFLI.EXE que también se incluye en el mismo subdirectorío, y que hay que ejecutar desde DOS.

### *PhotoMorph:*



Otro programa de morfismos incluido es el Photomorf. La diferencia con el anterior es que no emplea el método de mallas, sino el de puntos para trazar las guías del morfismo. Es más sencillo de usar y más rápido en el proceso de cálculo, pero no controla el desplazamiento de cada parte de la imagen como lo hace el winmorph.

Para situar los puntos llave, hay que pinchar con el ratón en la imagen de origen y aparece un punto gemelo en la de destino, que luego hay que desplazar para indicar donde irá a parar. Hay que situar puntos llave en sitios estratégicos, como son en el caso de una cara, los ojos, el contorno de los labios, las cejas, etc... Hay que emplear bastantes puntos, pero tampoco exceder de 100 ya que el tiempo de cálculo se hace mayor y el resultado puede quedar algo confuso.

Este programa, además permite transiciones entre las 2 imágenes de otras forma (cortinillas, fundidos,etc...). Las imágenes con las que trabaja son en formato JPEG, por lo que puede que haya que usar el conversor GWS. El resultado del morfismo o transición realizada la almacena en formato AVI sin necesidad de ejecutar ningún otro programa adicional.

## Popout:



Este programa de reciente aparición desarrolla otra aplicación de las técnicas informáticas aplicadas a la imagen, que se ha puesto muy de moda actualmente. Se trata de los llamados Estéreogramas. Son imágenes aparentemente ruidosas, sin ninguna información, pero que tras un desenfoque con la vista, dan paso a una imagen aparentemente tridimensional de gran realismo.

El programa funciona con una imagen BMP de origen que tenga 16 niveles de gris y cada uno de esos niveles de gris se representará a una altura diferente en la vista tridimensional. El efecto se consigue separando los puntos que conforman una figura a una cierta distancia, de forma que cuando la visión se desenfoca sobre la imagen hasta el punto de hacerlos coincidir en el sitio exacto, aparece la figura completa y separada del fondo.

Las pruebas realizadas con imágenes complejas de 16 bits (fotos, etc...) no dieron muy buen resultado, ya que requieren mayor habilidad para "descubrir" la figura. El ejemplo que se incluye en la demostración y en la presente memoria es muy sencillo de ver gracias a que solo hay 2 niveles de profundidad muy separados (la imagen original contenía sólo blanco y negro).

La versión que se distribuye en shareware no contempla muchas opciones que se podrían llevar a cabo con la versión registrada, como son el emplear un patrón diferente del ruido, y además hacerlo en color.

Si se desea confeccionar una imagen tridimensional, se aconseja tomar una imagen relativamente sencilla, convertirla a 16 colores (posterización a 16 niveles) con cualquiera de los programas que se incluyen o de que dispone el equipo del laboratorio, pasarla como parámetro de entrada en el programa popout y elegir uno cualquiera de los métodos para hacer el cálculo, que se incluyen en el subdirectorio y dar un nombre al fichero de salida.

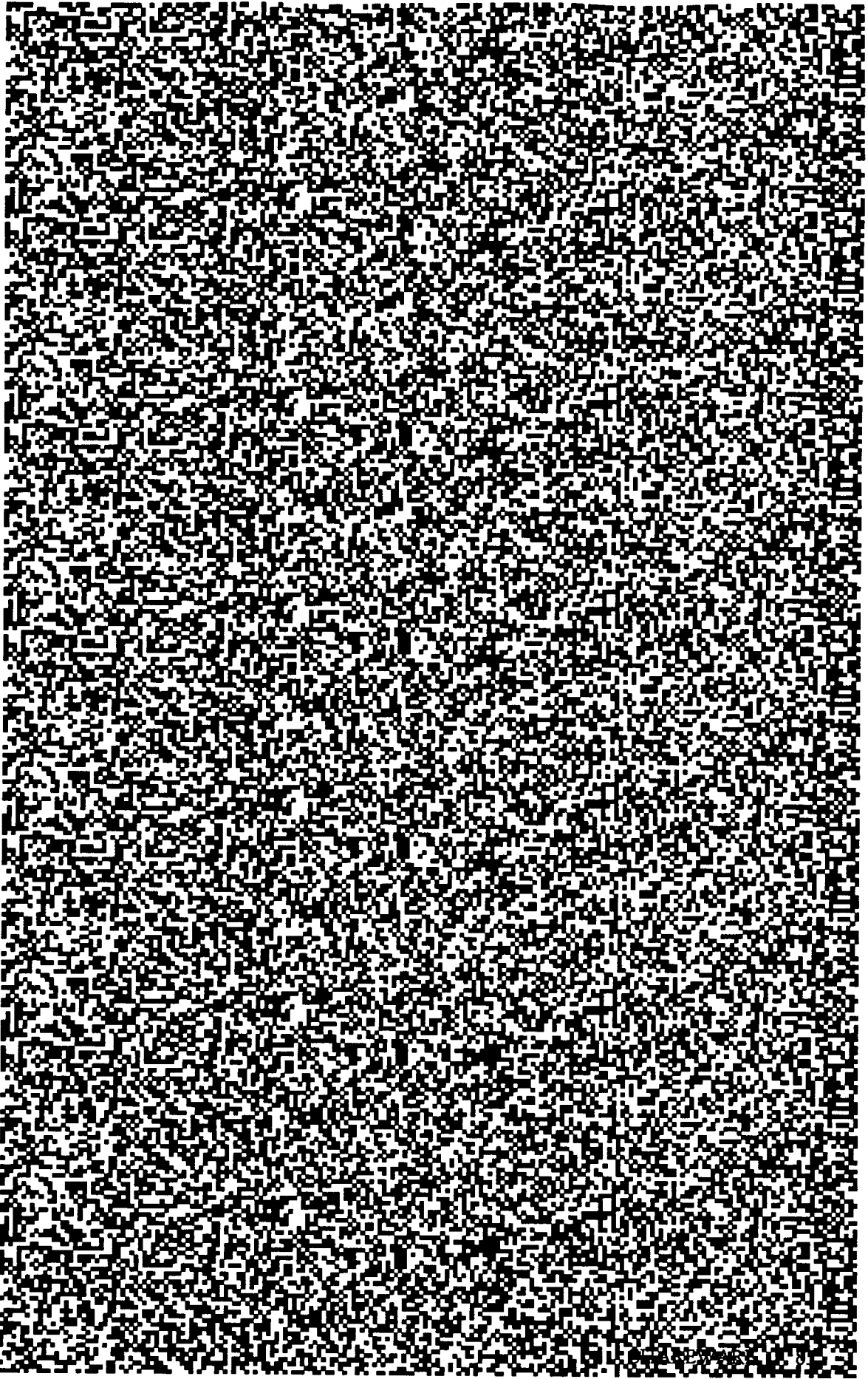
Una de las técnicas para "descubrir" las imágenes tridimensionales es acercarse mucho a la imagen (hasta tocarla con la nariz) y con la vista desenfocada, ir alejando paulatinamente la imagen manteniendo el desenfoque hasta una distancia de unos 20 o 30 cm donde irá surgiendo un baile de puntos precediendo al descubrimiento de la imagen tridimensional.

Hay publicada una serie de libros con gran cantidad de estas imágenes con el título de 'el Ojo Mágico', donde además en alguno de ellos se detallan otras técnicas para ver las imágenes y el fundamento en que se basan para construirlas.

La imagen de ejemplo que se incluye es un estéreograma del siguiente dibujo realizado con PhotoStyler:



Pity





## Truedemo:

Una de las aplicaciones de la imagen digital que se ha abierto un amplio campo de actuación es la de los entornos virtuales por medio de imágenes tridimensionales. De momento, el último eslabón al que nos llevan estas técnicas, es el de la realidad virtual que se está perfilando como una gran ayuda en numerosas aplicaciones.

Sin entrar en demasiado detalle, este programa es un buen ejemplo de creación de escenarios virtuales tridimensionales. Se ha elegido este programa ya que presenta directamente los objetos en una vista tridimensional y resulta más didáctico. Otros programas como el 3D Studio trabaja con multivista y puede ser algo más complejo de entender.

Se recomienda arrancar la demo de este programa picando en Self-running demo 800x600 observando las posibilidades que tiene y luego ejecutar el TrueSpace Demo y hacer pruebas sobre lo que se acaba de ver.

Se incluye una imagen de este programa donde se aprecia el escenario tridimensional donde se generan los objetos:



## 7. Librería de imágenes.

Junto a los programas que se incluyen en el CD-ROM también se adjuntan una serie de imágenes recogidas de muy diversas fuentes, aunque en su mayoría de otros CD-ROM's. La naturaleza de las imágenes procura ser de lo más variada, desde paisajes naturales, retratos, dibujos, comics, hasta escenas de alguna película, pasando por imágenes astronómicas, actores de cine o algunas generadas por ordenador.

