

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA



UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

**PROYECTO FIN DE
CARRERA**

INTERFAZ GRÁFICA DE USUARIO PARA EL REGISTRADO DE
IMÁGENES MÉDICAS

JOSÉ CARLOS RODRÍGUEZ PÉREZ

Las Palmas de Gran Canaria, diciembre 2002

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

ESCUELA UNIVERSITARIA DE
INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN



PROYECTO FIN DE CARRERA

INTERFAZ GRÁFICA DE USUARIO
PARA EL
REGISTRADO DE IMÁGENES MÉDICAS

ESPECIALIDAD : Sonido e Imagen

TUTORES : D. Juan Ruiz Alzola y D. Eduardo Suárez Santana

AUTOR : Jose Carlos Rodríguez Pérez

FECHA : Diciembre de 2002



| | |
|--------------------------|--------|
| BIBLIOTECA UNIVERSITARIA | |
| LAS PALMAS DE G. CANARIA | |
| N.º Documento | _____ |
| N.º Copia | 698773 |

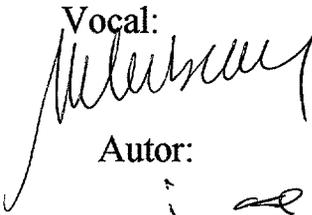
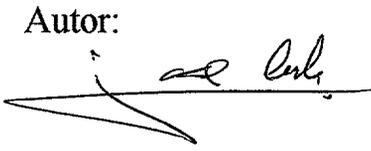
UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

**ESCUELA UNIVERSITARIA DE
INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN**



PROYECTO FIN DE CARRERA

**INTERFAZ GRÁFICA DE USUARIO
PARA EL
REGISTRADO DE IMÁGENES MÉDICAS**

Presidente:  Secretario:  Vocal: 
Tutores:   Autor: 
NOTA : M.H. (10)

ESPECIALIDAD : Sonido e Imagen

TUTORES : D. Juan Ruiz Alzola y D. Eduardo Suárez Santana

AUTOR : Jose Carlos Rodríguez Pérez

FECHA : Diciembre de 2002

A mis padres, por su apoyo incondicional, paciencia y aliento en los momentos difíciles.

A Eduardo por el tiempo que me regaló en estos meses.

A Maxi y Ricardo por su valiosa ayuda.

A Laura por sus consejos artísticos.

AGRADECIMIENTOS

INDICE

| | |
|--|-----------|
| MEMORIA | 5 |
| 1. INTRODUCCIÓN..... | 5 |
| 2. OBJETIVOS..... | 7 |
| 3. SOFTWARE PARA EL REGISTRADO DE IMÁGENES MÉDICAS | 8 |
| 3.1. INTRODUCCIÓN..... | 8 |
| 3.2. IMÁGENES MÉDICAS | 8 |
| 3.2.1. Rayos X..... | 9 |
| 3.2.2. Tomografía computerizada | 10 |
| 3.2.2.1. Tomografía computerizada helicoidal..... | 10 |
| 3.2.3. Resonancia magnética..... | 11 |
| 3.3. EL REGISTRADO EN LA CADENA DE PROCESADO | 12 |
| 3.4. APLICACIONES DE REGISTRADO..... | 14 |
| 3.4.1. Modalidades involucradas..... | 15 |
| 3.4.2. Objeto del registrado..... | 16 |
| 3.5. INTERFACES GRÁFICAS DE USUARIO | 16 |
| 3.5.1. Principios para el diseño | 18 |
| 3.5.2. Interfaces gráficas para software de registrado médico..... | 20 |
| • 3D Slicer..... | 21 |
| • Analyze 4.0..... | 23 |
| • Amira™ 2.3. Advanced 3D Visualization and Volume Modelling..... | 25 |
| • MNI AutoReg..... | 29 |
| • CISG..... | 31 |
| • AIR..... | 33 |
| • SPM..... | 35 |
| 4. REGISTRADO..... | 37 |
| 4.1. INTRODUCCIÓN..... | 37 |
| 4.2. CLASIFICACIÓN DE LOS MÉTODOS DE REGISTRADO..... | 37 |
| 4.3. DIMENSIONALIDAD | 39 |
| 4.3.1. Métodos de registrado espacial..... | 39 |
| 4.3.2. Registrado de series temporales..... | 40 |
| 4.4. NATURALEZA DE LAS BASES DEL REGISTRADO..... | 40 |
| 4.4.1. Métodos de registrado extrínsecos..... | 41 |
| 4.4.2. Métodos de registrado intrínsecos..... | 42 |
| 4.4.2.1. Métodos basados en puntos de control o landmarks..... | 42 |
| 4.4.2.2. Métodos basados en la segmentación..... | 43 |
| 4.4.2.3. Métodos basados en las propiedades del vóxel..... | 45 |
| 4.4.2.4. Métodos no basados en imágenes..... | 46 |
| 4.5. NATURALEZA DE LA TRANSFORMACIÓN..... | 47 |
| 4.6. DOMINIO DE LA TRANSFORMACIÓN..... | 49 |
| 4.7. INTERACCIÓN | 50 |
| 4.8. PROCEDIMIENTO DE OPTIMIZACIÓN..... | 50 |
| 4.9. MODALIDADES INVOLUCRADAS | 51 |
| 4.10. SUJETO | 52 |

| | | |
|-----------|---|-----------|
| 5. | DISEÑO DE UNA HERRAMIENTA DE REGISTRADO | 52 |
| 5.1. | INTRODUCCIÓN | 52 |
| 5.2. | TIPOS DE DATOS | 53 |
| 5.3. | FLUJO DE PROCESADO | 59 |
| 5.4. | PROCESOS | 60 |
| | • <i>Selector manual de puntos clave</i> | 60 |
| | • <i>Sintonía fina de puntos clave</i> | 61 |
| | • <i>Registrado rígido de imagen</i> | 61 |
| | • <i>Registrado rígido de puntos clave</i> | 64 |
| | • <i>Registrado no rígido imagen</i> | 65 |
| | • <i>Registrado no rígido de puntos clave</i> | 65 |
| | • <i>Obtener Campo de deformación denso a partir de Transformación afin</i> | 67 |
| | • <i>Interpolación</i> | 67 |
| | • <i>Deformar a partir de Transformación afin</i> | 68 |
| | • <i>Deformar a partir de Campo de deformación disperso</i> | 68 |
| | • <i>Deformar a partir de Campo de deformación denso</i> | 68 |
| 5.5. | PLATAFORMA..... | 69 |
| 5.5.1. | Requisitos de implementación..... | 69 |
| 5.5.2. | Elección adecuada | 70 |
| 5.5.3. | Matlab®..... | 71 |
| 5.5.4. | Decisiones de implementación | 72 |
| | 5.5.4.1. <i>Sintonía fina de puntos clave</i> | 72 |
| | 5.5.4.2. <i>Registrado rígido de imagen</i> | 72 |
| | 5.5.4.3. <i>Registrado no rígido imagen</i> | 72 |
| | 5.5.4.4. <i>Obtener Campo denso a partir de Transformación afin</i> | 73 |
| | 5.5.4.5. <i>Interpolación</i> | 73 |
| | 5.5.4.6. <i>Deformar a partir de Transformación afin</i> | 74 |
| | 5.5.4.7. <i>Deformar a partir de Campo de deformación denso</i> | 74 |
| 5.5.5. | Empaquetado | 74 |
| | 5.5.5.1. <i>Ficheros de Datos</i> | 75 |
| | 5.5.5.2. <i>Ficheros de Procesado</i> | 75 |
| | 5.5.5.3. <i>Ficheros de interfaz</i> | 76 |
| 6. | MANUAL DE USUARIO..... | 77 |
| 6.1. | INTRODUCCIÓN | 77 |
| 6.2. | INSTALACIÓN Y EJECUCIÓN DE LA INTERFAZ | 78 |
| 6.3. | ENTORNO DE TRABAJO | 79 |
| 6.3.1. | Barra de estado | 79 |
| 6.3.2. | Menú de opciones | 80 |
| | • <i>Menú Archivo</i> :..... | 81 |
| | • <i>Menú Ver</i> :..... | 83 |
| | • <i>Menú Herramientas</i> : | 84 |
| | • <i>Menú Ventana</i> : | 84 |
| | • <i>Menú Ayuda</i> : | 85 |
| 6.3.3. | Barra de herramientas | 85 |
| 6.3.4. | Área de trabajo..... | 98 |
| | 6.3.4.1. <i>Modo Ventana Imagen</i> | 98 |
| | 6.3.4.2. <i>Modo Ventana Puntos Clave</i> | 100 |
| | • <i>Selector manual de puntos clave</i> | 102 |

| | | |
|--|--|------------|
| 6.3.4.3. | <i>Modo Ventana Imagen Deformada</i> | 105 |
| 6.3.4.4. | <i>Modo Ventana Transformación</i> | 106 |
| 6.3.4.5. | <i>Modo Ventana Campo Disperso</i> | 107 |
| 6.3.4.6. | <i>Modo Ventana Campo Denso</i> | 108 |
| 7. | CONCLUSIONES | 110 |
| 8. | LÍNEAS FUTURAS | 111 |
| APÉNDICES | | 113 |
| I. INFORMACIÓN PARA DESARROLLADORES | | 113 |
| 1. | ESTRUCTURA..... | 113 |
| 2. | FICHEROS..... | 117 |
| 2.1. | Ficheros de Datos..... | 117 |
| 2.2. | Ficheros de Procesado..... | 118 |
| 2.3. | Ficheros de interfaz..... | 122 |
| 3. | ACOPLAMIENTO DE UN NUEVO MÓDULO..... | 124 |
| II. CONTENIDO DEL CD | | 126 |
| BIBLIOGRAFÍA | | 127 |
| PLIEGO DE CONDICIONES | | 131 |
| PRESUPUESTO | | 133 |
| 1. | Trabajo tarifado por tiempo establecido..... | 133 |
| 2. | Coste de las licencias del software..... | 134 |
| 3. | Coste del hardware..... | 134 |
| 4. | Derechos de visado del COITT..... | 135 |
| 5. | Gastos de administración y envío..... | 135 |
| 6. | Coste total del proyecto..... | 136 |

MEMORIA

MEMORIA

1. INTRODUCCIÓN

La incorporación de nuevas tecnologías siempre ha sido un factor decisivo para el desarrollo y evolución de la medicina a lo largo de la historia.

Por esto el constante e imparable avance tecnológico experimentado en las últimas décadas por la tecnología de la informática, la electrónica y las comunicaciones, entre otras, no podía pasar desapercibido para la medicina.

El software médico se ha convertido en los últimos años, en parte importante de las herramientas utilizadas por la medicina moderna, involucrando en su desarrollo tanto a especialistas de la medicina como a técnicos. Destaca de forma especial aquel software especializado en el procesamiento de imágenes, dado que el análisis de imágenes biomédicas es cada vez más importante tanto, para la investigación en las Ciencias de la Salud como en la generación de diagnósticos.

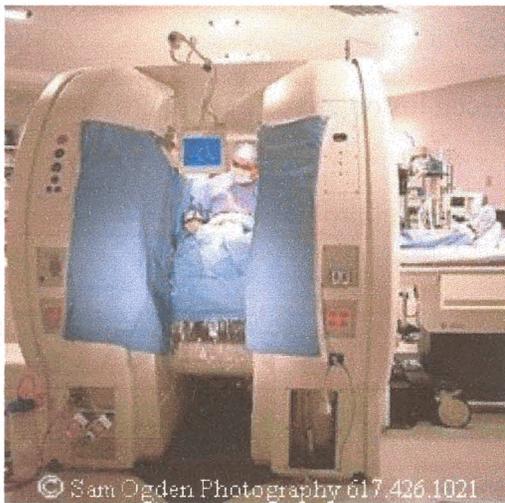


Figura 1.1. Las imágenes médicas pueden ser útiles para la planificación quirúrgica.

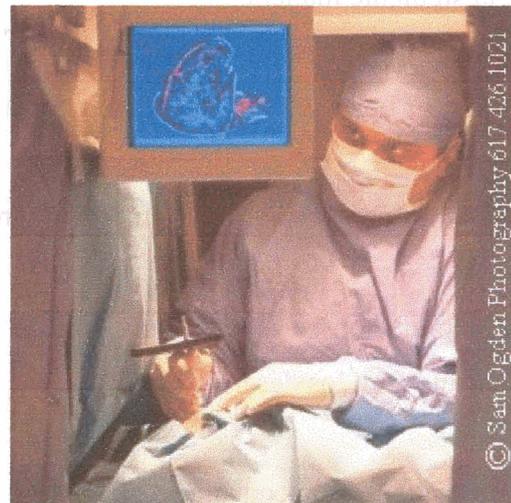


Figura 1.2. Un cirujano observa en el monitor LCD la imagen 3D de la zona a intervenir.

Cada vez más los profesionales de la salud necesitan herramientas de análisis de imágenes que ayuden a automatizar las tareas de investigación y diagnóstico.

Si bien el 70% de las imágenes médicas son todavía radiografías, cada vez hay un mayor incremento en todos los ámbitos de las Ciencias de la Salud del uso de imágenes digitales para un posterior tratamiento y almacenamiento: radiografías digitales, tomografías, resonancias magnéticas, ultrasonidos, medicina nuclear, entre otras, que ocupan el 30% restante [26].

En las últimas dos décadas han aparecido diversas modalidades de imágenes médicas que ofrecen a los médicos y profesionales sanitarios información valiosa, y en ocasiones imprescindible, para el diagnóstico de patologías, seguimiento de pacientes y planificación quirúrgica.

El presente trabajo está dedicado básicamente al desarrollo de una herramienta que permita llevar a cabo una técnica denominada **registrado de imágenes** (*image registration*) que no es más que el proceso mediante el cual se alinean dos conjuntos de datos o imágenes.

Este documento pretende servir de exposición del trabajo realizado así como de manual de usuario de la herramienta desarrollada. Como guía al lector el documento está organizado de la siguiente manera.