Un total de más de 10 Megas de imágenes para ser empleadas como base en el manejo de los diversos programas de tratamiento digital. Se han preferido 3 formatos que son los más populares: pcx, bmp y gifs.

Las imágenes incluidas son las siguientes:

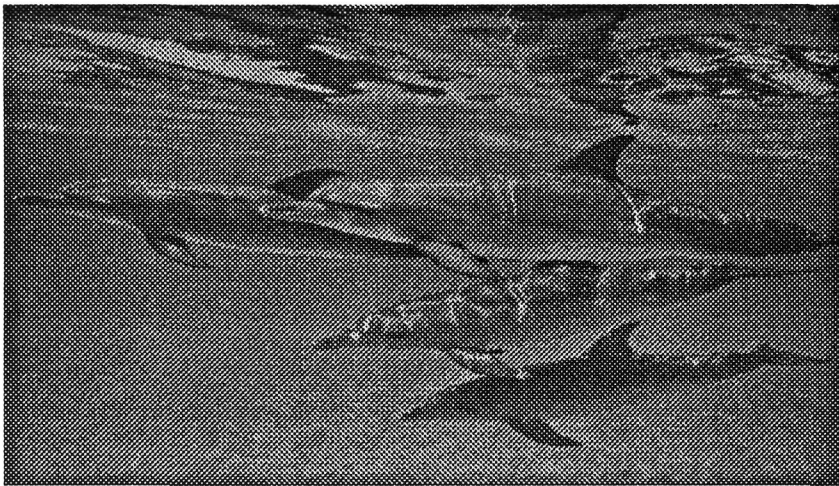
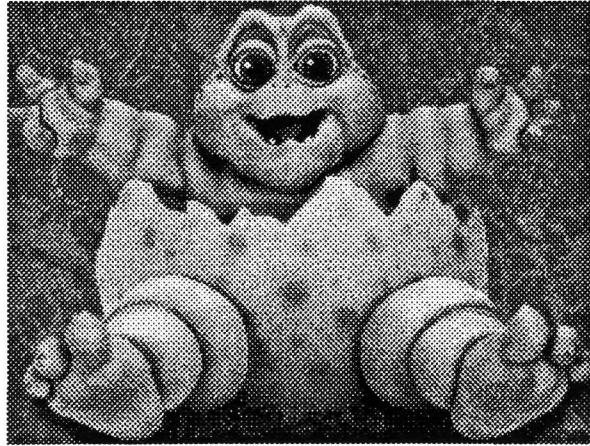
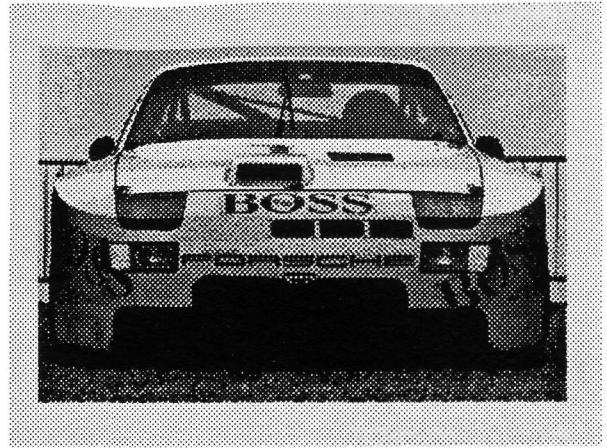
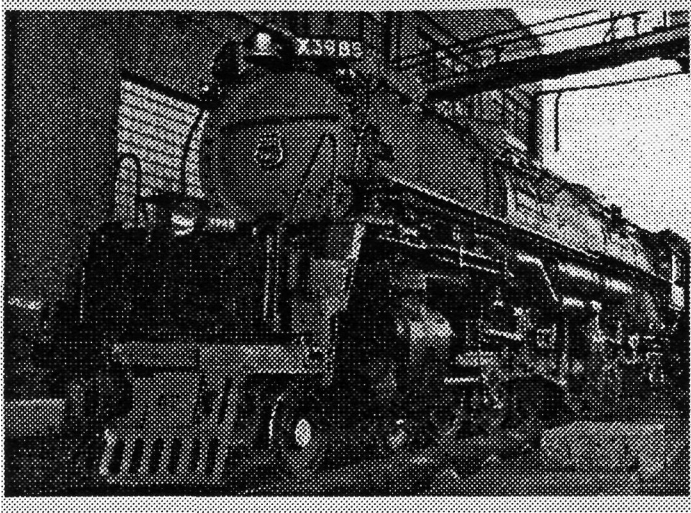
ESCUDO.BMP	GUMP1.BMP	PITY.BMP	PITY-3D.BMP
12G1.BMP	12G2.BMP	TRACY&HE.BMP	INGRID.BMP
CHAPLIN.BMP	GUMP.BMP	4G3.BMP	6G2.BMP
COMIC2.BMP	COMIC1.BMP	SAVWLD10.BMP	SAVWLD01.BMP
SAVWLD02.BMP	SAVWLD05.BMP	SAVWLD07.BMP	SAVWLD08.BMP
SAVWLD09.BMP	METELOGO.GIF	METEGLOB.GIF	METEOSAT.GIF
METEOMAP.GIF	NGC5128.GIF	JULIA.GIF	JUPITER.GIF
MARS.GIF	NEPTUNE.GIF	MERCURY.GIF	SATURN.GIF
VENUS.GIF	MAK12.GIF	M20.GIF	M27.GIF
M31.GIF	M_F.GIF	OCEANO.GIF	RUECA2.GIF

CINDY030.GIF	CINDY018.GIF	DORADO.GIF	FASH-322.GIF
NGC2024.GIF	NGC2264.GIF	NGC6979.GIF	ARISTOCA.GIF
AXIS.GIF	FOREST.GIF	GARDNLAK.GIF	JULIART.GIF
MACR02.GIF	MARILYN.GIF	METEOR.GIF	OLDMILL2.GIF
SANFRAN5.GIF	SHUTTLE8.GIF	SPACE_01.GIF	FLOR4.PCX
944GTP.PCX	BABYDINO.PCX	3985B.PCX	BRAZIL.PCX
SKYLINE5.PCX	FALL.PCX	CLRDOPLH.PCX	BEARS.PCX
ANEMONE.PCX	PARROT.PCX	COMPCHIP.PCX	CORN.PCX
HORSE.GIF	CROMA.GIF	TRUE1.BMP	SA01.PCX
SA04.PCX	SA10.PCX		

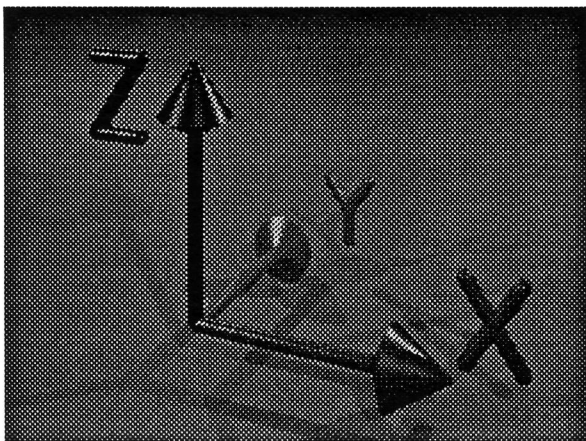
Estas imágenes han sido elegidas por algún motivo particular: calidad en la fotografía, diferentes resoluciones y temas variados, simpatía, buen contraste, ...

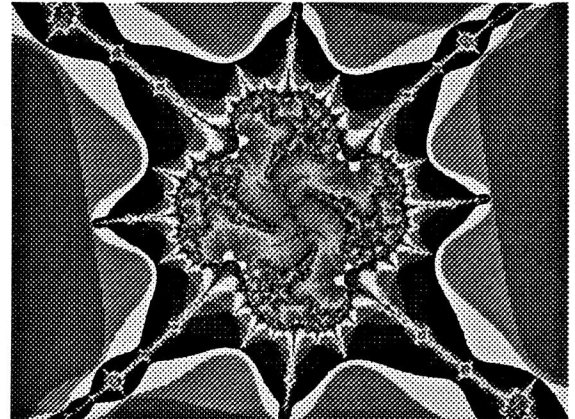
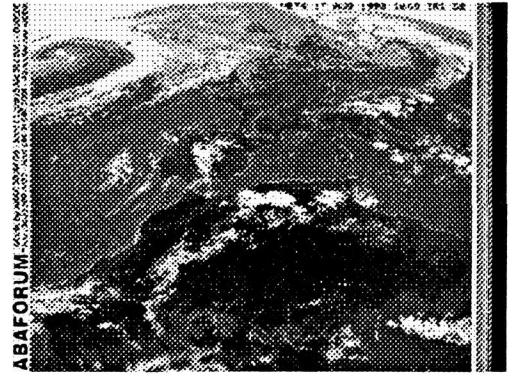
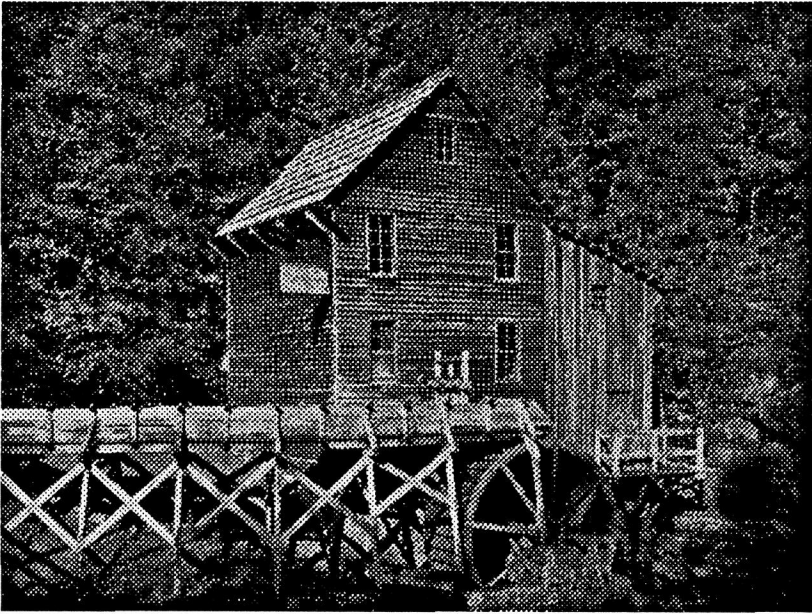
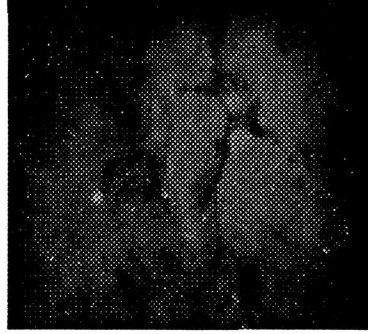
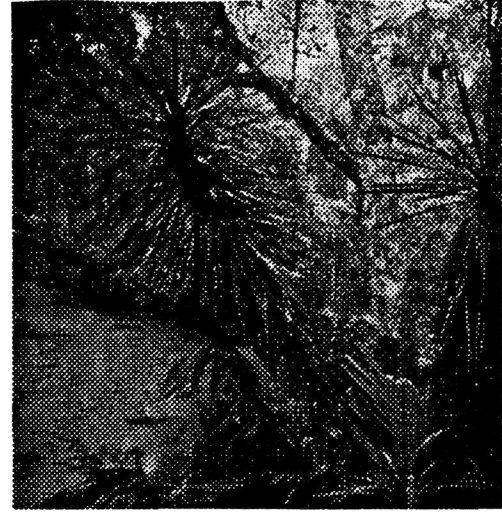
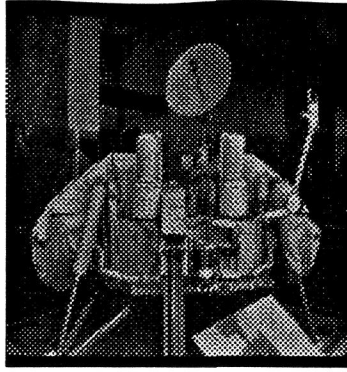
La intención es que se disponga de una librería de imágenes variada, apropiada y entretenida, para practicar las técnicas de procesamiento de imágenes digitales. Para mostrar el estilo de imágenes y las características que se comentan se ofrecen algunas de ellas en un fotomontaje.

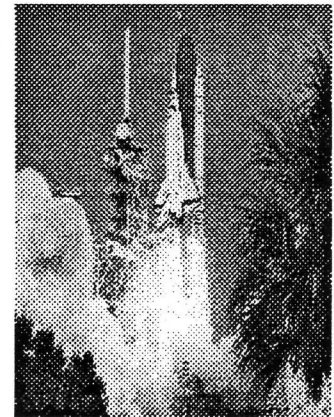
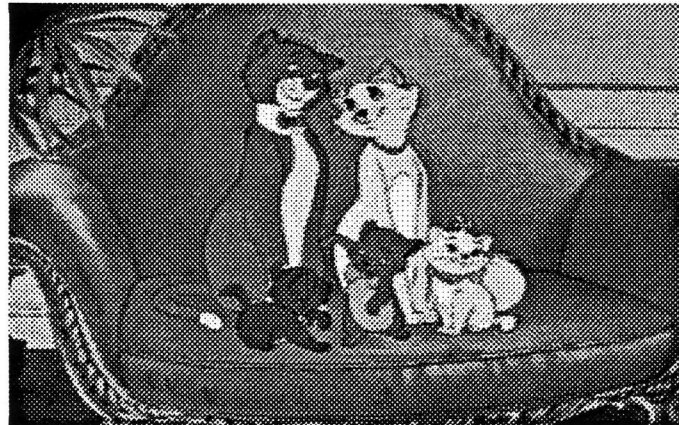
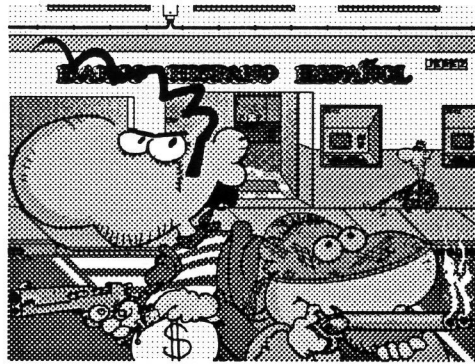
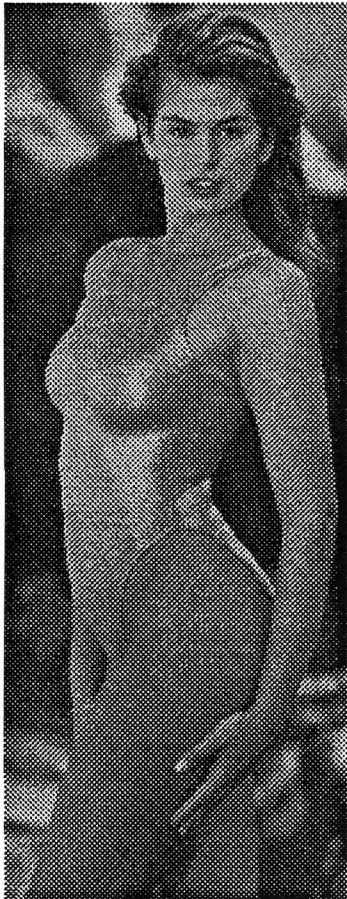
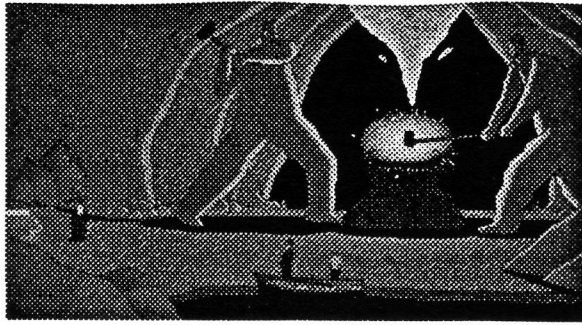




From 9001 "Full-Page Floral Wallpaper Pattern" Bookline: Dec 4/91 Austin, Texas 78769







## 8. Aplicación a una imagen digital concreta: Espectrogramas.

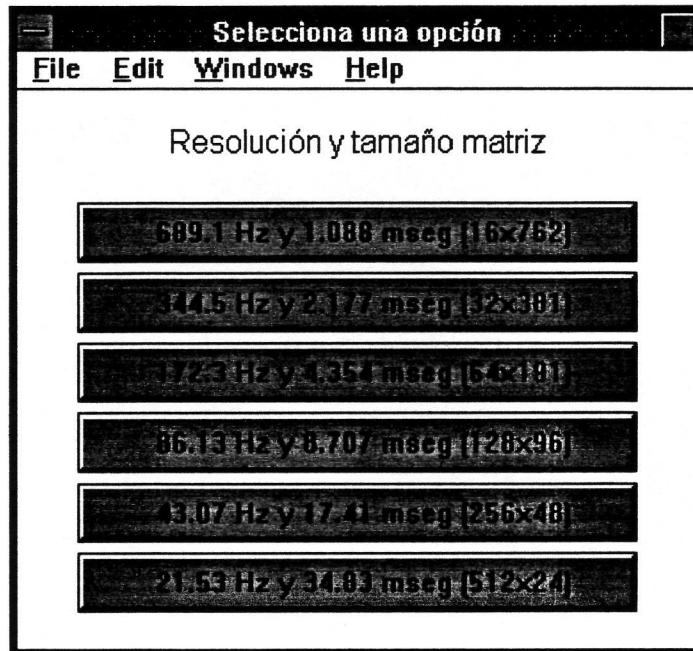
En el tema del audio digital, las posibilidades de tratamiento son muy extensas. En este trabajo se ha incluido un programa que calcula por medios digitales una representación tiempo-frecuencia, llamada Espectrograma. Esta representación de la señal de audio es muy empleada en estudio de señales de voz principalmente y hasta hace pocos años se representaba por medio de un equipo analógico llamado espectrógrafo.