En primer lugar se hace una introducción y se establecen los objetivos de este proyecto.

Posteriormente se presenta la memoria propiamente dicha del proyecto.

A continuación se dispone una sección de apéndices en los que se incluye información relevante para los desarrolladores.

La documentación consultada se muestra en la sección de Bibliografía.

El pliego de condiciones, así como el coste de la aplicación *software* desarrollada, se presentan al final de este tomo.

2. OBJETIVOS

La meta final de este proyecto consiste en desarrollar una herramienta *software* orientada a los prototipos que permita al especialista realizar una operación de registrado de imágenes médicas.

La aplicación a desarrollar habrá de ser simple e intuitiva en el manejo, que permita modificar los algoritmos empleados de manera sencilla, así como facilitar la incorporación de módulos que añadan nuevos algoritmos.

Los algoritmos utilizados han de poder ser reutilizados en *software* posterior.

La aplicación a desarrollar será utilizada básicamente en laboratorios destinados a la investigación y desarrollo de *software* médico (de ahí el concepto *orientada a los prototipos*), por tanto, laboratorios que disponen de imágenes médicas y en los que se suele trabajar con diversos sistemas informáticos.

En resumidas cuentas podemos establecer que el objetivo de este proyecto es el diseño e implementación de una interfaz gráfica de usuario que permita:

- **Realizar el registrado de imágenes médicas.**
- **Ser integrada en un entorno de trabajo orientado a la investigación.**
- **Reutilizar el código para ser aplicado en la generación de nuevas interfaces.**
- **Validar diferentes algoritmos de registrado, sin necesidad de crear nuevas interfaces gráficas de usuario.**

3. SOFTWARE PARA EL REGISTRADO DE IMÁGENES MÉDICAS

3.1. Introducción

Una imagen médica representa una sección o proyección del cuerpo humano imposible de visualizar con una simple exploración física externa. Existen distintas técnicas para adquirir imágenes, gracias a las cuales es posible obtener información relevante que asista en el diagnóstico, tratamiento de un paciente o planificación quirúrgica. Es necesario entonces elegir correctamente la modalidad de imagen más adecuada, en función de la sospecha clínica.

3.2. Imágenes médicas

Para obtener una imagen de cualquier objeto o, en el caso que se presenta, de un órgano en particular, es necesario utilizar ondas (ya sean electromagnéticas o sonoras) que tengan un gran poder de penetración en los tejidos biológicos. Esta energía interacciona con el órgano y posteriormente se recoge en el receptor adecuado. Se utilizan, en función del objetivo que se persigue, diferentes tipos de radiación u ondas tanto electromagnéticas (rayos X) como mecánicas (ultrasonidos). Si estas ondas proceden del propio paciente, se habla de imagen por emisión, mientras que si proceden de una fuente externa de radiación se trata de imagen por reflexión. Al proceder, atravesar o reflejarse en los distintos tejidos del organismo, las ondas contienen información que permite obtener una imagen del mismo. Para ello, cada tipo de energía debe recogerse en un detector específico, sensible a dicha energía.

La obtención de una imagen se lleva a cabo gracias a un equipo radiológico preparado para ello. A cada equipo de adquisición de imagen radiológica se le denomina *modalidad* y, dependiendo de ésta, se obtendrá un tipo de imagen u otro.

Por ejemplo en la tomografía computerizada la formación de imagen se inicia a través de un detector de rayos X, que en función de la radiación recibida induce una corriente eléctrica analógica, que posteriormente es digitalizada. En la ecografía es el eco recibido por un transductor de ultrasonidos el encargado de inducir esa corriente. En la resonancia magnética las diferencias de magnetismo son captadas por una antena receptora y, como ocurre en los otros dos casos, induce una corriente eléctrica analógica que posteriormente es digitalizada.

Estas modalidades son consideradas técnicas digitales porque la respuesta analógica es digitalizada antes de la constitución de la imagen radiológica. Pero existe también la posibilidad de digitalizar una película convencional. En este caso se puede hacer mediante una cámara de vídeo, un escáner de barrido normal o con un escáner de barrido láser. Con este último se obtienen las resoluciones superiores. La calidad final de la digitalización de una imagen analógica dependerá no sólo del método de digitalización de una imagen analógica sino también de la calidad de la imagen original.

Aunque existen muchos tipos de imágenes, se analizarán principalmente los tipos de imagen médica de los cuales se han hablado hasta ahora. Estos son cuatro:

La radiografía convencional (Rx), la tomografía computerizada (TC) y la resonancia magnética (MR).

3.2.1. Rayos X

Los rayos X se originan cuando los electrones inciden con gran velocidad sobre la materia y son frenados repentinamente, cediendo o perdiendo energía. Cuando un haz de rayos X incide sobre la materia (radiación incidente), parte de esta radiación se absorbe, parte se dispersa y parte no se modifica y atraviesa la materia. Es precisamente esta propiedad de absorción la que permite la posterior visualización de la materia. Porque, en función a la composición del organismo que es atravesado por un haz de rayos X, éste se atenúa con mayor o menor intensidad, lo que sale del mismo es otro haz de rayos X, pero esta vez heterogéneo, con distintas intensidades.

Este fenómeno es el que da origen a la imagen radiológica, que puede ser visualizada bien en una película fotosensible especial (imagen permanente) o bien en una pantalla fluorescente (imagen transitoria).

Pero la imagen radiológica es la representación bidimensional de una estructura tridimensional. Por tanto, en una radiografía simple las imágenes de todos los planos se superponen. En ocasiones la imagen de un plano oscurece la de otro, lo que plantea un problema de interpretación. Gracias a la tomografía convencional, este inconveniente ha quedado solucionado. Esta técnica proporciona una imagen por planos o secciones del órgano afectado, que pueden ajustarse en cuanto a localización y grosor.

3.2.2. Tomografía computerizada

En la Tomografía Computerizada (TC) el haz de rayos X que atraviesa al paciente describe una circunferencia alrededor de éste.

El sistema dispone de una corona de detectores que, a la vez que gira junto con el tubo de rayos X, realiza múltiples medidas de la intensidad transmitida, tomando valores de la atenuación de los rayos X en la sección explorada.

Los datos adquiridos son almacenados en el ordenador, en una matriz $N \times N$ cuyos elementos son proporcionales a la atenuación experimentada en cada estructura responsable de dicha atenuación.

3.2.2.1. Tomografía computerizada helicoidal

Se trata de una mejora de la Tomografía Computerizada. En este caso, los valores de atenuación no se toman a lo largo de una circunferencia, sino de una hélice.

Este sistema se basa en la rotación constante dentro de una carcasa, de un tubo de rayos X que emite una radiación continua, mientras la mesa donde se acuesta el paciente se desplaza a lo largo de dicha carcasa.

Este sistema permite una rápida adquisición volumétrica de datos y minimiza las distorsiones producidas por la respiración y el movimiento.

En la Figura 3.1. se puede ver una representación de este sistema.

Los rayos X trazan una hélice de proyecciones de la cual se generan imágenes multiplanares.

Para obtener una imagen auténticamente axial los datos recogidos por encima y por debajo del plano seccionado se interpolan, y luego se analizan en cada plano axial. De este modo se pueden obtener imágenes de calidad idéntica en cualquier plano de reconstrucción.

Mediante este sistema es posible explorar una región de varios centímetros en un pequeño periodo de apnea del paciente. Adecuando el tiempo de adquisición se obtienen estudios completos del tórax o del abdomen superior en un único período de apnea del paciente.

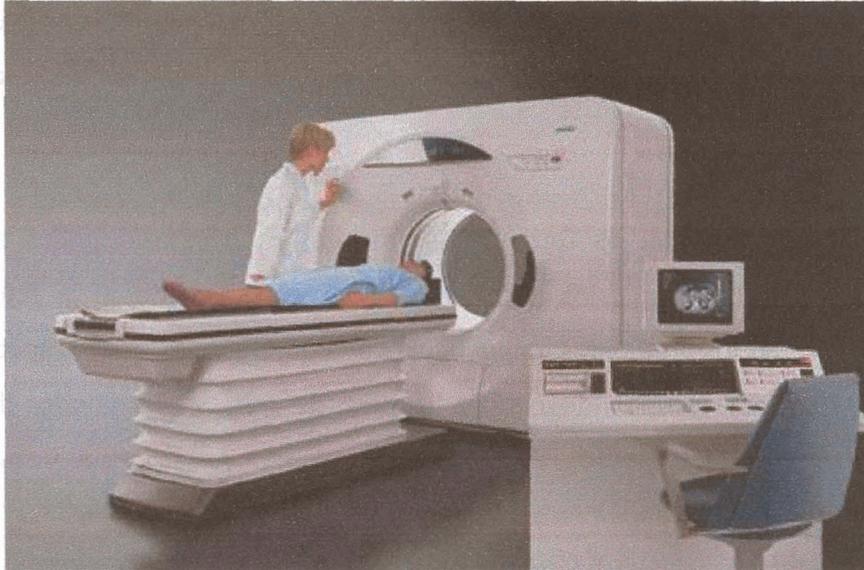


Figura 3.1.

3.2.3. Resonancia magnética

En la resonancia magnética (MR) se utiliza una combinación de imanes grandes, radiofrecuencias y una computadora para producir imágenes detalladas de los órganos y las estructuras internas del cuerpo.

Se basa en la propiedad que tienen los núcleos de hidrógeno de absorber energía electromagnética cuando están sometidos a un campo magnético intenso. Por eso se utiliza en los casos en los que se quieran visualizar tejidos blancos como los del corazón o los del cerebro, puesto que éstos tienen una mayor composición de agua por el organismo.

La diferencia con la imagen de TC, es que en ésta se representa un mapa de la densidad de electrones de los tejidos incluidos en el plano de examen, mientras que en MR se representa un mapa de la densidad de protones.

La máquina de MRI crea un fuerte campo magnético alrededor del paciente. Este campo magnético, junto con una radiofrecuencia, altera el alineamiento natural de los átomos de hidrógeno en el cuerpo. Posteriormente se usan computadoras para formar imágenes bidimensionales del órgano o estructura del cuerpo basadas en la actividad de los átomos de hidrógeno. Pueden obtenerse vistas transversales para revelar más detalles. Las imágenes por resonancia magnética no utilizan radiación, como lo hacen los rayos X o la TC.

Puesto que el objetivo final es formar una imagen, es necesario poder asignar una localización espacial correcta a los núcleos de hidrógeno que emitan la señal. De este cometido se encargan los gradientes magnéticos, cuyo objeto es codificar espacialmente la señal de MR. Producen campos magnéticos lineales, de pocos milisegundos de duración, multidireccionales, que pueden dirigir a lo largo de cada uno de los ejes x, y, z del espacio, y con variaciones de aproximadamente 10 militeslas por metro.

3.3. El registrado en la cadena de procesado

Las imágenes médicas se caracterizan por poseer gran cantidad de ruido y dado los diferentes sistemas captadores, CT, MR, PET, etc., cada cual con sus características y funcionalidades, hace que generar información válida para ser analizada no sea una cuestión trivial.

El principal propósito del procesado de imágenes es proporcionar al analista una imagen en la que se hayan realizado ciertos elementos estructurales para así facilitar su interpretación, así como proporcionar otros subsidios, inclusive generando productos que puedan ser posteriormente sometidos a otros procesos.

Es al conjunto global de procesos aplicados a una imagen, lo que se denomina Cadena de Procesado, representada de manera esquemática en la Ilustración 3.1.

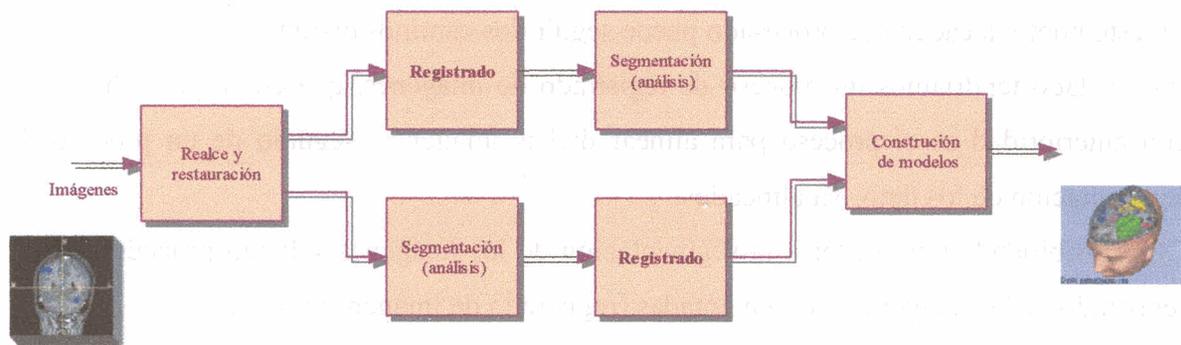


Ilustración 3.1

Dado que la imagen adquirida presenta ruido (i.e. la aparición de información no deseada en la imagen) es por ello que es necesaria una etapa de restauración en la que se trata de eliminar el ruido. Generalmente se recurre al **filtrado** de la imagen. Existen multitud de filtros diferentes, no sólo para eliminar ruido, sino que algunos se utilizan para realzar detalles de una imagen, para detectar determinadas características, o para explorar las propiedades geométricas de la imagen.

Las técnicas de filtrado son transformaciones de la imagen *píxel a píxel*, que no dependen únicamente de los niveles de intensidad de un determinado píxel central, sino que tienen en cuenta también los niveles de los píxeles vecinos del citado píxel central. No nos detendremos a exponer las distintas técnicas de filtrado pues se sale del ámbito de este proyecto. No obstante comentaremos algunas de las técnicas que se pueden utilizar para eliminar ruido.