El método para calcular el espectrograma pasa por digitalizar la señal de audio en primer lugar, y con un método de inventanado apropiado, calcular el espectro de pequeños segmentos (1 a 50 mseg). El espectro de estos pequeños segmentos, se muestra como columnas de una matriz que finalmente constituirá el espectrograma.

Vista una pequeña introducción, veamos cómo se maneja el programa:

Para ejecutarlo hay que teclear 'spectr' en la ventana de comandos de Matlab. Inmediatamente aparece una ventana de diálogo solicitando en nombre del fichero wav que contiene el sonido. Los ficheros \*.wav contienen sonido digitalizado normalmente a frecuencias preestablecidas de 11025, 22050 ó 44100 Hz. Hay gran número de programas que permiten grabar sonidos en ficheros wav, como el incluido en el equipo del laboratorio, que se adjuntaba a la tarjeta de audio, llamado WaveStudio.

Una vez indicado el fichero, que no debe ser demasiado extenso, para no esperar al procesado mucho tiempo, hay que establecer la resolución con que se calculará el espectrograma, esto es el tamaño de la ventana para calcular el espectro.



En la elección de la resolución aparecen siempre 6 opciones que trabajan con 32, 64, 128, 256, 512 o 1024 datos respectivamente. Lo que cambia es la resolución en frecuencia y tiempo que se consigue, ya que éste depende de la frecuencia a la que se muestrease el sonido. En la ventana de diálogo aparecerá siempre: espectrograma de banda ancha en la parte superior de las opciones, que destacan los detalles en tiempo, pero sacrifican los detalles en frecuencia, o lo que es más aconsejable, espectrogramas de banda estrecha en las opciones inferiores que destacan los detalles en frecuencia y pierden resolución en tiempo. La denominación de banda estrecha o banda ancha viene del espectrógrafo analógico y del ancho de banda que empleaba en los filtros paso-banda.

Tras elegir la resolución, comienza el cálculo del espectrograma que consiste en lo siguiente:

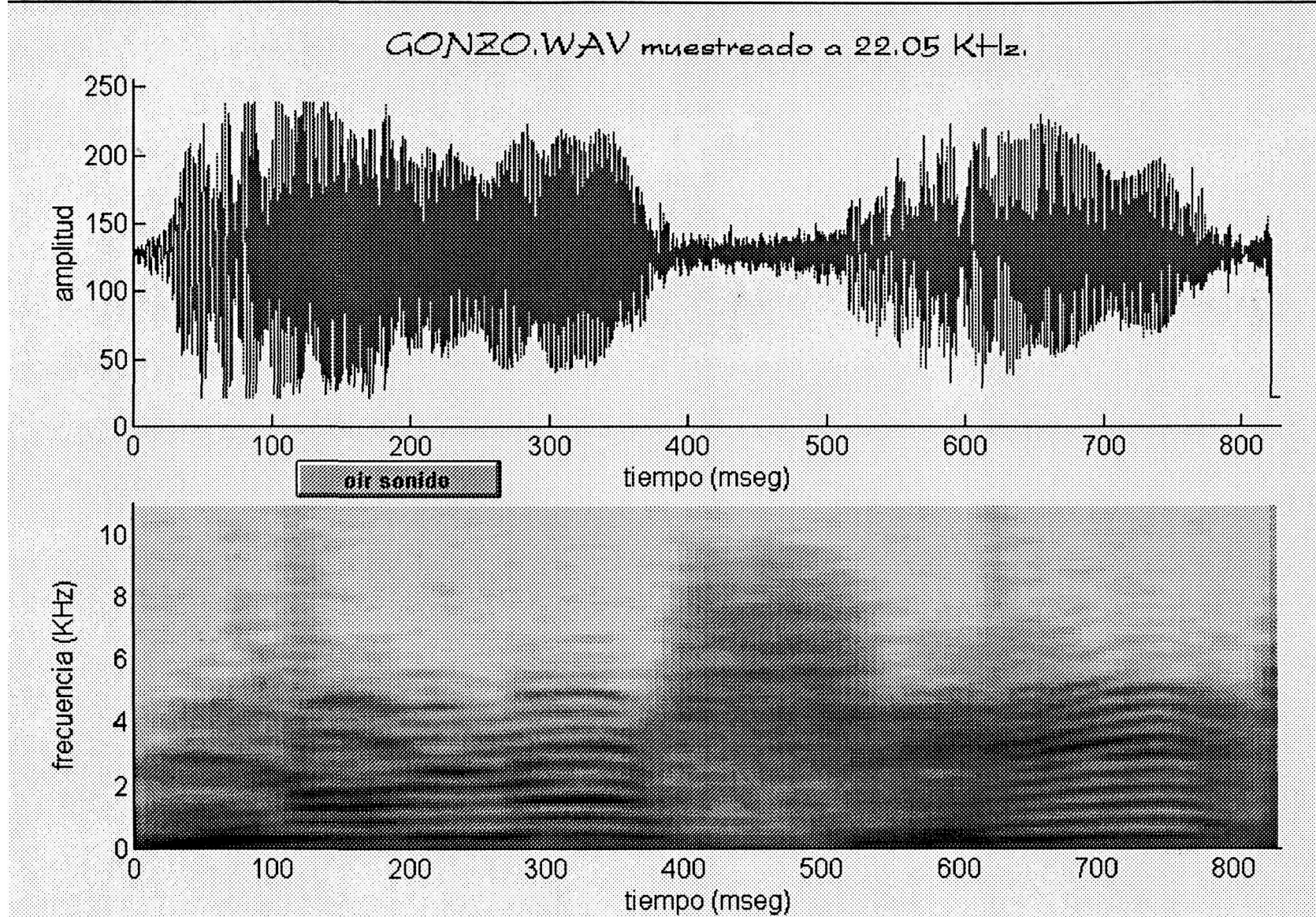
Para calcular el espectro del número de muestras que se haya seleccionado en la resolución, se separa el grupo en dos mitades, a las que se aplica una ventana de Hanning y se calcula su FFT para quedarnos con el módulo elevado al cuadrado. Para lograr mayor calidad, las ventanas se superponen 1/4 de su tamaño (método overlap-add). Finalmente nos quedamos con la mitad de las muestras por ser éstas simétricas.

Se hace una corrección de unidades, se normaliza y se pasa a escala logarítmica con un margen dinámico de 100 dBs. Durante el proceso de cálculo va dibujando la señal en el tiempo, para indicar el total calculado hasta el momento. Por defecto se dibuja en tres trozos para no perder mucho tiempo en representaciones gráficas. Cuando concluye, filtra la matriz resultante con un filtro paso bajo de 3x3 cuyos coeficientes valen todos 1/9, para suavizar los picos. El resultado final queda como se muestra en la figura:

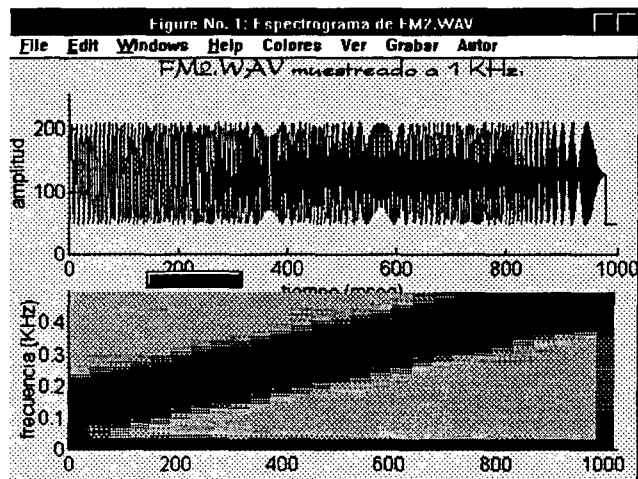
Hay 2 ventanas con la misma escala de tiempos en milisegundos, donde se compara la señal habitual en amplitud-tiempo y el espectrograma donde se ve una representación con 256 niveles de grises en los ejes tiempo-frecuencia, y donde la intensidad de cada punto representa la potencia a esa frecuencia y en ese instante. Es curioso observar como el sonido que comienza en 400 mseg. y dura unos 120 mseg., a pesar de tener muy poca amplitud tiene mucha potencia a alta frecuencia; la explicación es que se trata del fonema /s/. También se pueden intuir los formantes de las vocales que aparecen en la frase.