El proceso de filtrado se realiza generalmente por medio de la convolución de la imagen con unas matrices denominadas *Máscaras*. El ruido generalmente se presenta como píxeles con un valor de intensidad sensiblemente diferente al de sus vecinos, bien sean niveles muy bajos (puntos negros) o niveles muy altos (puntos blancos). La distribución de estos puntos es un factor que es necesario tener en cuenta a la hora de elegir la máscara adecuada, así, dichos puntos pueden aparecer aleatoriamente distribuidos o hacerlo siguiendo un patrón, como puede ser en franjas horizontales o verticales (algo que puede deberse a fallos en el sistema captador). Los filtros pueden ser isótropos (si se filtra en todas las direcciones por igual) o anisótropos [7]. En el caso particular de las imágenes médicas, suele ser más conveniente la utilización de los anisótropos puesto que al filtrar tan sólo en una dirección, son capaces de discriminar mejor el ruido.

En este punto la cadena de procesado puede seguir dos caminos distintos.

Por un lado tendríamos un proceso de registrado de imágenes, que como ya se ha dicho con anterioridad es el proceso para alinear dichas imágenes, seguido de un proceso de segmentación de los datos ya alineados.

Otra posibilidad es proceder a la segmentación de las imágenes y luego proceder con el registrado de las imágenes ya segmentadas (registrado de imágenes binarias).

El proceso de segmentación se define como la división de la imagen en regiones. Es uno de los primeros pasos que conducen al análisis e interpretación de la imagen. El objetivo es agrupar en una región todas aquellas áreas de la imagen que son significativas para una determinada tarea específica. Esto permite por ejemplo la detección de órganos que aparecen en imágenes CT o MR.

Por último en algunos casos se puede proceder a la construcción de modelos en 3D, mediante la eliminación de áreas ocultas, la asignación de sombreados o transparencias y la proyección en perspectiva desde un ángulo determinado.

3.4. Aplicaciones de registrado

Existen distintas modalidades de imágenes biomédicas que van desde los rayos X, hasta la resonancia magnética (MRI *magnetic resonance imaging*), pasando por ultrasonidos (US), tomografías computerizadas (TC, conocida también como tomografía axial computerizada TAC, o por sus siglas en ingles CT, CAT), o la tomografía por emisión de positrones (PET *Positron emission tomography*).

Dado que la información que se obtiene a partir de dos imágenes de distintas modalidades para el seguimiento de los eventos clínicos, generalmente es de naturaleza complementaria, es de gran utilidad una correcta *integración* de los datos obtenidos.

Un primer paso en el proceso de integración es alinear espacialmente las imágenes involucradas, es dicho proceso el que se denomina **registrado**. Con posterioridad al registrado, es necesario un proceso de **fusión**, para la integración visual de los datos involucrados.

Desafortunadamente, términos como registrado, fusión, integración, correlación o emparejamiento, aparecen con frecuencia en la literatura especializada con el mismo significado, unas veces refiriéndose a un paso singular o a todo el proceso de integración de modalidades. En éste documento se utilizarán las definiciones de registrado y fusión tal y como se han descrito anteriormente.

Un ejemplo significativo del uso del registrado de imágenes médicas de diferentes modalidades, denominado **registrado multimodal**, se puede ver en el área de los tratamientos radioterapéuticos, donde los pacientes se ven sometidos a exploraciones de resonancia magnética y tomografía computerizada. La primera modalidad es ideal para delinear el tejido tumoral, mientras que la segunda es necesaria para un cálculo preciso de la dosis de radiación.

Aparte del registrado multimodal, existen importantes áreas de aplicación para el **registrado monomodal**, es decir, aquel en el que las imágenes son de una misma modalidad. Ejemplos de esto podrían ser la verificación del tratamiento, comparando imágenes pre- y post-operatorias, o el seguimiento de la evolución de un tumor cerebral, comparando series temporales de imágenes de resonancia magnética.

Puede hablarse de las posibles aplicaciones de registrado de imágenes desde el punto de vista de las modalidades involucradas, del objeto del registrado, etc.

3.4.1. Modalidades involucradas

De entre los distintos tipos de aplicaciones *monomodales* de registrado de imágenes podríamos destacar, CT (principalmente registrado 3D intrasujeto), registrado rígido y afin de MR, registrado de Rayos-X y PET (mayoritariamente registrado rígido 3D).

Si hablásemos de las aplicaciones *multimodales*, el registrado mediante puntos clave CT-MR, el registrado rígido 3D CT-PET, el registrado rígido PET-MR y el registrado de Rayos-X con MR o CT serían los más importantes. Otro tipo de aplicación es el registrado de modalidad con Atlas anatómicos, en el que por lo general la transformación suele ser de tipo curvo. Como se ha dicho con anterioridad el Atlas no es más que un modelo obtenido del análisis estadístico a partir de un considerable número de sujetos.

3.4.2. Objeto del registrado

En función del objeto de la imagen a registrar, es decir, de la anatomía en concreto que va a sufrir el proceso de registrado, destacamos los siguientes tipos.

- Cabeza
 - Cerebro o cráneo
 - Ojo
 - Dental
- Tórax
 - Completo
 - Cardíaco
 - Pecho
- Abdomen
 - General
 - hígado
 - riñón
- Pelvis y perineo
- Extremidades
 - General
 - Fémur
 - Húmero
 - Mano

3.5. Interfaces gráficas de usuario

Cuando en los laboratorios de Xerox Corp. en Palo Alto Florida (EEUU) desarrollaron hacia finales de la década de los años 70 su nueva concepción de *apuntar y pulsar*, estaban seguros de que revolucionarían la forma de relación entre el usuario y el computador, dando paso a una nueva relación humano máquina [15].

Se había creado la primera interfaz gráfica de usuario (GUI *Graphical User Interface*).

Hasta ese momento el campo de la informática estaba dominado por las interfaces de línea de comandos (i.e., aquellas que exigen al usuario introducir instrucciones breves mediante un teclado), limitando su acceso a usuarios de nivel medio dada la complejidad de su uso.

Con la interfaz gráfica de usuario GUI, se posibilitó de manera considerable el acceso de millones de personas al mundo de la informática.

La tecnología de interfaz gráfica de usuario, replanteo la forma de establecer el nexo humano computadora, el lenguaje se hizo comprensible y accesible, pero sobre todo atractivo. La atracción que provocan las ventanas de navegación, en una aplicación desarrollada por tecnología gráfica de usuario, es un punto clave para cautivarlos y hacer que éstos usen sus recursos.

La GUI permite al usuario elegir comandos, iniciar programas y ver listas de archivos y otras opciones utilizando las representaciones visuales (íconos) y las listas de elementos del menú. Las selecciones pueden activarse bien a través del teclado o mediante un dispositivo apuntador conocido como *mouse* o ratón.

Los elementos de un GUI incluyen elementos tales como ventanas, menús extensibles, botones, iconos, etc.

La interfaz tiene dos principales "misiones": la primera es informar al usuario acerca de los elementos que tiene a su disposición y para qué sirven, y la segunda es brindar la posibilidad de elegir entre las distintas opciones.

El uso de las GUI es más sencillo que el de las interfaces de línea de comandos. Sin embargo, la introducción de instrucciones con una GUI es más lenta, por lo que las GUI suelen tener la opción de emplear un sistema equivalente al de línea de instrucciones como alternativa rápida para los usuarios más expertos.

La interfaz gráfica de usuario juega un papel preponderante en el desarrollo y puesta en marcha de todo sistema. Es la carta de presentación del mismo y en ocasiones resulta determinante para la aceptación o rechazo de todo un proyecto. Según un estudio del Centro de Investigación y Estudios Avanzados de México, se estima que, en promedio, del 35 % al 45% de los gastos destinados a un proyecto son dirigidos al diseño de la interfaz.

3.5.1. Principios para el diseño

Una buena interacción de manipulación directa da el sentido de control de la aplicación al usuario. Un buen ejemplo de esto puede ser un panel de control o preferencias en el cual el usuario pueda cambiar los colores de la aplicación, etc.

Shneiderman (1992) propuso en su terminología los siguientes principios para describir la manipulación directa:

- Continua representación de los objetos y acciones de interés.
- Acciones físicas o presión de botones etiquetados en lugar de sintaxis compleja.

Mediante la manipulación directa se puede evitar el uso de sintaxis compleja, por lo tanto, se reducen los errores cometidos en su uso, ayudando a los novatos a aprender más rápidamente la funcionalidad básica de la herramienta. Sin embargo, la velocidad en usuarios experimentados se ve reducida si el usuario es forzado a utilizar sólo la manipulación directa cuando otro tipo de interacción puede ser más rápida.

La interfaz de la herramienta debe de ser diseñada para diferentes tipos de usuarios, desde el usuario nuevo e inexperto hasta el usuario más experto que deberá de trabajar muchas horas. Existen métodos con los cuales los sistemas pueden lograr esto, como pueden ser:

- El uso de aceleradores
- La disponibilidad de ayuda al usuario
- Minimizar memorización
- Ocultar la complejidad

Los usuarios nuevos comúnmente trabajan mejor con menús, formas y otros estilos que favorecen el aprendizaje de la aplicación. A los usuarios expertos, en cambio la posibilidad de utilizar comandos a través del teclado o combinaciones de teclas, les permite una mayor rapidez. Estos estilos de interacciones rápidas se denominan **aceleradores**.

El acelerador (combinación de teclas que se encuentra a la derecha de la opción en el menú desplegado) es utilizado para seleccionar una opción de una manera más rápida que la simple selección en el menú.

Proveer de **ayuda** al usuario, facilita la obtención adicional de información acerca de los conceptos de la aplicación, tareas, comandos y métodos utilizados, así como las técnicas de interacción. La ayuda debe ser llamada de la misma forma desde cualquier parte de la interfaz. Algunos sistemas de ayuda tapan el área de trabajo, forzando al usuario a memorizar el contexto a la hora de interpretar el mensaje de ayuda. Esto se puede evitar con el uso de ventanas, de modo que en una ventana se encuentra la aplicación y en otra se muestra la ayuda.

En algunos diseños de interfaces muchas veces se fuerza al usuario a memorizar mucha información que no es necesaria. Es importante involucrar al usuario en reconocer en lugar de memorizar. El escoger un objeto desplegado en la pantalla elimina la necesidad de memorización.

El activar una etiqueta (dando la descripción o función de lo que pasará presionándolo) después de ubicar durante varios segundos el cursor del ratón sobre un icono, es una buena manera de **minimizar la memorización** de la funcionalidad de los objetos representados.

Ocultar la complejidad permite a los usuarios noveles aprender los comandos básicos y comenzar su trabajo de manera productiva sin necesidad de conocer a fondo el sistema.

Si se quiere llevar a buen puerto la realización de una interfaz *amigable*, de fácil manejo y que respete su funcionalidad, es necesario considerar elementos como:

- Elegancia y simplicidad
- Proporcionalidad, contraste y escalamiento
- Organización estructural
- Modularidad
- Representación de imágenes
- Guías de estilo
- Estandarización
- Interacción
- Manejo de errores
- Ayudas

El diseño visual de interfaces involucra diferentes disciplinas. Es por esto por lo que se insiste en la necesidad de crear objetos que interactúen con el especialista de manera sencilla y eficaz, pues estos serán el principal medio de comunicación con el exterior.

Naturalmente, la elección de una u otra interfaz depende de la aplicación médica que se quiera realizar.

Por ejemplo, para manejar un programa mientras se efectúa una operación quirúrgica, es útil una interfaz cuyos botones sean grandes y bien visibles. O en caso de que se requiera visualizar al mismo tiempo una imagen médica desde distintos puntos de vista, será más práctico una interfaz que permita mostrar varias ventanas a la vez.

A modo de conclusión, convenimos que la herramienta debe presentar una interfaz de usuario sencilla e intuitiva, capaz de resolver el problema propuesto. Sin más botonera que la propia elección del algoritmo específico y su inmediata ejecución, minimizando el ajuste de parámetros y otras opciones que disperse y diluya el universo de la problemática.

Asimismo se proveerá a la herramienta de un manual de usuario que facilite el acercamiento del usuario novel a la aplicación, se utilizarán etiquetas informativas para minimizar la necesidad de memorización por parte del usuario, y se introducirá el uso de aceleradores que permitirán agilizar la productividad de aquellos usuarios más avanzados.

3.5.2. Interfaces gráficas para software de registrado médico

Actualmente existen múltiples paquetes software que presentan buenas interfaces de usuario con las que llevar a cabo actividades de procesado de imágenes médicas.

Son interfaces de visualización, registrado, segmentación, y cuantificación de datos médicos.

A continuación se comentarán algunas de ellas, centrándonos en aquellas que permiten aplicaciones de registrado de imágenes o datos médicos.

El 3D Slicer es un software de código fuente abierto, para la visualización, registrado, segmentación y cuantificación de datos médicos.

Se trata de un proyecto desarrollado mediante la colaboración del Laboratorio de Inteligencia Artificial del MIT (*Massachusetts Institute of Technology*), y el Laboratorio de Planificación Quirúrgica en Brigham & Women's Hospital, afiliado a la Harvard Medical School [20].

Como en la mayoría de este tipo de navegadores, dispone de interfaces de carga de volúmenes de datos, visualización de imágenes de distintas maneras, segmentación de imágenes, creación de modelos en 3D, almacenamiento de datos, etc.

La Figura 3.2 muestra la interfaz inicial, que permite cargar el volumen deseado para empezar a trabajar con él.



Figura 3.2

La imagen correspondiente a una de las secciones del volumen solicitado aparecerá en pantalla.

El 3D Slicer ofrece un amplio abanico de posibilidades a la hora de visualizar el volumen de datos, permite vistas 2D y 3D, se puede seleccionar que planos se van a mostrar, modificar los colores, etc. Véase Figura 3.3.