Figure No. 1: Espectrograma de GONZO.WAV

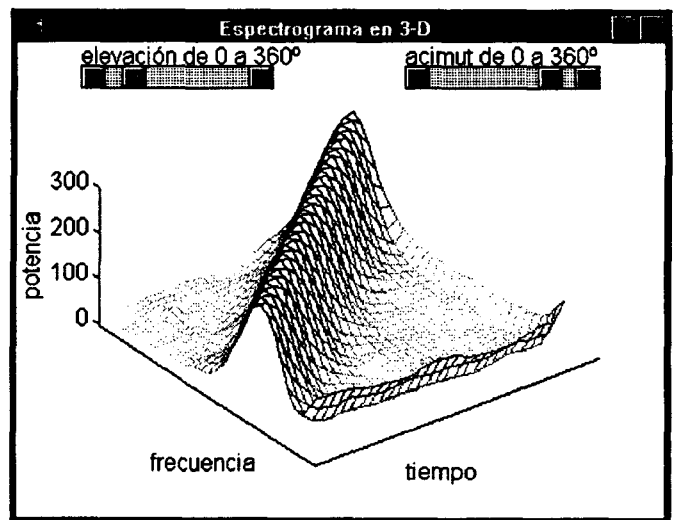
File Edit Windows Help Colores Ver Grabar Autor



Además hay un menú desplegable en la parte superior, conteniendo las opciones habituales de las figuras en Matlab y que incluye un menú de Colores con todas las paletas predefinidas y opciones como aumentar y reducir el brillo, un menú Ver en el que se incluyen visualizaciones 3D y de corte transversal, una opción para Grabar la imagen en cualquiera de los formatos bmp, pcx, gif y tiff y por último una opción donde figura el autor en la típica opción acerca de...

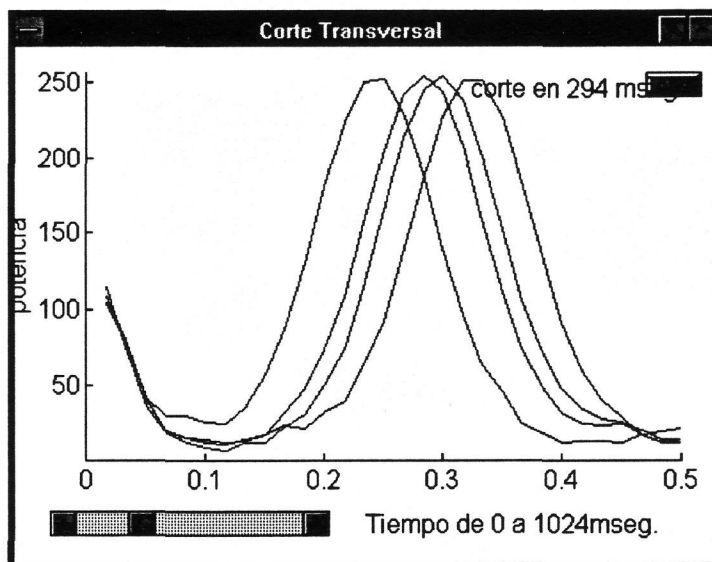


La opción Ver consta de dos tipos de visualizaciones con el fin de obtener la máxima información posible del espectrograma. En la vista tridimensional se puede elegir entre un sombreado y una malla transparente. En la figura que aparece, existen 2 cursores deslizantes para situar el punto de vista en coordenadas de elevación y acimut en grados.





También se puede visualizar un corte transversal del espectrograma, que indica la potencia espectral respecto a la frecuencia. Para indicar el tiempo del corte hay una barra deslizante y un indicador del instante donde se efectúa el corte.



Con estas herramientas se facilita el conocimiento exhaustivo del resultado calculado por el programa. Además se puede escuchar el sonido digitalizado pulsando el botón 'Oír sonido'.

Por último se incorpora el menú acerca de... con una breve animación del autor que se llevó a cabo digitalizando 2 imágenes parecidas y almacenándolas en la memoria del matlab con el comando `getframe`, tras reducir la paleta de colores a una sola con 256 colores, y animando el resultado con el comando `movie`. Se muestra el copyright y el título del programa.



```

function spectr
%SPECTR calcula el espectrograma de un sonido que se encuentre
grabado en disco
%      como fichero .WAV
%      Establece diferentes resoluciones que permiten
destacar los detalles en
%      frecuencia (banda estrecha) perdiendo detalle en tiempo
o viceversa.
%      Permite grabar la imagen del espectrograma resu
ltante a cualquier formato
%      para su posterior análisis con un software de tratamien
to de imágenes aunque se
%      puede ver directamente con varias paletas de colores.
%      Puede mostrar una vista tridimensional del espe
ctrograma desde cualquier
%      punto de vista o un corte transversal para estudiar el
espectro de un instante
%      determinado.
%      No precisa parámetros de entrada.
%
%

global resul dur fs;

[fich,camino] = uigetfile('*.*wav','Cargar un sonido');

if ~(isstr(fich)) disp('No se ha cargado ningún sonido')
else

global y fs

[y,fs] = loadwave([camino,fich]);

tam =length(y);

%=====
%                      Figura Principal
%=====

v1 = figure('Units','normalized','Position',[0 0 1 1],'Name',['
Espectrograma de ',fich],...
           'Color',[1 1 .8]);
%=====

axes('Units','normalized','Position',[.0938 .5417 .875 .375],...
     'XLimMode','manual','XLim',[0 floor(1000*tam/fs)],...
     'YLimMode','manual','YLim',[0 255]);

xlabel('tiempo (mseg)');

ylabel('amplitud');

title([fich,' muestreado a ',num2str(fs/1000),' KHz.']);

```

```

set(get(gca,'title'),'Color',[0 0 1],'FontName','Hansa','Font
Size',16,'FontWeight','light');

o1 = [num2str(fs/32),' Hz y ',num2str(24*1000/fs),' mseg (16x',
num2str(ceil(tam/24)),')'];
o2 = [num2str(fs/64),' Hz y ',num2str(48*1000/fs),' mseg (32x',
num2str(ceil(tam/48)),')'];
o3 = [num2str(fs/128),' Hz y ',num2str(96*1000/fs),' mseg (64x',
,num2str(ceil(tam/96)),')'];
o4 = [num2str(fs/256),' Hz y ',num2str(192*1000/fs),' mseg (128
x',num2str(ceil(tam/192)),')'];
o5 = [num2str(fs/512),' Hz y ',num2str(384*1000/fs),' mseg (256
x',num2str(ceil(tam/384)),')'];
o6 = [num2str(fs/1024),' Hz y ',num2str(768*1000/fs),' mseg (51
2x',num2str(ceil(tam/768)),')'];

n = menu('Resolución y tamaño matriz',o1,o2,o3,o4,o5,o6);

figure(v1);

n = 2^(n+4); % tamaño de la ventana
tam = ceil(tam/n)*n; % tamaño de y múltiplo
de n
y(tam) = 0;

eje = axes('Units','normalized','Position',[.0938 .5417 .875 .3
75],...
'XLimMode','manual','XLim',[0 tam],...
'Visible','off','NextPlot','add');

y = y-round(mean(y)); % elimina continúa

md = 100; % margen dinámico en dB
's
c = 10^(-md/20); % margen dinámico pasad
o a escala lineal

fin = (4*tam/n)-3;
lista = linspace(1,fin,3); % dibuja y(t) en 3 troz
os (se puede modificar el 3)

%=====
%          Proceso de cálculo del Espectrograma
%=====

for i = 1:fin
    emp = (i-1)*(n/4)+1; % hay un solapamiento d
e 1/4 de ventana

% espectro de la ventana:
    varf = spectrum(y(emp:emp+n-1),n/2,n/4);

    l = length(varf);
    varf = 2*sqrt(varf/l); % corrige unidades
    resul(1:l,i) = (varf(1:-1:1))'; % asigna a resul las co

```

```

lumnas calculadas
    if any(lista==i) plot(y(1:emp+n-1));drawnow
    end;

end;
%=====

resul = resul/max(resul(:));           % normaliza a 1

resul = 20*log10(resul+c);           % pasa a escala logarít
mica

h = 3;
hpb = ones(h);                       % filtro paso bajo de s
uperficie
resul = conv2(resul,hpb,'valid');    % suaviza los saltos br
uscos en la matriz resul

[a,b] = size(resul);

resul = 256+(((resul-max(resul(:)))*255/(max(resul(:))-min(resu
l(:)))));
resul = 257-floor(resul);           % pasa a escala lineal
de 1 a 256

dur = floor(1000*tam/fs);

axes('Units','normalized','Position',[.0938 .0833 .875 .375],...
.
    'XLimMode','manual','XLim',[0 dur],...
    'YLimMode','manual','YLim',[0 fs/2000]);

axes('Units','normalized','Position',[.0938 .0833 .875 .375],'V
isible','off');

image(resul);                       % dibuja el espectrogra
ma

axis('off')

colormap(gray(256));

xlabel('tiempo (mseg)');
ylabel('frecuencia (KHz)');

%=====
%                               Menús auxiliares
%=====

colormen;                            % incluye menu Colores

ver3d = uimenu(v1,'label','Ver');    % incluye menu Vista en
3-D y Corte Transversal