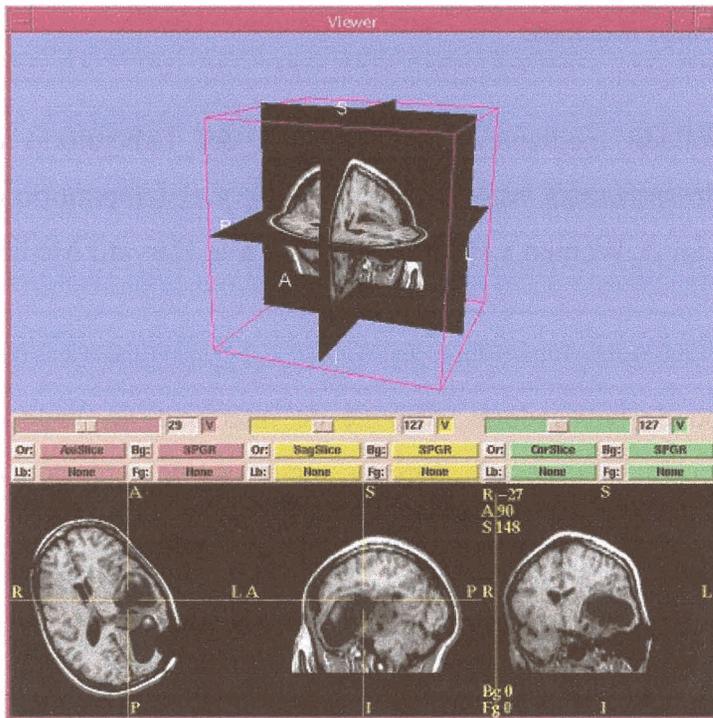


Figura 3.3. 3D Slicer ofrece un gran potencial visual.

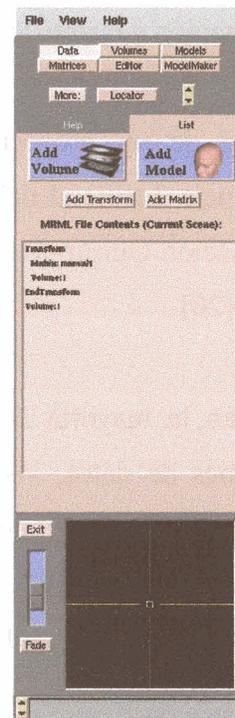


Figura 3.4

El registrado en la interfaz del 3D Slicer es accesible bajo el panel de Matrices. Véase Figura 3.4.

El 3D Slicer realiza el registrado rígido manual de datos tridimensionales, permitiendo al usuario especificar qué volumen de datos desea mover.

El usuario puede de manera intuitiva trasladar y rotar el volumen de datos, simplemente con pulsar y arrastrar con el cursor del ratón sobre las imágenes, en la ventana 2D. El volumen de referencia se muestra como la capa inferior, y el volumen a registrar, se muestra como una capa semitransparente superior. Además, el resultado del registrado manual, puede ser usado como el punto de partida inicial para un registrado automático basado en la maximización de la información mutua.

Analyze 4.0

El Analyze es un paquete software que permite la visualización y análisis de datos biomédicos 3D. Se trata de un producto comercial diseñado y desarrollado en el *Biomedical imaging resource* (BIR) de la Clínica Mayo [19].

El Analyze dispone de distintos procesos (filtrado, segmentación, modelado, etc.), entre esos procesos se encuentra el de registrado, denominado aquí como Fusión de imágenes.

Por medio del modulo de Fusión, volúmenes de datos de distintas modalidades (por ejemplo, PET o MRI) pueden ser *fusionados* en tres dimensiones, permitiendo de este modo que datos estructurales puedan ser registrados de manera rápida y precisa con datos funcionales, tal y como se puede ver en la Figura 3.5.

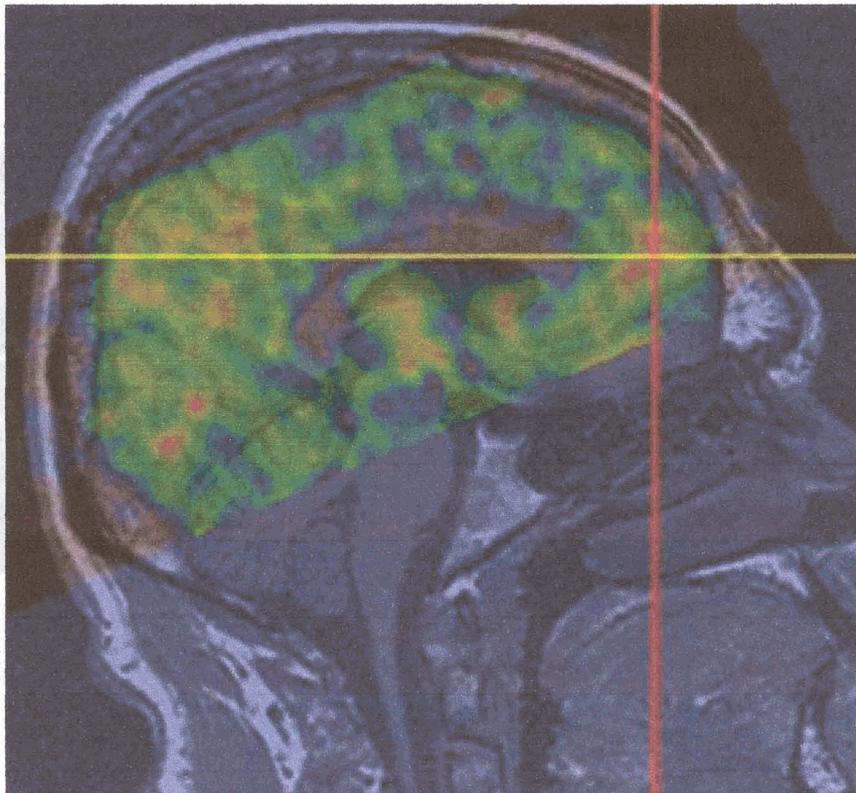


Figura 3.5. Registrado 3D de datos funcionales PET con datos estructurales MRI.

Este modulo puede fusionar imágenes 2D y conjuntos de datos volumétricos 3D, tomados en distintos instantes, o datos de distintas modalidades obtenidos en el mismo instante.

Asimismo Analyze permite el registrado sin la necesidad de segmentar con anterioridad los datos. El usuario simplemente carga el volumen de referencia y el volumen a registrar, selecciona el comando *Surface Match*, y registra los datos. El volumen a registrar es escalado, trasladado y rotado en cuestión de segundos, hasta ajustarse al volumen de referencia.

Entre las distintas opciones de registrado en el Analyze, tenemos:

- Registrado 2D.
- Registrado no rígido 2D.
- Registrado 3D.
- Cubos deformables
- Atlas Cerebral

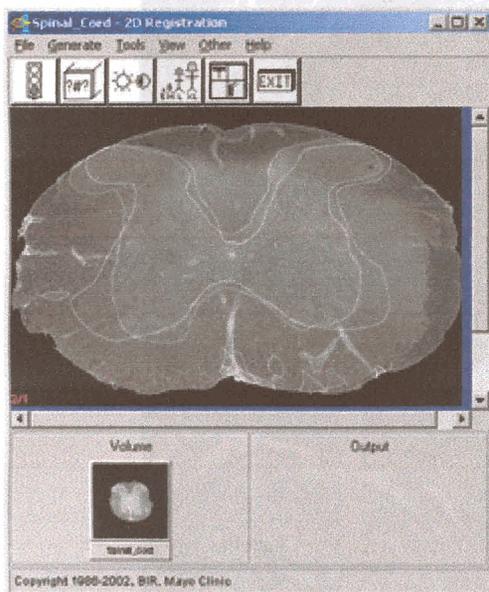


Figura 3.6.

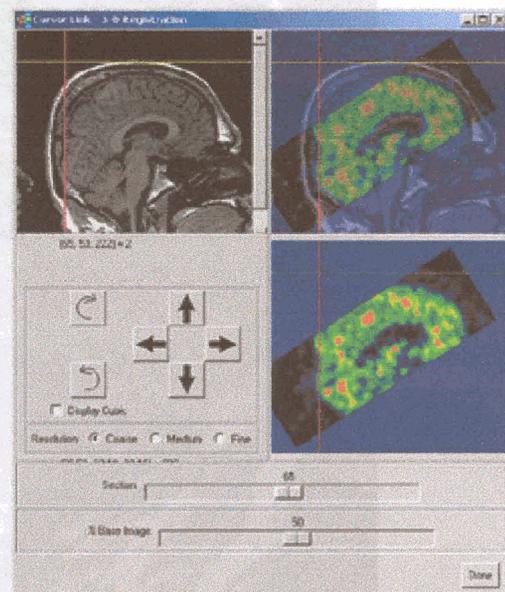


Figura 3.7.

Ventana para el registrado 2D en el Analyze.

Ventana para el registrado 3D en el Analyze.

Amira™ es un sistema software orientado a objetos, modular. Sus componentes básicos son módulos y objetos de datos. Los módulos se usan para visualizar los datos o para realizar operaciones computacionales sobre éstos [21].

Los datos se crean de manera automática al abrir el fichero de datos, o se obtienen como salida del proceso computacional de un módulo. Los módulos correspondientes a un dato existente, se crean como instancias particulares de los distintos tipos de módulos disponibles para ese objeto vía un menú contextual.

En lo que al aspecto gráfico de la interfaz, la ventana de visualización 3D ocupa la mayor parte del área de la pantalla, como se observa en la Figura 3.8. En la versión de desarrollador de Amira, se facilita el uso de extensiones definidas por el usuario.

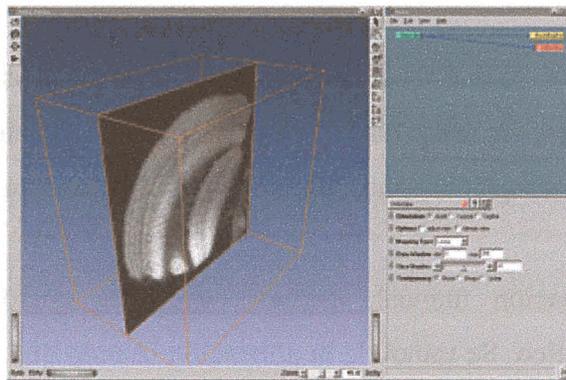


Figura 3.8

Entre las principales características del Amira, cabe destacar las siguientes:

- Visualización de datos 3D de distintas modalidades como CT, MR, etc.
- Segmentación automática e interactivo de estos datos.
- Un editor de transformaciones 3D intuitivo.
- Registrado automático e interactivo de datos 3D.
- Alineamiento de datos 3D.
- Algoritmos avanzados de visualización de campos vectoriales.

El software Amira es especialmente útil, cuando la muestra se analiza por capas, como por ejemplo en técnicas microscópicas en las que la muestra es físicamente cortada en rodajas, y las imágenes se forman a partir de cada una de las rodajas de forma independiente.

Normalmente estas imágenes sufrirán una desalineación relativa entre ellas. Antes de proceder a la reconstrucción del modelo tridimensional de la muestra, las imágenes deberán ser alineadas teniendo en cuenta la posible traslación y rotación.

Esta tarea se puede llevar a cabo por distintos medios:

- *Alineación básica manual.*

Al seleccionar el módulo de alineación básico manual, se lanza una nueva ventana que permite al usuario realizar de manera interactiva el alineamiento de las distintas capas o rodajas de la pila de imágenes que conforman el dato tridimensional, ver Figura 3.9. Para facilitar esta tarea se muestran dos capas consecutivas de manera simultánea. Una de estas capas es editable, i.e. se le puede aplicar una traslación y/ o una rotación con un simple movimiento del cursor del ratón.

En cualquier momento se puede seleccionar qué capa queremos editar, si la superior o la inferior, de las que se muestran. Asimismo es posible seleccionar cuáles de las múltiples capas de la imagen 3D queremos mostrar en esta ventana de alineación.

Aparte de la alineación manual, existen otras tres opciones que permiten el alineamiento automático. Se tratan de la alineación usando una transformación de ejes principales, el alineamiento automático por medio de la optimización de la función de Calidad y el alineamiento a partir de puntos de control definidos por el usuario.

El primero de ellos es idóneo para imágenes en las que existe una clara separación entre el objeto y el fondo de la imagen. El método de optimización requiere que las imágenes estén aproximadamente alineadas. El método de alineación por puntos de control podría proporcionar el pre-alineamiento necesario para efectuar la alineación automática por optimización.

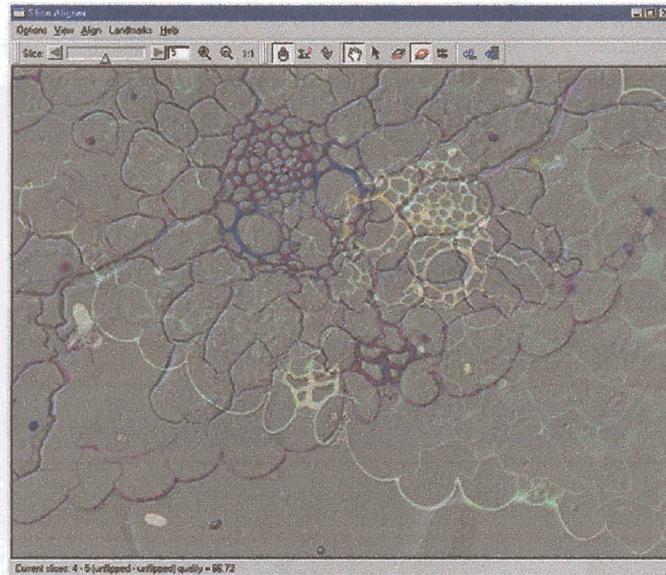


Figura 3.9

- *Alineación a partir de puntos de control.*

Esta alineación requiere en primer lugar la inserción por parte del usuario de una serie de puntos de control o puntos clave (también llamados *landmarks*), esto es posible en el modo de edición de puntos clave. En este modo únicamente se muestra una capa, en vez de las dos del modo de alineamiento manual. Por defecto se definen dos puntos clave en cada capa. El usuario puede seleccionar con una simple pulsación del botón del ratón, donde desea ubicar dichos puntos clave.