```

```

v3 =    uimenu(ver3d,'label','3-D');
        uimenu(v3,'label','mallado','callback','dibres1');
        uimenu(v3,'label','sombreado','callback','dibres2');
        uimenu(ver3d,'label','Corte','callback','corte');

graba = uimenu(v1,'label','Grabar');    % incluye menu Grabar
        uimenu(graba,'label','Imagen espectrograma','callback',
'global resul;imsalvar(resul,gray(256))');

ayuda = uimenu(v1,'label','Autor');    % incluye menu Acerca d
e...

        uimenu(ayuda,'label','Acerca de...','callback','acerca'
);

%=====
%          Botón para oír el sonido estudiado
%=====

s =['global y fs;playwave(y,fs)'];
uicontrol('Units','normalized','Style','Pushbutton','Position',
[.2188 .4625 .1563 .0417],'Callback',s,'String','oír sonido');

%=====

end;

end

```

```

%ACERCA abre una ventana y muestra el copyrigh del
% programa que calcula los espectrogramas
% bajo Matlab y una pequeña animación del
% autor del software.
%

load acerca.mat

figure('Units','pixels','Position',[250 250 400 200],...
'Menubar','none','NumberTitle','off','Name','Acerca de ...','Co
lor',[.49 1 .83]);

axes('Units','normalized','Position',[.5 .8 .3 .3],'Visible','o
ff');
te1 = text(.05,0,'Espectrogramas para');
te2 = text(.2,-.4,' Matlab® v.4. ');
te3 = text(.1,-.75,'Javi Selvi Nácher ©1994');
set(te1,'FontName','Goudii','FontSize',15,'Color',[0 0 1]);
set(te2,'FontName','Goudii','FontSize',15,'Color',[1 0 0]);
set(te3,'FontName','Goudii','FontSize',12,'Color',[0 0 .8]);

axes('Units','normalized','Position',[.55 .07 .4 .4],'Visible',
'off');
image(resul)
axis('off')

axes('Units','normalized','Position',[.05 .05 .8 .8],'Visible',
'off');
colormap(map);
axis('off')
movie(M,14,5);
pause
close(gcf)
end

```

```

%CORTE visualiza un corte transversal de la función tridimensio
nal
%      resul obtenida del calculo del espectrograma.
%

figure('Units','normalized','Position',[.27 .29 .72 .66],...
       'Name','Corte Transversal','NumberTitle','off','Menuba
r','none');

global resul dur fs

axes('Position',[.1 .2 .85 .75],'XLim',[0 fs/2000],'YLim',[1 25
6]);

ylabel('potencia');

[a,b] = size(resul);

va = 'Value';
lugo1 = 'corte en ';
lugo2 = ' mseg.';
eje1 = axes('Position',[.1 .2 .85 .75],'YLim',[1 256],'Visible'
,'off','NextPlot','add');
eje2 = axes('Position',[.65 .92 .4 .1],'Visible','off','NextPlo
t','replace');

haz = ['t=get(boton,va);axes(eje1);'...
       'plot(256-flipud(resul(:,t)));axes(eje2);cla;',...
       'text(0,0,[lugo1,int2str(round(t*dur/b)),lugo2]);'];

boton = uicontrol('Style','slider','String','tiempo (mseg)',...
                 'Units','normalized','Position',[.05 .05 .4 .
05],...
                 'Min',1,'Max',b,'Value',b/2,'Callback',haz);
eval(haz);

axes('Position',[.5 .075 .4 .05],'Visible','off');

text(0,0,['Tiempo de 0 a ',int2str(dur), 'mseg.']);

uicontrol('Style','pushbutton','Units','normalized','Position',
[.9 .9 .08 .05],'String','Borrar','callback','axes(eje1);cla');

```

```

%DIBRES1 dibuja la función resul del espectrograma en 3-D
%      como una malla.

figure('Units','normalized','Position',[.27 .29 .72 .66],...
      'Name','Espectrograma en 3-D','NumberTitle','off','Men
ubar','none');

axes('Position',[.1 .97 .3 .02],'NextPlot','new','Visible','off
');

text(0,0,'elevación de 0 a 360º');

axes('Position',[.6 .97 .3 .02],'NextPlot','new','Visible','off
');

text(0,0,'acimut de 0 a 360º');

global resul

mesh(256-flipud(resul));

colormap(flipud(bone(256)))

set(gca,'XTick',[],'YTick',[]);

xlabel('tiempo')
ylabel('frecuencia')
zlabel('potencia')

va = 'Value';

vista = ['view(get(b2,va),get(b1,va))'];

b1 = uicontrol('Style','slider','Units','normalized','Position'
,[.1 .9 .3 .05],'Max',360,'Value',30,'callback',vista);

b2 = uicontrol('Style','slider','Units','normalized','Position'
,[.6 .9 .3 .05],'Max',360,'Value',322.5,'callback',vista);

```



```

%DIBRES2 dibuja la función resul del espectrograma en 3-D
%      como una superficie.

figure('Units','normalized','Position',[.27 .29 .72 .66],...
       'Name','Espectrograma en 3-D','NumberTitle','off','Menubar','none');

axes('Position',[.1 .97 .3 .02],'NextPlot','new','Visible','off');

text(0,0,'elevación de 0 a 360º');

axes('Position',[.6 .97 .3 .02],'NextPlot','new','Visible','off');

text(0,0,'acimut de 0 a 360º');

global resul

surfl(256-flipud(resul));

shading interp                % realiza un sombread de la mala por interpolación
la por interpolación

colormap(flipud(bone(256)))    % usa un mapa de colores con tonos azulados
nos azulados

set(gca,'XTick',[],'YTick',[]);

xlabel('tiempo')
ylabel('frecuencia')
zlabel('potencia')

va = 'Value';
vista = ['view(get(b2,va),get(b1,va))'];

b1 = uicontrol('Style','slider','Units','normalized','Position',[.1 .9 .3 .05],
'Max',360,'Value',30,'callback',vista);

b2 = uicontrol('Style','slider','Units','normalized','Position',[.6 .9 .3 .05],
'Max',360,'Value',322.5,'callback',vista);

```

## 9.El CD-ROM como medio de almacenamiento ideal para el proyecto.



La tecnología del láser ha obsequiado a los equipos informáticos con un medio de almacenamiento ideal para necesidades multimedia. El problema de la gran memoria requerida para el almacenamiento de imágenes de alta calidad y sonidos digitalizados conjuntamente queda solventado con la gran capacidad de los CD-ROM. Hasta 680 Megabytes de información que suponen 150.000 páginas impresas o unas 5000 imágenes es capaz de recordar un pequeño disco de 12 centímetros de diámetro y apenas 20 gramos de peso.



Han aparecido otros medios de almacenamiento, pero éste ha cuajado a nivel comercial y la oferta de títulos grabados en CD-ROM ha alcanzado un gran volumen y crece día a día. Otra posible opción es el WORM de escritura única y lectura múltiple, que se comentará a continuación. La diferencia es que el CD-ROM no puede ser grabado por el usuario y debe ser enviado a una empresa específica de grabación con el consiguiente encarecimiento; de todos modos, la necesidad de grabación es bastante esporádica y no ha determinado el crecimiento del WORM.



El laboratorio de TVII de la E.U.I.T.T. dispone de una unidad lectora de CD-ROM que le da acceso al gran volumen de información que actualmente se oferta en discos. Las características del presente Trabajo Fin de Carrera lo hacen idóneo para estar disponible en un disco CD-ROM.

Principalmente, es el gran volumen de información, más de 100 Megabytes, el que imposibilita su disponibilidad en diskettes, ya que se trataría de más de 70 unidades de alta densidad (1'44 Megabytes) y no se lograría la continuidad de las diversas demostraciones interactivas. Tampoco es aconsejable reservar el volumen del disco duro para este proyecto, ya que el ordenador quedaría limitado a su presentación únicamente. Resta por lo tanto la unidad de CD-ROM que se muestra al ordenador como una unidad más.

La configuración de los equipos de entrada/salida del ordenador del laboratorio es como sigue:

Unidad A: Discos de 5'25 pulgadas (hasta 1'44 Megabytes)

Unidad B: Discos de 3'5 pulgadas (hasta 1'44 Megabytes)

Unidad C: Disco duro de 250 Megabytes de capacidad.

Unidad D: Disco Compacto de Sólo Lectura CD-ROM.

Sistema de Backup: Cintas de hasta 250 Megabytes.



Cada pequeño disco de CD-ROM es comparable a un gran disco duro con la limitación de no poder escribir en él. También existen diferencias en la velocidad de transferencia que en el CD-ROM se ralentiza hasta unos 300 Kilobytes por segundo y en el tiempo de acceso que puede llegar a ser de unos 300 milisegundos, aunque estos detalles se verán más en profundidad en el apartado de características técnicas de los CD-ROM. De todos modos, aunque el CD-ROM constituye una fuente de datos inagotable, no ocupa el lugar del disco duro, donde se almacenan los programas de uso frecuente y los

ficheros creados por el usuario. Son equipos absolutamente complementarios para dotar al ordenador de la potencia que precisa en el área multimedia.