Es posible la adición de más puntos clave, así como también es posible eliminar algunos siempre y cuando exista un mínimo de dos puntos clave por capa.

Una vez se han fijado todos los puntos clave, se regresa a la ventana de alineamiento manual, y desde allí se puede proceder al alineamiento automático bien de todas las capas a la vez, bien de las dos capas que se muestran simultáneamente en dicha ventana. Véase Figura 3.10.

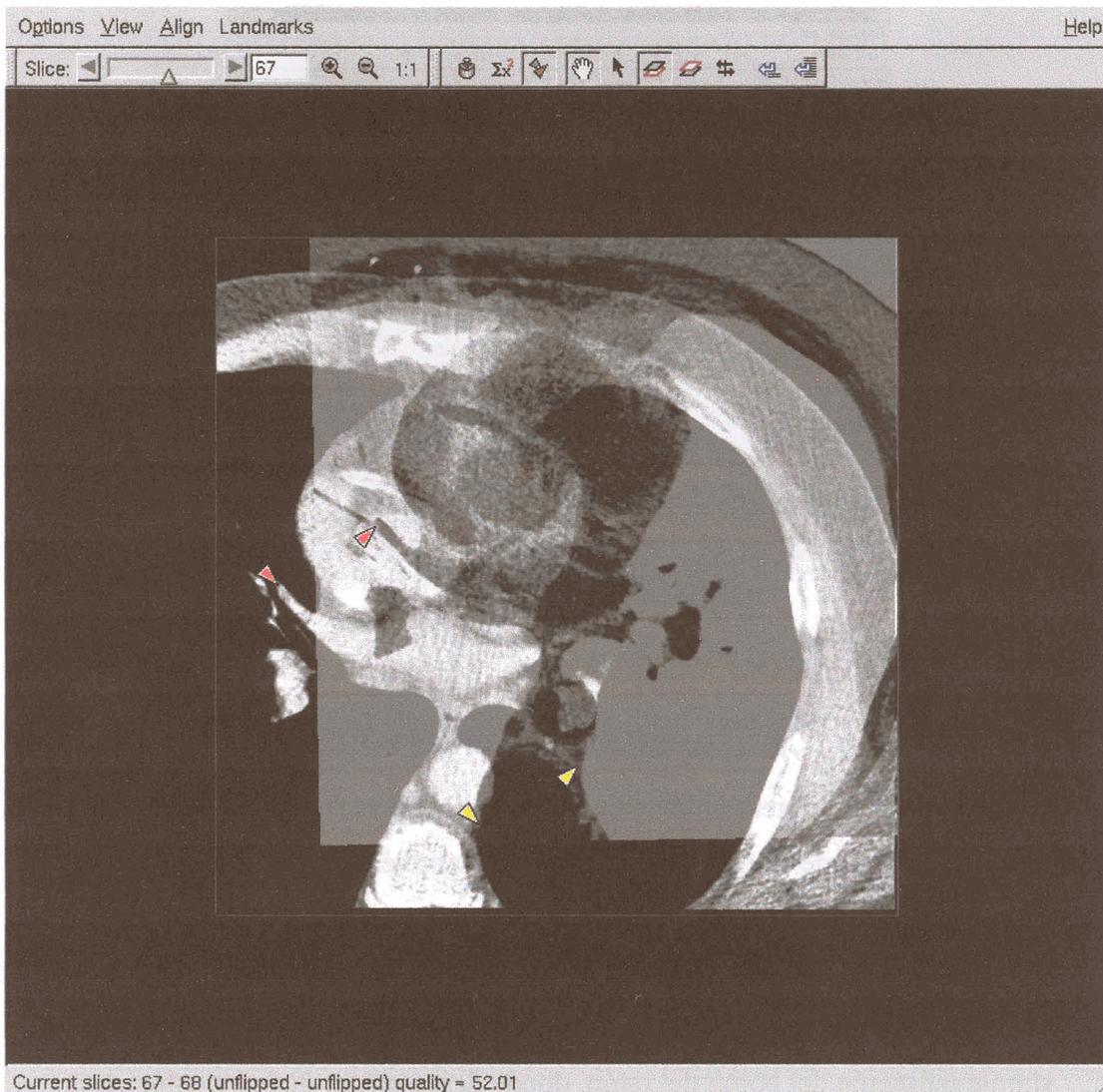


Figura 3.10

- *Optimización de la función de Calidad.*

Una vez que todas las capas han sido alineadas a grosso modo se puede proceder al mejorar dicha alineación por medio del método automático de optimización de la función de Calidad. La función de Calidad se calcula como resultado de la evaluación de la diferencia cuadrática entre los niveles de gris de las dos capas. El método de optimización trata de maximizar dicha función de Calidad.

La alineación automática es un proceso iterativo, y por tanto puede llevar una considerable cantidad de tiempo obtener un resultado, dependiendo de la resolución de las imágenes y de la calidad del pre-alineamiento.

MNI AutoReg

MNI AutoReg es un paquete software para uso exclusivo en investigación, desarrollado en el *Brain Imaging Centre, del Montreal Neurological Institute, McGill University* [23].

Tanto en aplicaciones de diagnóstico como de investigación, la interpretación de las imágenes de resonancia magnética (MR) del cerebro humano es más sencilla, cuando diferentes conjuntos de datos pueden ser comparados por medio de la inspección visual de posiciones anatómicas equivalentes.



Figura 3.11. Aspecto de la interfaz de MNI AutoReg.

El análisis cuantitativo con plantillas de atlas predefinidos generalmente requiere una alineación inicial entre el atlas y cada plano de imagen.

Desafortunadamente, el plano axial de adquisición es generalmente diferente entre las distintas sesiones de exploración, difiriendo tanto en posición relativa como en orientación, de modo que las distintas capas de la imagen tridimensional no son coplanares con sus equivalentes en el Atlas.

El MNI AutoReg utiliza un método completamente automático basado en la correlación cruzada tridimensional multiescala para registrar el conjunto de datos volumétricos con los de un cerebro MRI promedio (promediado a partir de más de 300 cerebros), alineado usando el sistema de coordenadas estéreo-táctico Talairach.

Una vez que el conjunto de datos haya sido remuestreado usando la transformación generada por el algoritmo, las capas del atlas serán sobre impresionadas en las capas correspondientes del volumen remuestreado.

El uso de éste espacio estandarizado permite la comparación directa, vóxel a vóxel, de dos o más conjuntos de datos traídos al espacio estéreo-táctico.

El programa automáticamente encuentra la mejor transformación lineal para mapear un conjunto de datos en el otro.

El programa utiliza la optimización de una serie de parámetros, seleccionables por el usuario para identificar los mejores valores de transformación.

El CISG (*Computational Imaging Science Group*) es un paquete software desarrollado por el departamento ciencias radiológicas del *King's College* de Londres [18]. El CISG proporciona dos algoritmos de registro de imágenes basados en vóxel, fundamentados en la maximización de la información mutua normalizada.

Las principales características de este software son:

- Registrado afin basado en vóxel, que utiliza la información mutua normalizada.
- Registrado no rígido basado en vóxel, fundamentado en la interpolación B-spline de la información mutua normalizada.
- Interfaz gráfica de usuario que permite visualizar pares de imágenes, lanzar el proceso de registro y analizar los resultados.

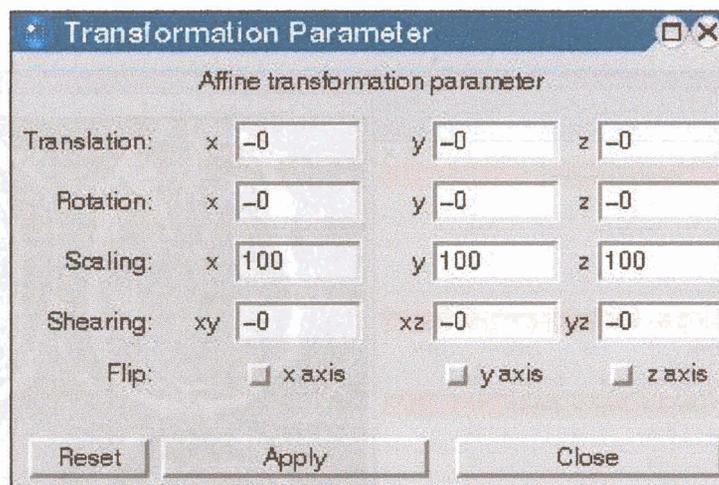


Figura 3.12. Interfaz para editar parámetros de la transformación afin.

La interfaz gráfica de usuario del CISG es muy intuitiva y amigable e incluye herramientas para la visualización de pares de imágenes por separado, superpuestas, etc. que permiten el análisis de los resultados de registro. Tal y como muestran las siguientes figuras.

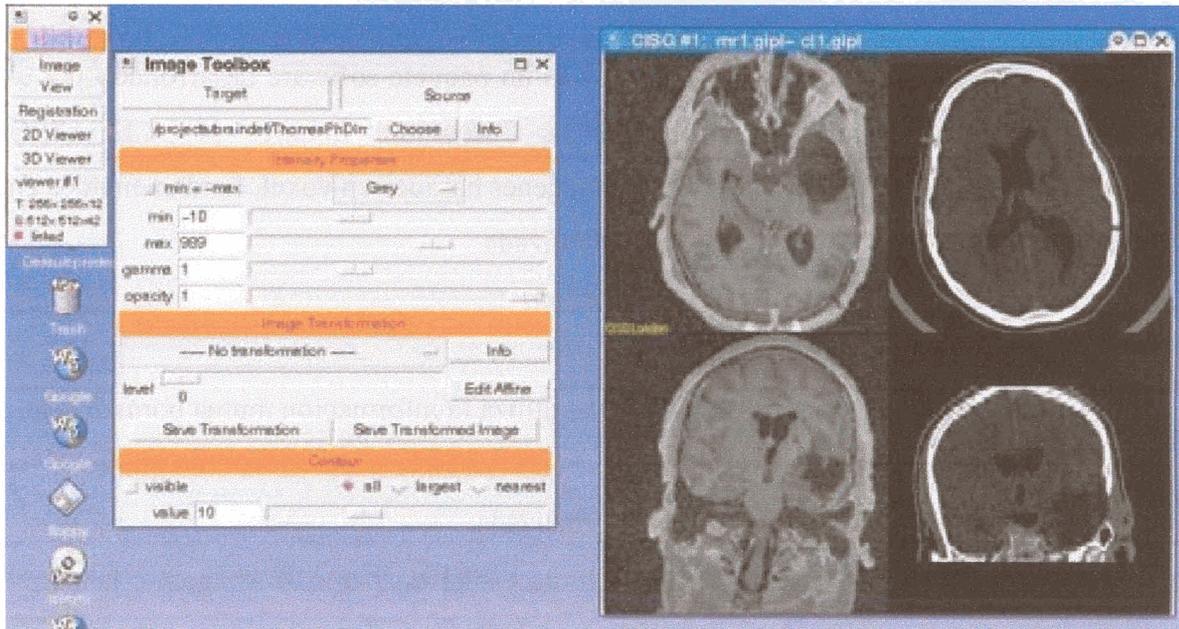


Figura 3.13. En la ventana de visualización se muestran los datos que se van a registrar.

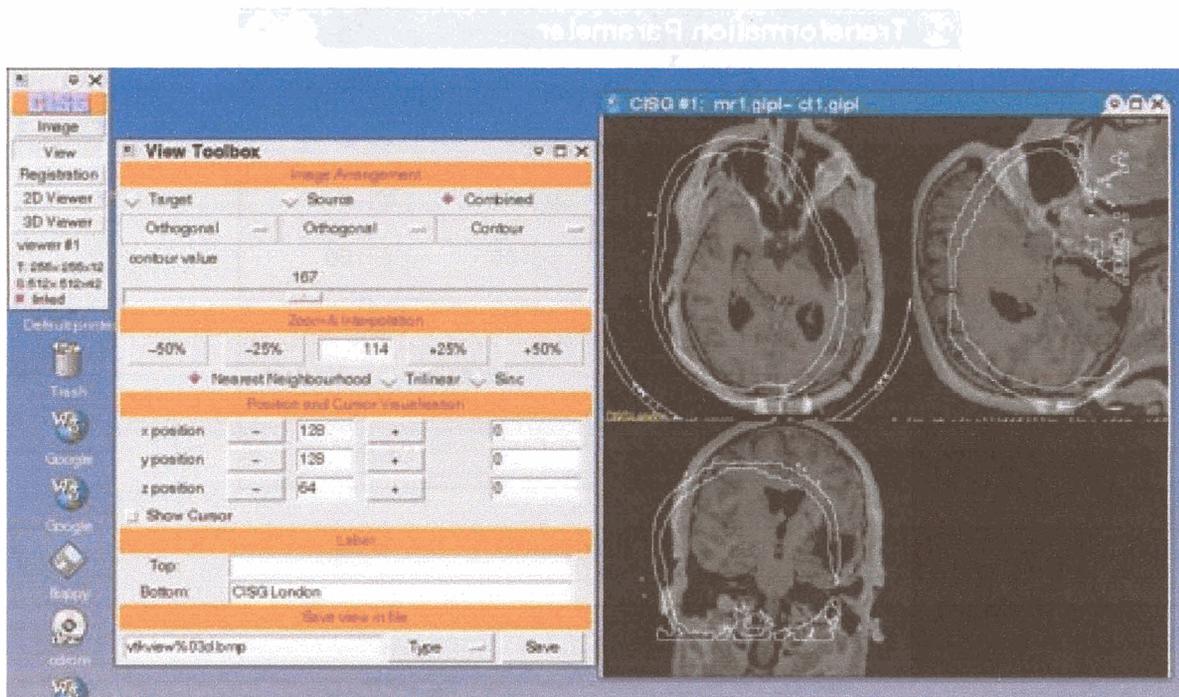


Figura 3.14. También es posible superponer ambas imágenes.



El AIR (*Automated Image Registration*) es un paquete software desarrollado para la investigación, y se encuentra disponible de manera gratuita para la comunidad científica [24].

El paquete de software AIR incluye rutinas automáticas para la alineación de datos tomográficos. El AIR no proporciona capacidades de visualización de imágenes, y se trata de un programa que funciona basándose en comandos y que dispone de una simple interfaz paramétrica, representada en la Figura 3.15.

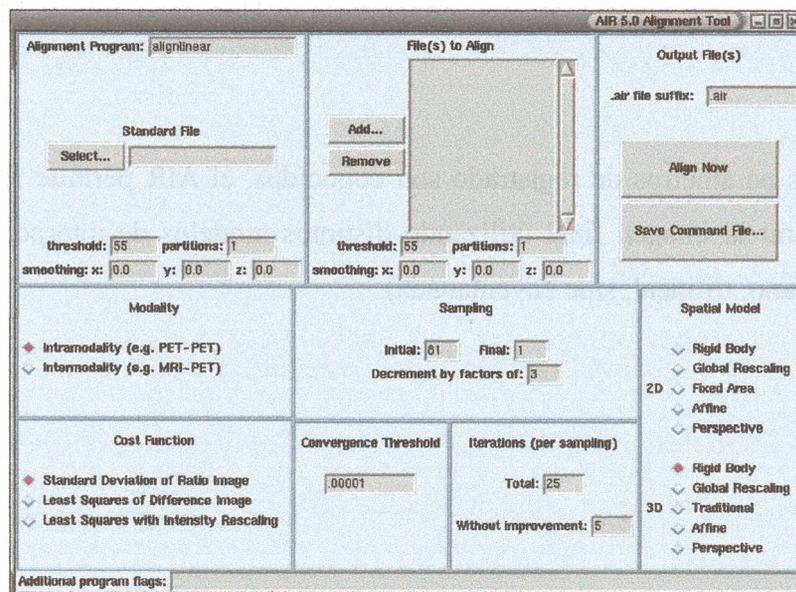


Figura 3.15

AIR permite el registro automático de imágenes 3D y 2D, intra-sujetos e intersujetos y en ocasiones entre distintas modalidades de imagen. Este producto ha sido diseñado para resolver diferentes problemas de registro que surgen en los conjuntos de datos topográficos:

- Intra-sujeto, registro intra-modalidad (e.g. de PET a PET, de MRI a MRI)
- Intra-sujeto, registro intermodalidad (e.g. de PET a MRI)
- Registro intersujeto (e.g. de sujeto a sujeto, o de sujeto a plantilla de Atlas)

El registrado intra-sujeto de imágenes cerebrales utiliza el modelo de cuerpo rígido. El registrado intermodalidad ha sido validado para algunos (MRI - PET), pero no para todas las modalidades [24].

El registrado intersujeto puede ser realizado usando cualquiera de entre la variedad de modelos lineales y no lineales para registrar un sujeto respecto a otro sujeto, o respecto a una plantilla de Atlas (por ejemplo un cerebro promediado en el espacio Talairach). Estos modelos pueden ser útiles para el registrado intra-sujetos de órganos que son más deformables que un cerebro, o para el seguimiento intra-sujeto de cambios a lo largo del tiempo

Aparte de los modelos 3D, AIR incluye modelos de deformación 2D homólogos que pueden ser útiles en circunstancias muy concretas(e.g. Una capa simple MRI, imágenes 2D, etc.).

Una vez que los parámetros de registrado son conocidos, el AIR permite remuestrear los datos para generar la imagen final utilizando distintos modelos de interpolación. (e.g. el vecino más cercano, trilineal, sinc enventanada).

El *Statistical Parametric Mapping*, es un conjunto de funciones y subrutinas implementadas bajo el entorno de programación Matlab®, que han sido desarrolladas por el *Methodology Group* del *Wellcome Department of Cognitive Neurology*, con el fin de interpretar y organizar datos de neuroimágenes funcionales [22].

Las principales características de este software son:

- Realineamiento de secuencias de imágenes.
- Normalización espacial no lineal automática.

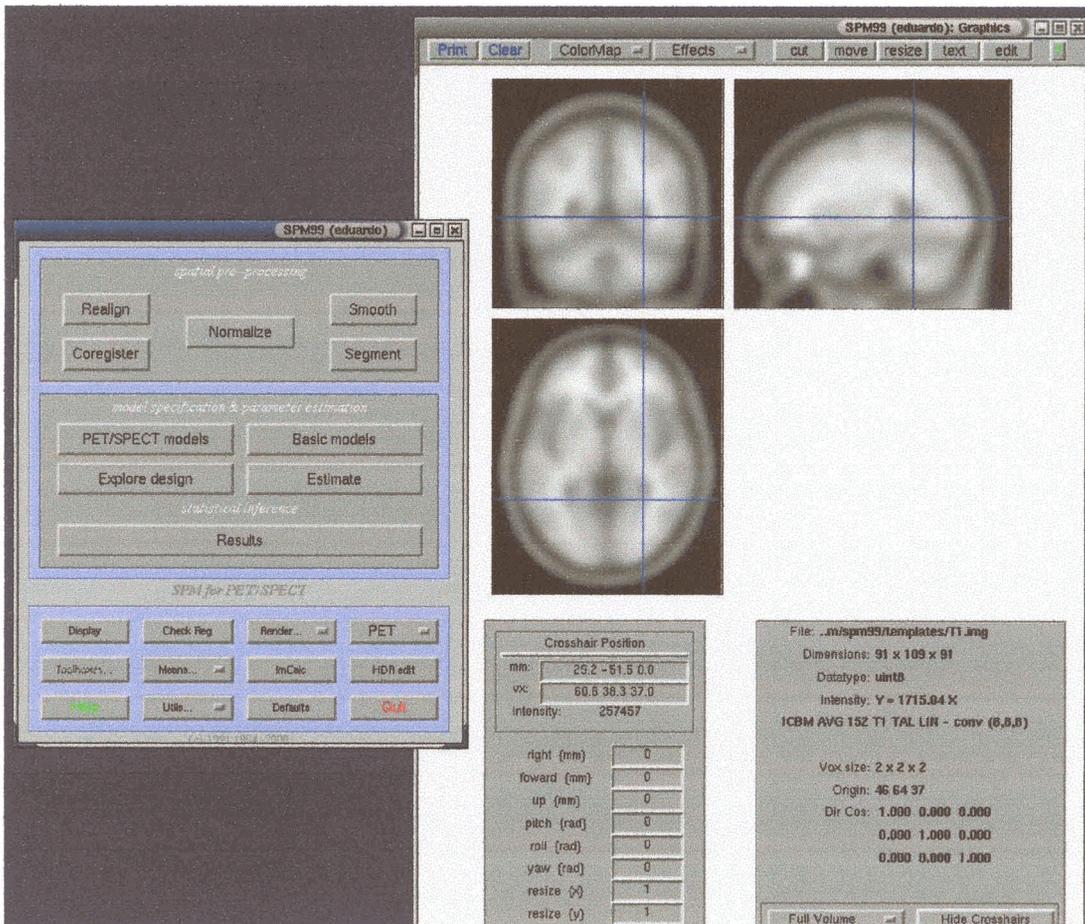


Figura 3.16. Aspecto de la interfaz del SPM.

El SPM está basado en vóxel.

- Las imágenes son normalizadas espacialmente y suavizadas.
- Modelos paramétricos estadísticos se asumen para cada vóxel, usando el modelo lineal general para describir la variabilidad en los datos, en términos experimentales y variabilidad residual. Las hipótesis expresadas en términos del modelo paramétrico son evaluadas para cada vóxel con estadísticos invariantes.

Esto proporciona una imagen cuyos valores de vóxel son estadísticos, por así decirlo una *Imagen Estadística* o *Mapa Paramétrico Estadístico*. La convolución temporal del modelo lineal general para las imágenes MRI permite la aplicación de los resultados a partir de una regresión correlada serial, permitiendo la construcción de imágenes estadísticas a partir de series temporales MRI.

4. REGISTRADO

4.1. Introducción

Dentro del ámbito clínico actual, la imagen biomédica es un componente vital para un gran número de aplicaciones. Estas aplicaciones no se refieren únicamente al ámbito del diagnóstico, también afectan al área de la planificación, realización y evaluación de procedimientos quirúrgicos y radioterapéuticos.

Las modalidades de imágenes biomédicas que se emplean se pueden englobar en dos categorías, **anatómicas** y **funcionales**. Las modalidades anatómicas representan primordialmente morfología, y comprenden rayos X, tomografía computerizada (CT), resonancia magnética (MR), ultrasonidos(US), entre otras.

Las modalidades funcionales representan principalmente información del metabolismo de la anatomía subyacente y comprenden entre otras la tomografía por emisión de positrones (PET), la tomografía computerizada por emisión única de fotones (SPECT o SPET *Single photon emission computed tomography*), y la resonancia magnética funcional (fMRI). Las dos primeras constituyen básicamente el grueso de las modalidades de imágenes de la medicina nuclear.

4.2. Clasificación de los métodos de registrado.

Para mejorar nuestro conocimiento del registrado de imágenes médicas, incluimos aquí un conjunto de criterios que nos permiten clasificar las diferentes maneras en las que se puede llevar a cabo el registrado.

El proceso de registrado siempre puede ser descompuesto en tres puntos básicos, el *enunciado* del problema, el *paradigma de registrado*, y el *procedimiento de optimización*. El enunciado del problema, la elección del paradigma y del procedimiento de optimización proporciona una única clasificación de acuerdo a los nueve criterios anteriormente citados [10].

La siguiente clasificación utiliza una serie de criterios básicos que a su vez, en algunos casos se pueden subdividir en uno o varios niveles.

- 1) Dimensionalidad
- 2) Naturaleza de las bases del registrado
 - a. Extrínseca
 - b. Intrínseca
 - c. No basada en imágenes
- 3) Naturaleza de la transformación
 - a. Rígida
 - b. Afín
 - c. Proyectiva
 - d. Curva
- 4) Dominio de la transformación
 - a) Local
 - b) Global
- 5) Interacción
 - a) Interactiva
 - Inicializada
 - Sin inicializar
 - b) Semiautomática
 - Inicializada por el usuario
 - Dirigida / corregida por el usuario
 - Ambas
 - c) Automática
- 6) Procedimiento de optimización
- 7) Modalidades involucradas
 - a. Monomodal
 - b. Multimodal
 - c. Modalidad hacia modelo
 - d. Paciente hacia modalidad
- 8) Sujeto
 - a. Intrasujeto
 - b. Intersujeto
 - c. Atlas

4.3. Dimensionalidad

La principal división que hemos de hacer es determinar si todas las dimensiones presentes son espaciales o si el tiempo es una dimensión añadida. En ambos casos el problema puede ser categorizado en función de las dimensiones involucradas.

- Únicamente dimensiones espaciales:
 - 2D/2D
 - 2D/3D
 - 3D/3D
- Series temporales(más de dos imágenes), con dimensiones espaciales:
 - 2D/2D
 - 2D/3D
 - 3D/3D

4.3.1. Métodos de registrado espacial.

El registrado 3D/3D normalmente aparece como el registrado de dos conjuntos de datos tomográficos.

El registrado 2D/3D puede verse en el alineamiento directo de datos espaciales hacia datos proyectivos (e.g., alineamiento de una tomografía computerizada preoperatoria con una imagen de rayos X intraoperatoria), o en el alineamiento de una capa o rodaja tomográfica con datos espaciales.

El registrado 2D/2D se puede aplicar por ejemplo, a capas separadas de un conjunto de datos tomográficos.

En comparación con el registrado 3D/3D, el registrado 2D/2D es menos complejo en un orden de magnitud, tanto en el número de parámetros (traslación, rotación y factor de escala para cada dimensión), como en lo que al volumen de los datos se refiere, de modo que realizar el registrado es generalmente más simple y rápido que en el caso de registrado 3D/3D.

4.3.2. Registrado de series temporales.

La adquisición de series temporales de imágenes se puede dar en varios casos, como pueden ser el de observar la evolución del crecimiento de los huesos en un niño (intervalo de tiempo largo), controlar el crecimiento de un tumor (intervalo de tiempo medio), o la verificación postoperatorio de la efectividad de una intervención (periodo de tiempo corto). Para poder comparar dos imágenes será necesario realizar una operación de registrado.

4.4. Naturaleza de las bases del registrado

El registrado basado en imágenes puede ser dividido en métodos *extrínsecos*, i.e. basados en objetos externos introducidos en el espacio de la imagen, y métodos *intrínsecos*, i.e. basados en la información de la imagen tal y como es captada del paciente.

De ésta forma, y atendiendo a la naturaleza de las bases del registrado podemos hacer la siguiente clasificación:

- Extrínseca
 - Invasiva
 - Marco Estereotáctico
 - Fiduciales (marcadores atornillados)
 - No invasiva
 - Moldura, marco, adaptador dental, etc.
 - Fiduciales (marcadores de piel)
- Intrínseca
 - Basada en puntos clave o *landmarks*
 - Anatómica
 - Geométrica
 - Basada en la segmentación
 - Modelos rígidos (puntos, curvas, superficies)
 - Modelos deformables (redes, *snakes*)
 - Basada en las propiedades del vóxel
 - Reducción a escalares / vectores (momentos, ejes principales)
 - Uso del contenido completo de la imagen
- No basada en imágenes

4.4.1. Métodos de registrado extrínsecos

Los métodos de naturaleza extrínseca se basan en adosar objetos artificiales al paciente. Objetos que han sido diseñados para ser detectados de manera precisa por el sistema captador en las modalidades pertinentes.

De este modo, el registrado de las imágenes adquiridas es relativamente sencillo y rápido, generalmente se puede automatizar y dado que los parámetros de registrado pueden ser calculados de manera explícita, no hay necesidad de complejos algoritmos de optimización.