Durante la realización del proyecto fue necesaria la consulta de la máxima cantidad de CD-ROM para lograr 2 objetivos:

- Conocer la forma en que se presentaban las demostraciones interactivas, que en su mayoría eran publicitarias de productos informáticos, o de grandes empresas multimedia. Conjuntamente se comprobó que la estructura en el reparto de los programas, imágenes y sonidos era muy similar aún en discos de diferentes firmas.

- Extraer algunos programas que se distribuían en modo *shareware* y que resultarían necesarios para complementar el juego de herramientas que darían al alumno al que se destina el proyecto, la posibilidad de aplicar todo aquello que se muestra en las demostraciones. Principalmente son programas para visualizar imágenes y realizar transformaciones sobre ellas, tanto bajo DOS como bajo Windows, aunque también hay programas de morfismos, de imágenes tridimensionales o estereogramas y alguno para reproducir música digitalizada.

Hay que explicar que los programas llamados shareware no son un tipo determinado de software, sino una forma de distribución que permite a los usuarios disponer de una versión reducida del programa definitivo a modo de prueba para pasar a registrarse si así lo creen conveniente; de este modo, estos programas son de libre distribución con las condiciones especiales que algunos de ellos establecen.



De este modo, se han consultado más allá de 10 Gigabytes de información, en busca de imágenes apropiadas, sonidos especiales y sobretodo programas sencillos en el manejo pero de utilidad en el tratamiento digital de las imágenes. Este proyecto procura ser un buen ejemplo de la enorme utilidad que el CD-ROM está suponiendo en los sistemas multimedia.

## Especificaciones generales de los CD-ROM:



La característica más relevante de un CD-ROM es su gran capacidad de almacenamiento. Los datos se codifican en una sola pista que transcurre a lo largo del disco en forma de espiral, como en los discos de vinilo, pero con la salvedad de que la lectura se efectúa desde dentro hacia fuera.

Esta pista está dividida en sectores de idéntico tamaño, cosa que no sucede con los discos duros o con los diskettes. Posteriormente se explicará el porqué de esta diferencia. En cada sector se incluyen junto con los datos del programa, 304 bytes destinados a la corrección de posibles errores de lectura, cantidad algo superior a la empleada en los discos compactos de audio. Asimismo existen sectores con información de tipo técnico, como sincronismos, duraciones, comienzos y finales de canciones, etc...

Otro de los aspectos que caracteriza al CD-ROM es el hecho de que es un soporte de sólo lectura, como su propio nombre indica (ROM=Read Only Memory). Esto significa que su contenido no puede ser alterado una vez impreso. Tan sólo en el caso de los llamados discos WORM (Write Once Read Many) se puede llegar a grabar datos; sin embargo el precio de las

unidades que utilizan esta técnica lo hacen, de momento, inasequible para el usuario, y la calidad obtenida nunca será equivalente a la obtenida en la estampación en fábrica.

Uno de los componentes del reproductor que juega un papel muy importante a la hora de transferir los datos al disco es el *buffer*, que actúa como memoria intermedia entre el lector y el interface del ordenador. Dependiendo de su tamaño, se ejecutarán con mayor o menor eficiencia las transacciones de información.

El nombrado interface es necesario para conectar el reproductor al ordenador. Éste consiste en una tarjeta controladora que se encarga de gestionar las transferencias, así como de traducir las órdenes lógicas de búsqueda a órdenes físicas. Además, ofrece una serie de servicios adicionales, como interrupciones, mensajes de error, etc... Del amplio espectro de modelos existentes en el mercado hay dos que se destacan por su aceptación: SCSI y AT-BUS. El interface SCSI está pensado para ser usado por gran cantidad de dispositivos, por lo que se perfila como la apuesta que acabará imponiéndose, a pesar de que su precio sea algo superior al del AT-BUS. Este último difiere en el comportamiento entre marcas de diferentes fabricantes, lo que hace que los lectores CD-ROM que se conecten a esta tarjeta se vendan junto a ella.

Una de las desventajas del CD-ROM es su *elevado tiempo de búsqueda*, esto es, el tiempo empleado en encontrar el sector requerido por la controladora. Esto se debe en gran medida a la velocidad de giro del lector. El CD-ROM se graba con una velocidad lineal constante, lo que significa que el láser recorre una vuelta de disco en un tiempo fijo, independiente de la posición en la que se encuentre. Para conseguir esto, el motor del lector debe variar de 3'5 r.p.s. en su parte más externa a 8 r.p.s. en las inmediaciones del

centro del disco. Así se evita que los sectores del disco situados al principio sean leídos más lentamente que los del final. De este modo se consigue reunir mayor cantidad de información que en los discos magnéticos, donde el motor siempre gira a la misma velocidad; en este último caso, el disco sí que poseerá una velocidad angular constante, solucionando el problema anteriormente citado al variar el tamaño de los sectores.

El hecho de variar la velocidad de giro tiene su consecuencia más directa en el tiempo de búsqueda entre sectores. Este tiempo es considerablemente mayor (hasta 20 veces más) que en los discos magnéticos, dado que el láser, además de tener que situarse en el sector buscado, debe esperarse a que el disco adquiera la velocidad correcta, proceso no demasiado rápido debido a la fuerza de inercia.

Otro tiempo a tener en cuenta es el de *transferencia de datos* del disco al ordenador. La tasa media en condiciones normales es de 150 Kilobytes por segundo (velocidad del CD-ROM del laboratorio) aunque de un tiempo a esta parte han surgido reproductores de doble, triple y hasta cuádruple velocidad, que pueden llegar a transmitir 600 Kb/seg. Esto se debe a un aumento en la velocidad del motor, aunque hay que tener en consideración que a mayor velocidad mayor inercia poseerá el disco.

Actualmente se está empezando a considerar al CD-ROM como algo más que una mera opción con la que ampliar la capacidad del ordenador en el futuro. Para constatarlo, sólo hace falta fijarse en el impacto que ha tenido en las últimas versiones de los sistemas operativos más difundidos. El sistema MS-DOS 6 incluye un archivo llamado MSCDEX.EXE destinado a la gestión del CD-ROM. El extendido interface gráfico Windows posee en su versión 3.1 y 3.11 (la instalada en el equipo del laboratorio) una serie de extensiones dedicadas por entero al entorno multimedia, como puede observarse en el

Panel de Control o en algunos iconos del grupo Accesorios. Por último, aunque no por ello menos importante, se encuentra el sistema OS/2 2.1. Este gracias a su concepción de multitarea, a sus 32 bits de proceso y a una serie de herramientas desarrolladas especialmente para explotar al máximo las posibilidades de los nuevos dispositivos, puede considerarse como un sistema operativo verdaderamente multimedia.

Cuando se decide incluir información digital en un CD-ROM, resulta evidente que ésta no necesita someterse a ningún tratamiento de conversión. Pero cuando aparecen elementos como el sonido, que en principio es una onda analógica continua, se hace necesario digitalizarlos para poderlos almacenar en formato digital. En el caso de los compact disc, la frecuencia de muestreo empleada es de 44100 Hz a 16 bits para cada uno de los 2 canales (izquierdo y derecho), resultando un total de 1'4 millones de bits por segundo y más de 10 Megabytes por minuto.

El CD-ROM es una variedad del disco óptico, del que existen *distintos formatos* y por tanto conviene enmarcarlo debidamente:

CD-A (CD Audio): es el formato de los discos compactos digitales cuyas normas fueron definidas por las empresas Philips y Sony.

CD-ROM (CD Read Only Memory): es una extensión del formato CD-A a la que se ha complementado con la capacidad de almacenar ficheros con todo tipo de datos.

CD-ROM-XA (CD-ROM extended Architecture): Microsoft intervino junto a Philips y Sony en el desarrollo de la extensión del CD-ROM, caracterizada por poseer canales entrecruzados para el almacenaje de datos y audio. Una unidad completamente compatible con este formato deberá ser capaz de reproducir el canal de audio y leer los datos grabados



simultáneamente. Como es comprensible, este formato es el corazón de las aplicaciones multimedia.

La capacidad del CD-ROM para almacenar una gran cantidad de imágenes en formato digital impulsó a Kodak a idear el servicio Kodak Photo-CD mediante el cual una persona que desee visualizar sus fotografías en la pantalla del ordenador o del televisor, no tiene más que entregar sus negativos o diapositivas a Kodak para recibir un CD-ROM-XA con resoluciones en sus fotos que alcanzan hasta los 2048x3072 puntos con una gama de 16 millones de colores. Si no se dispone de ordenador con unidad de CD-ROM, Kodak suministra un reproductor similar al CD-I para la visualización de fotos en el televisor.