El principal inconveniente de los métodos de registrado extrínseco es el carácter invasivo de los marcadores. También se usan marcadores no invasivos pero por norma son menos precisos. Un objeto fiducial de uso normal es el marco estereotáctico, que firmemente atornillado al cráneo del paciente, posibilita un registrado de gran precisión. Semejantes marcos se usan para propósitos de localización y guiado en operaciones de neurocirugía.

Como la neurocirugía es uno de los principales campos de aplicación del registrado y dado su ya de por sí carácter invasivo, el uso del marco estereotáctico no supone grave problema para el paciente.

Sin embargo, el montaje de un marco estereotáctico con el único propósito de hacer un registrado preciso es inadmisibles.

Por lo tanto, generalmente se utilizan marcadores no invasivos, entre los más populares tenemos marcadores que son adheridos a la piel, molduras de espuma o adaptadores dentales, entre otros.

Dado que los métodos extrínsecos por definición no incluyen la información relativa al paciente, la naturaleza de la transformación está restringida a ser rígida (únicamente traslaciones y rotaciones). Debido a ésta restricción, así como otras consideraciones de carácter técnico el uso de métodos extrínsecos 3D/3D se reduce en gran medida a las imágenes cerebrales y ortopédicas, aunque se pueden usar marcadores en imágenes proyectivas de cualquier área del cuerpo.

En algunos casos es posible el uso de marcadores en transformaciones no rígidas, e.g. estudios del latido del corazón de un animal, en el que se fijan los marcadores a las paredes cardiacas.

4.4.2. Métodos de registrado intrínsecos

Los métodos intrínsecos se basan únicamente en el contenido de la imagen del paciente.

Existen distintos métodos, algunos utilizan puntos salientes o significativos para hacer el registrado, es el registrado basado en puntos de control o *landmarks*, en otros métodos el registrado se realiza basándose en el alineamiento de estructuras binarias producto de un proceso previo de segmentación, son los llamados métodos basados en la segmentación.

Por ultimo están aquellos que se basan directamente en los valores de gris de los vóxeles, son los denominados métodos basados en las propiedades del vóxel.

4.4.2.1. Métodos basados en puntos de control o landmarks

Los puntos de control o *landmarks* pueden ser de tipo **anatómico** (i.e. puntos prominentes, fácilmente localizables de la morfología de la anatomía visible), generalmente identificados por el propio usuario; o de tipo **geométrico** (i.e. puntos en el emplazamiento de un máximo de una propiedad geométrica), generalmente localizados de un modo automático e.g., extremos de una curva, esquinas, etc.

Técnicamente la identificación de puntos de control se puede considerar como un proceso de segmentación, pero reservamos la denominación de métodos basados en la segmentación para referirnos a estructuras de orden superior tales como curvas, superficies, y volúmenes. El registrado basado en puntos de control es versátil en el sentido en que, al menos en teoría, puede aplicarse a todo tipo de imágenes independientemente de su tipo o finalidad.

Los métodos de registrado basados en puntos de control se utilizan principalmente para obtener transformaciones rígidas o afines. Si el conjunto de puntos es lo suficientemente amplio, se podrían utilizar, al menos en teoría, para calcular transformaciones más complejas.

En ocasiones se utiliza el registrado basado en puntos de control en combinación con métodos de registrado con una base completamente distinta. Así los métodos basados en la optimización de un parámetro espacial que no es cuasi-convexo son propensos a atascarse en un máximo local, pudiendo dar lugar a un mal registrado. Restringiendo el espacio de búsqueda de acuerdo con los puntos de control anatómicos, semejantes errores se pueden evitar, acelerando además los cálculos.

El principal inconveniente de los métodos basados en puntos de control radica en la necesaria intervención del usuario a la hora de identificar los puntos de control.

En el registrado basado en puntos de control, el conjunto de puntos es escaso en comparación con el contenido original de la imagen, lo que propicia algoritmos de optimización relativamente rápidos. Estos algoritmos optimizan medidas tales como la distancia entre cada punto de control y su vecino más cercano (métrica procustea), o como las distancias mínimas iteradas entre puntos clave, entre otras. Para la optimización de esta última, son muy populares el uso del algoritmo ICP *iterative closest point* y algoritmos derivados.

4.4.2.2. *Métodos basados en la segmentación*

Dentro del amplio campo del registrado, la opción más sencilla es la del registrado de datos ya segmentados.

Los datos segmentados están siempre relacionados con una adquisición de datos médicos común en escala de grises. En este mismo sistema de referencia, y de un tamaño igual al de los datos en escala de grises, identificamos aquellos elementos de la imagen (píxeles) pertenecientes a un órgano o estructura anatómica determinada.

Por tanto, previo paso por un especialista que se ha encargado de hacer la segmentación, es posible tomar dos órganos segmentados en diferentes imágenes de diferentes sistemas de referencia, y buscar la transformación afin que nos relaciona ambas imágenes etiquetadas. Esta transformación sería la misma que nos llevaría de un conjunto de datos en escala de grises a otro semejante. El registrado se lleva a cabo simplemente hallando el centro de masas de cada una de las etiquetas y haciendo el desplazamiento correspondiente.

Los métodos basados en la segmentación pueden estar basados en un modelo rígido, donde las mismas estructuras anatómicas (principalmente superficies) son extraídas de las dos imágenes a registrar, y son usadas como únicos datos de entrada para el algoritmo de alineado. Asimismo, pueden estar basados en un modelo deformable, en este caso la estructura extraída de una imagen es deformada elásticamente hasta encajar con la de la otra imagen.

El enfoque del modelo rígido es de uso frecuente en el ámbito clínico, esto se debe en parte al éxito del método *head-hat*, fundamentado en la segmentación que de la superficie de la piel se hace en las imágenes CT, MR y PET de la cabeza. Dado que la tarea de segmentación es relativamente sencilla de realizar y la complejidad computacional, relativamente baja, el método goza de gran popularidad e incluso se plantea el automatizar el proceso de segmentación. Otra causa que justifica el uso del modelo rígido es la rapidez de la técnica de alineamiento de estructuras binarias por medio de una transformación de distancia denominada *Chamfer matching*.

Un inconveniente de los métodos basados en la segmentación es que la precisión del registrado se ve condicionada por la precisión del proceso de segmentación.

En teoría el registrado basado en la segmentación es aplicable a diferentes áreas del cuerpo, pero en la práctica su uso se ha limitado a la imagen cerebral y la imagen ortopédica. El proceso de segmentación se realiza la mayoría de las veces de manera semiautomática.

En cambio, con el modelo deformable, el criterio de optimización es diferente. Siempre se define y calcula de manera local, y la deformación está restringida por los límites del modelo elástico impuesto a la curva o superficie segmentada. Dichas curvas deformables se denominan en la literatura especializada como contornos activos o *snakes*, de igual manera los modelos deformables 3D se denominan mallas o redes.

Para facilitar el modelado físico, la estructura de datos del modelo deformable se representa utilizando funciones localizadas con curvas parametrizadas o *splines*, en vez de por un conjunto de puntos. El proceso de deformación se lleva a cabo de manera iterativa a partir de pequeñas deformaciones. El enfoque del modelo deformable requiere la definición de una plantilla en una de las imágenes.

Posteriormente se pueden seguir dos estrategias, deformar la plantilla hasta hacerla encajar con la estructura segmentada de la segunda imagen, o bien hacerla encajar con la segunda imagen sin segmentar, en éste último caso el criterio debe ser utilizar el borde de la región para encajar la malla o contorno activo.

Al contrario que el registrado basado en modelos rígidos, más conveniente para el registrado intrasujeto, el registrado basado en modelos deformables resulta más apropiado para un registrado intersujeto, de Atlas, así como para el registrado de una plantilla obtenida de un paciente, con el modelo matemático de una plantilla anatómica general.

Un inconveniente de los modelos deformables es que generalmente requieren una buena posición inicial para una correcta convergencia. Otro inconveniente es la naturaleza imprevisiblemente errática de las deformaciones a nivel local, de modo que es frecuente que el modelo deformable se adapte a la anatomía de una manera perfecta, excepto en la parte más interesante de la imagen, donde por ejemplo aparece un gran desarrollo de un tumor. La solución a este problema pasa por adaptar la restricciones locales de elasticidad.

Los modelos deformables son más recomendables para hallar transformaciones curvas locales entre imágenes, y no tanto para hallar transformaciones rígidas o afines. Se pueden utilizar para prácticamente cualquier modalidad o área anatómica, y es generalmente automatizada, excepto el proceso de segmentación.

Una de las aplicaciones más utilizadas hasta el momento son el registrado del contorno de los huesos obtenidos por una imagen CT, dado que su contorno se extrae fácilmente al no haber ningún otro que perturbe la correcta convergencia. Otro uso frecuente se da en el registrado cortical de imágenes MR, la importancia en éste caso radica en que si es posible encontrar el registrado cortical entre dos cerebros, la segmentación de la corteza cerebral de uno, puede ser transferida de manera instantánea al otro.

4.4.2.3. Métodos basados en las propiedades del vóxel

Los métodos de registrado *basados en las propiedades del vóxel*, se diferencian de los otros métodos *intrínsecos* por el hecho de operar directamente sobre los valores de gris de la imagen, sin una previa reducción de datos por parte del usuario ni segmentación.

Existen dos estrategias diferentes, una consiste en *reducir* el contenido de grises de la imagen a un conjunto representativo de escalares y orientaciones, la otra, consiste en utilizar todo el contenido de la imagen a lo largo del todo el proceso de registrado.

Métodos *basados en los ejes principales o en momentos* son los ejemplos más significativos de los métodos de registrado reductivos. En ellos, el centro de gravedad de la imagen así como los ejes principales de orientación de la imagen son calculados a partir de los momentos de cero y primer orden. El registrado se hace pues, alineando el centro de gravedad y los ejes principales. Este método puede no ser muy preciso, pero su uso está muy extendido en aplicaciones sin grandes exigencias en cuanto a la precisión debido a la facilidad de implementación, rapidez y naturaleza automática.

Los métodos que utilizan todo el contenido de la imagen son de los más interesantes de los métodos descubiertos recientemente. Teóricamente se tratan de los métodos más flexibles dado que no existe una reducción previa de la información de la imagen, sino que se usa todo el contenido a lo largo de todo el proceso de registrado. A pesar de que estos métodos han sido usados desde hace tiempo, su uso intensivo en aplicaciones clínicas 3D/3D ha estado extraordinariamente limitado por el considerable costo computacional. Sólo la necesidad de un registrado más preciso y retroactivo, junto con el desarrollo de computadoras más potentes con mayores memorias internas ha permitido su uso en la practica clínica, a pesar de esto, aún no se ha introducido su uso en aplicaciones donde el tiempo de registrado es un factor clave, como el registrado 2D/3D intraoperativo.

4.4.2.4. Métodos no basados en imágenes

Resulta paradójico que el registrado *multimodal* de imágenes pueda ser no basado en imágenes, pero esto es posible si los sistema de coordenadas de los dos escáneres involucrados, están de alguna manera calibrados entre sí. Esto requeriría en principio que ambos escáneres estuviesen en la misma ubicación física y la suposición de que el paciente permanece absolutamente inmóvil entre ambas adquisiciones. Estos prerrequisitos son prohibitivos en casi todas las aplicaciones, pero se cumplen razonadamente en aplicaciones que incluyan el uso de ultrasonidos. Dado que los sistemas de ultrasonido pueden usarse como instrumentos manuales, y si están dotados de un sistema de localización espacial que permite una sencilla calibración, resulta adecuado su uso mientras el paciente está inmovilizado durante la adquisición del CT o MR .

4.5. Naturaleza de la transformación

Se denomina transformación **rígida** de una imagen, a aquella en la que únicamente traslaciones y rotaciones están permitidas. Si la transformación delinea las líneas paralelas como líneas paralelas, la transformación se llama **afín**. Si delinea las líneas como líneas, se denomina **proyectiva**. Si delinea las líneas como curvas, entonces se la conoce como transformación **curva** o **elástica**.

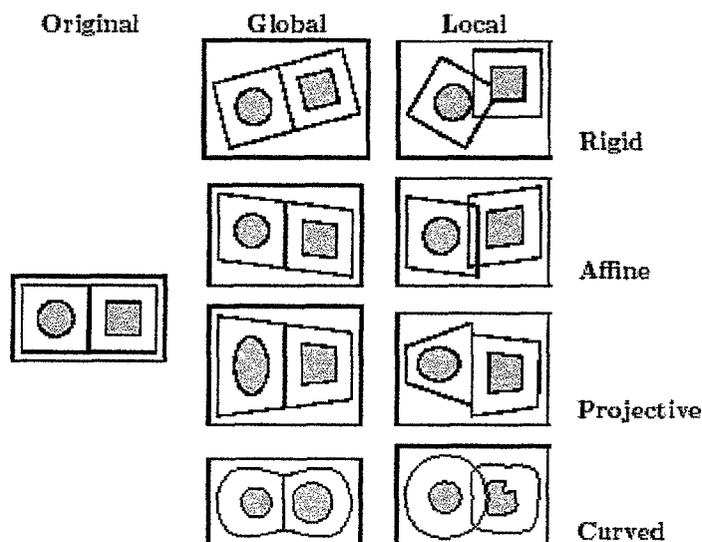


Ilustración 4.1. Ejemplos de tipos de transformaciones en 2D.

Cada tipo de transformación contiene como casos específicos los definidos anteriormente, así la transformación rígida es un caso especial de transformación afín. De igual manera la composición de más de una transformación, se puede categorizar como una transformación simple del tipo más complejo de la composición, e.g., la composición de una transformación proyectiva y una afín es una transformación proyectiva.

Una transformación 3D rígida o afín, puede describirse usando una simple ecuación lineal matricial:

$$y_i = a_{ij}x_j$$

Donde x e y son el antiguo y nuevo vector de coordenadas respectivamente.