CD-I (CD Interactive): Philips describe este formato de disco compacto, que es capaz de almacenar datos, canal de audio e imágenes. En este caso el dispositivo de visualización no es el monitor del ordenador, sino un aparato común de televisión, al que se conecta el reproductor de discos CD-I. Un mando a distancia permite interactuar con dicho reproductor para elegir la secuencia de imágenes y sonido que queramos contemplar. Con un dispositivo adicional, que también distribuye Philips, es posible visualizar discos compactos de verdadero vídeo en movimiento.

El CD-ROM, al igual que en compact disc de audio, representa los bits como una serie de hendiduras en una capa de aluminio. El láser atraviesa un revestimiento de policarbonato para después reflejarse en el aluminio, reflejo que será distinto si se ha incidido en una hendidura o no. Estos cambios son interpretados en el reproductor como cambios de ceros a unos y viceversa. Los bytes leídos se van almacenando en el buffer que, cuando alcanza su valor límite, transmite su contenido a la controladora. Ésta se encarga de

suministra al programa la información pedida, que según el archivo, se tratará de texto, audio, imagen, vídeo, etc...

## *Grabación del CD-ROM*

Para grabar toda la información del trabajo fin de carrera en un CD-ROM se buscaron las empresas que ofrecían este servicio. De una lista de unas 10, finalmente se decidió hacer el encargo a CD-EASY de Madrid, por el precio ajustado que ofrecía y la rapidez de servicio (1 día para grabarlo y volver a mandarlo).

El envío de la información se llevo a cabo en una cinta de 250 Mb y el software empleado para volcarla lo remitió la propia empresa y es el programa Tracker de la casa Colorado. El precio de la grabación se establece según el siguiente baremo:

Grabación del 1er. disco:

Completo.....:	7000 ptas.
Por megas:.....de 1 a 100:	3500ptas.
.....de 101 a 600:	3000+8 ptas/mega.

Grabación de las copias: 3500 ptas/copia.

En esta empresa no podían realizar serigrafía sobre el disco, tan sólo en aquellas donde se exigían volúmenes mínimos de más de 50 CD-ROM y el precio era mucho mayor.

## 10. Características del equipo y posibles problemas.

El equipo del laboratorio donde se ha preparado el trabajo y donde se espera emplear es de sobra necesario para la ejecución de todo el software que contiene el CD-ROM.

Cuanto más potente sea el equipo, más rápido funcionarán todos los programas y con mejor calidad se verán las imágenes.

La configuración del equipo mínimo para emplear este trabajo es la siguiente:

- Ordenador PC con microprocesador 286 ó superior. Al menos 4 Mb de memoria RAM, aunque se aconsejan 8 Mb. Tarjeta de vídeo VGA o superior (se recomienda 1 Mb de memoria; esta característica es importante para que el entorno Windows sea fluido). Ratón.

- Tarjeta de sonido compatible con Soundblaster que funcione bajo Windows. Puede ser de 8 bits y mono.

- Unidad lectora de CD-ROM. No es necesario que sea de doble o triple velocidad. Bastará el modelo más sencillo.

## *Posibles problemas:*

Durante la realización del trabajo aparecieron bastantes problemas, principalmente con incompatibilidades de programas y tarjetas.

Una de las dificultades ha sido encontrar todos los drivers adecuados para que los programas puedan funcionar en la misma sesión y no aparezcan conflictos en sus dispositivos. Estos problemas están totalmente subsanados y en cada subdirectorío se entregan los drivers que puede necesitar otro equipo. El ordenador del laboratorio de TVII ha quedado configurado adecuadamente dejando paso a todas las herramientas multimedia que hay instaladas y que pueden instalarse posteriormente.

Otro detalle, pero a nivel de hardware, apareció con la instalación de todos los accesorios en el equipo de prácticas. Al cohabitar 2 unidades de disco flexible, un disco duro, una unidad de backup, el CD-ROM y además las tarjetas de entrada/salida de audio y vídeo, SoundBlaster ASP-16 y la Targa Plus de TrueVision, fue necesaria una buena distribución de la interrupciones. Comentar que se realiza la conexión interna del CD-ROM a la tarjeta de audio para facilitar la grabación de sonidos desde discos compactos directamente al ordenador.

El problema con todos estos dispositivos es que consumen gran parte de la memoria que el sistema operativo ofrece (640 Kb), reduciendo las prestaciones y velocidad del equipo; principalmente el reproductor de CD-ROM con su fichero MTMCDE que reside en memoria ocupando mas de 57 Kb, y 2 ficheros de la Targa, TARGAP y TPLUSTSR con 10 y 22 Kb respectivamente.

Por último, comentar que aunque el conjunto de programas shareware incluido en el proyecto es reducido, se han empleado otros muchos programas para realizar la captura de sonidos, tratamiento de los mismos, edición del texto, creación de los iconos en windows, captura y retoque de algunas imágenes, etc... También se han consultado otro gran número de programas en busca del software adecuado para cada labor, y para el aprendizaje de las novedosas tecnologías multimedia que están surgiendo con rapidez.

## **11. Bibliografía.**

La bibliografía consultada han sido determinados libros de procesado digital de imágenes, pero también se han consultado numerosas revistas, entre las que cabe destacar PCMEDIA, PCMANÍA, PRACTUAL y PCWORLD.

Gran parte de la información sobre la existencia y manejo de los programas multimedia se obtuvo de los ficheros de ayudas y propagandas de software que se encuentran en CD-ROM's. Muchos de estos CD-ROM's se incluían en las revistas antes mencionadas.

Los libros consultados fueron los siguientes:

- **Tratamiento Digital de Imágenes.**

Alberto Domingo Ajenjo, editorial Anaya Multimedia

- **Digital Image Processing.**

González y Wintz, Addison-Wesley Publishing Company

- **Digital Image Processing.**

William B. Green, Van Nostrans Reinhold

- **Fundamentals of Digital Image Processing.**

Anil K. Jain, Prentice Hall International Edition

- **Digital Image Restoration.**

H.C. Andrews, B.R. Hunt, Prentice Hall Signal Processing Series

- **An Introduction to Digital Image Processing.**

Wayne Niblack, Prentice Hall International

- **Digital Television.**

C. P. Sandbank, John Wisley & Sons

- **Speech Communication (human and machine).**

O'Shaughnessy, Addison Wesley Publishing Company

- **Morphological Methods in Image and Signal Processing.**

Giardina & Dougherty, Prentice Hall

## 12. Agradecimientos.

El autor quiere expresar su gratitud a la gente que ha prestado su colaboración desinteresada para la realización de este proyecto:

En primer lugar y muy especialmente quiero agradecer a mi hermano Jose, ya no sólo el apoyo y la gran ayuda para la elaboración de este trabajo, suministrándome todos los CD-ROMs que le fue posible, grandes cantidades de software para consultas y búsqueda de utilidades e ideas, revistas para PC y prestándome su equipo informático, reforzado con accesorios multimedia para esta ocasión, sino por la enseñanza que ha volcado en mí durante más de 11 años, desde los maravillosos tiempos del PRINT AT 10,10;" del ZX Spectrum. Creo que este trabajo multimedia grabado en un CD-ROM es la culminación de todas esas horas sentados juntos frente al monitor, creando programas de visualización tridimensional de funciones, objetos y letras, programas de diseño, generación de melodías, juegos y hasta programas de gestión de ficheros, etc... De algún modo este trabajo es también tuyo. Siento las crisis informáticas y las deserciones para programar en la HP.

También quiero agradecer a Dioni el apoyo permanente, los buenos consejos y la confianza depositada en el proyecto. De él espero que haga buen uso de este trabajo en las prácticas de la asignatura. Quedan muchos otros temas de tratamiento digital de imágenes médicas en proyecto, que espero que podamos continuar juntos. Por cierto, tenemos que presentar a Alejandro y a Ana.



Agradecer la colaboración a todos los profesores de Imagen y Sonido, que además me han soportado durante 2 intensos meses en el laboratorio. Gracias Cubero, Fátima, Pepe, Eduardo, Juanma, Fidel y Elena.

A Norberto la colaboración en el tema de ayudas de windows que finalmente no se llevó a cabo.

A los compañeros que me han suministrado imágenes fundamentales en el trabajo: Rafa, por sus imágenes "elite" y muchos otros programas, Domingo y Duli, por sus imágenes astronómicas y Anas por su imagen "horse3d" que aún no he conseguido ver.

A la gente que me ha proporcionado software durante largo tiempo como los gemelos Roberto y Antuán, Matías y Óscar que me proporcionó el toolbox de images para matlab, que resultó crucial.

A los amigos que han ido viendo crecer el trabajo aconsejándome y animándome, Carlos y las Perks en la etapa de Valencia; Antonio, Piedad, Manolo, Alberto y otros muchos en Las Palmas.

A mis hermanos y a mis padres que me han animado a trabajar duro para terminar pronto y volver a casa, aunque os sigáis preguntando qué demonios estoy haciendo de proyecto. Yo también os he echado mucho de menos.

*Gabi, your flügel are my flügel too. Thank you for giving me the drive that I needed.*