Para el caso de una transformación rígida, la ecuación esta restringida a ser del tipo:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \left(\begin{array}{ccc|c} & & & t \\ & r & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

Donde t es un vector de traslación arbitrario, y r es una matriz 3x3 de rotación definida tal que:

$$r_{ij} = r_{ij}^{(1)} r_{jk}^{(2)} r_{kl}^{(3)}$$

$$r^{(2)} = \begin{pmatrix} \cos \alpha_2 & 0 & \sin \alpha_2 \\ 0 & 1 & 0 \\ -\sin \alpha_2 & 0 & \cos \alpha_2 \end{pmatrix}$$

$$r^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_1 & -\sin \alpha_1 \\ 0 & \sin \alpha_1 & \cos \alpha_1 \end{pmatrix}$$

$$r^{(3)} = \begin{pmatrix} \cos \alpha_3 & -\sin \alpha_3 & 0 \\ \sin \alpha_3 & \cos \alpha_3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

i.e., $r^{(1)}$ rota la imagen un ángulo α_i alrededor del eje i .

En el caso de la transformación afin, r no tiene restricciones.

En el caso de transformación proyectiva, únicamente puede ser caracterizada por una ecuación matricial si utilizamos coordenadas homogéneas:

$$y_i = \frac{u_i}{u_4}; \quad u_i = \frac{a_{ij}}{x_j}$$

donde a es una matriz arbitraria de 4x4.

En general, las transformaciones curvas no se pueden representar utilizando la ecuación matricial. La mayoría se representan en términos de un campo de desplazamiento locales de vectores:

$$y_i = x_i + t_i(x)$$

o como transformaciones polinómicas en términos de las antiguas coordenadas.

4.6. Dominio de la transformación

Una transformación se considera **global** si se aplica a la totalidad de la imagen, y **local** si distintas subsecciones (al menos dos) de la imagen tienen definidas sus propias transformaciones, y éstas no pueden ser descritas como una única transformación global.

Las transformaciones locales se utilizan rara vez, porque quebrantarían la continuidad local y la biyectividad de la transformación, que dañaría de manera directa el proceso de remuestreo de la imagen al aplicarle la transformación.

Según la literatura especializada, por norma, las transformaciones rígidas y afines son globales y las transformaciones curvas son locales. Esto tiene mucho sentido dado el modelo físico subyacente de las transformaciones de tipo curvo, y teniendo en cuenta que la restricción del registrado de cuerpo rígido se satisface de manera global en la gran mayoría de las imágenes médicas.

Las transformaciones afines son utilizadas típicamente en casos de movimiento del cuerpo rígido en los que el factor de escala se desconoce o se sospecha que es erróneo (e.g., en imágenes MR debido a las distorsiones geométricas). La transformación de tipo proyectiva no tiene una base física real en el registrado de imágenes excepto en el caso del registrado 2D/3D. Dado que la información local de la anatomía es esencial para proporcionar una transformación local curva precisa, sus aplicaciones son casi siempre intrínsecas, generalmente basada en el modelo deformable o en métodos que utilizan todo el contenido de la imagen, y casi siempre semiautomática necesitando únicamente de una inicialización definida por el usuario. Su uso se limita casi exclusivamente a imágenes anatómicas (CT, MR) de la cabeza, y están especialmente recomendadas para registrado intersujeto y registrado de imagen hacia Atlas. Muchos métodos necesitan de un prerregistrado usando una transformación rígida o afín.

La transformación rígida global es la más usada en las aplicaciones de registrado, esto se deba a que en la mayoría de las imágenes médicas, la restricción del cuerpo rígido se considera cierta, al menos como una buena aproximación. Además, solamente es necesario determinar un reducido número de parámetros, y muchas técnicas de registrado no están habilitadas para proporcionar una transformación más compleja. La cabeza humana es el área de aplicación más común para este registrado.

4.7. Interacción

En lo que a los algoritmos de registrado se refiere, se reconocen tres niveles de interacción. *Automático*, donde el usuario únicamente proporciona al algoritmo las imágenes y la posible información de la adquisición de la imagen. *Interactivo*, donde el usuario realiza el registrado por sí mismo, asistido por el software, proporcionándole éste una referencia visual o numérica de la transformación actual, y posiblemente un pronóstico de la transformación inicial. *Semiautomática*, donde la interacción puede ser de dos tipos:

El usuario necesita inicializar el algoritmo, e.g., segmentando los datos. O el usuario guía al algoritmo, e.g. rechazando o aceptando las distintas hipótesis de registrado sugeridas por el programa.

Los métodos *extrínsecos* son generalmente *automáticos*, dado que los marcadores son diseñados para ser bien visibles y detectables en las imágenes involucradas. En ocasiones el usuario ha de seleccionar grosso modo el punto donde se localiza el marcador siendo entonces un proceso *semiautomático*. De los métodos *intrínsecos*, el de *puntos de control anatómicos* y los métodos *basados en la segmentación*, son generalmente *semiautomáticos* (inicializados por el usuario), y los de *puntos de control geométrico* y *basados en las propiedades del vóxel* suelen ser *automáticos*.

4.8. Procedimiento de optimización

Los parámetros que constituyen la transformación del registrado pueden ser bien *calculados* directamente, i.e., determinados de manera explícita a partir de los datos disponibles, o bien *cercados* i.e., determinados encontrando el óptimo de alguna función definida en el espacio de los parámetros.

Esta función trata de cuantificar la similitud tal y como estaba definida en el paradigma, entre dos imágenes dada una cierta transformación. Estas funciones son generalmente menos complejas en las aplicaciones de registrado monomodal, dado que la similitud se define de modo más directo. No es descabellado suponer que la función de similitud es casi convexa, de modo que se pueden usar algunas de las más conocidas y documentadas técnicas estándar. Algunas de estas técnicas son:

- Método de Powell
- Método Downhill simple
- Método de Brent y series de búsqueda unidimensionales
- Optimización Levenberg-Marquardt
- Iteración Newton-Raphson
- Métodos de búsqueda estocástica
- Métodos de decrecimiento de gradiente
- Métodos genéticos
- *Simulated annealing*
- *Hashing* geométrico
- Métodos de búsqueda casi-exhaustiva

De manera suplementaria, se emplean frecuentemente metodologías multi-resolución (e.g., pirámide) y multi-escala para acelerar el proceso de convergencia, reducir el número de transformaciones a examinar (algo muy importante en los métodos casi-exhaustivos) y evitar mínimos locales. Algunos métodos de registro emplean métodos de optimización no estándar, que son diseñados específicamente para la función de similitud en cuestión, como el algoritmo ICP, que fue creado para el registro basado en *modelo rígido*. Muchas aplicaciones utilizan más de una técnica de optimización, frecuentemente una técnica rápida aunque algo tosca, seguida de otra técnica más precisa aunque más lenta.

4.9. Modalidades involucradas

Se distinguen cuatro clases de tareas de registro en función de las distintas modalidades involucradas. En aplicaciones *monomodales*, las imágenes a registrar pertenecen a la misma modalidad, al contrario que en las tareas de registro *multimodal*, donde las imágenes a registrar proceden de dos modalidades distintas.

En el registro *modalidad hacia modelo* y *paciente hacia modalidad* únicamente está involucrada una imagen, siendo la otra “modalidad” un modelo (patrón) o el propio paciente. El registro *paciente hacia modalidad* aparece casi exclusivamente en aplicaciones intraoperativas y de radioterapia. El registro *modalidad hacia modelo* se utiliza en la recopilación de estadísticas de la morfología de tejidos (e.g., para encontrar anomalías relativas a estructuras normalizadas), y en tareas de segmentación.

El registrado *monomodal* es especialmente apropiado para el seguimiento del desarrollo de tejidos, verificación de intervención, sustracción de imágenes, entre otras. Las aplicaciones del registrado *multimodal* son diversas y abundantes, predominantemente para diagnóstico. Se podría establecer otra división grosso modo entre registrado *anatómico-anatómico* donde se combinan imágenes que muestran diferentes aspectos de la morfología de tejidos, o en *funcional-anatómico*, donde el metabolismo de los tejidos es relacionado con su ubicación espacial respecto a las estructuras anatómicas.

4.10. Sujeto

Cuando todas las imágenes involucradas en la tarea de registrado son adquiridas de un solo paciente, nos estaremos refiriendo al registrado *intrasujeto*. Si el registrado se realiza utilizando dos imágenes de pacientes diferentes (o un paciente y un modelo), es lo que se conoce como registrado *intersujeto*. Si una imagen proviene de un paciente y la otra imagen es construida de alguna manera a partir de la información contenida en una base de datos, formada por múltiples pacientes, entonces nos estamos refiriendo al registrado de *Atlas*.

El registrado *intrasujeto* es por mucho, el más frecuente de los tres, usado en casi todo tipo de procedimientos de diagnosis e intervención. El registrado *intersujeto* aparece principalmente en aplicaciones de imagen cerebral 3D/3D MR – CT. La naturaleza de la transformación de registrado es curva, estas aplicaciones son siempre intrínsecas, tanto la basada en segmentación como la basada en las propiedades del vóxel usando todo el contenido de la imagen. Generalmente es necesaria una inicialización manual por parte del usuario. El uso de registrado intersujeto y de atlas se puede ver de manera notable en áreas de compilación de estadísticos de la forma y tamaño de estructuras específicas, búsqueda de estructuras anómalas y transferencia de segmentaciones de una imagen a otra.

5. DISEÑO DE UNA HERRAMIENTA DE REGISTRADO

5.1. Introducción

Pensar en el concepto de *software* tan sólo como un inmenso conjunto de líneas de código puestas en marcha, es una idea totalmente obsoleta. Cada vez está más extendida la idea del *software* como producto de un riguroso estudio del problema y un previo modelado de la solución.

Modelar es importante en todas las disciplinas. Un claro ejemplo se observa en la arquitectura, pues la construcción de una casa sería harto difícil sin haber resuelto los planos a priori. En la informática sucede algo parecido. Es necesario visualizar y explicar las estructuras o el comportamiento de un sistema antes de comenzar a teclear. Por lo tanto, el primer paso a seguir en el diseño software es **crear un modelo**. Un modelo no es otra cosa que una simplificación de la realidad. Con ello se alcanzan básicamente cuatro objetivos:

- Visualizar cómo es o cómo se quiere que sea un sistema
- Permitir especificar las estructuras o comportamientos del sistema
- Proporcionar plantillas que guíen en la construcción de un sistema
- Documentar las decisiones que se han adoptado

En el *software* los modelos elegidos pueden afectar mucho a la visión del mundo. Si, por ejemplo, se construye un sistema con la mirada hacia bases de datos, habrá que centrarse en modelos entidad – relación. Si el punto de vista es hacia un sistema orientado a objetos, será necesario un modelo cuya arquitectura se centre en un mar de clases y los patrones que las gobiernan.

En el caso particular que nos atañe, el planteamiento se ha hecho como un análisis de estructuras, lo cual a desembocado en un modelo centrado en los algoritmos, con los datos fluyendo de proceso en proceso.

5.2. Tipos de datos

Puesto que se trata de un *software* orientado al registrado de imágenes médicas, los datos con los que se trabajan son principalmente imágenes bidimensionales de escala de grises y matrices de transformación, pero existe además otros tipos de datos que se pueden crear y cargar con la interfaz desarrollada. A continuación se describen todos los tipos de datos que se pueden manejar con la aplicación, así como una especificación de cómo van a ser implementados.

- **Imagen bidimensional en escala de grises:**

Se trata de una imagen médica en 2D, correspondiente a cualquiera de las modalidades vistas hasta ahora (CT, MR, US, etc.).

Una imagen digital bidimensional se representa mediante una retícula de celdillas a las que vamos asignando valores de brillo. Cada una de las celdillas de la retícula se denomina píxel (aunque ajeno al ámbito de este proyecto, decir también que si se trata de una imagen tridimensional, dichas celdillas reciben el nombre de **vóxel**). Este tipo de imágenes se conocen también como mapa de bits.

Desde el punto de vista matemático, dicho mapa de bits no es más que matriz bidimensional de $M \times N$ píxeles, tal y como muestra la Ilustración 5.1.

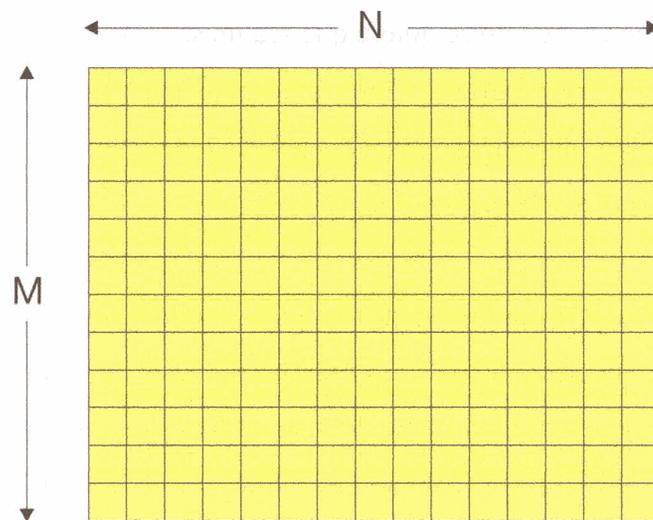


Ilustración 5.1. Una imagen es una matriz de $M \times N$ elementos.

Esta matriz almacena en cada una de sus posiciones un valor que representa el color del punto al que referencia. Si la imagen está en escala de grises, cada entrada de la matriz representa su valor de luminancia, i.e., la intensidad de gris del punto.

La imagen se muestra con una escala de grises, en la que los niveles más oscuros de la misma se corresponden con los niveles bajos de intensidad, y los niveles más claros se corresponden con los niveles más altos de intensidad.

Cada punto de la imagen viene especificado por dos coordenadas x_1 , y x_2 . A cada punto le corresponde un nivel de intensidad i , tal y como se expresa algebraicamente a continuación.

$$\begin{array}{l}
 I: \mathbb{R}^2 \longrightarrow \mathbb{R} \\
 (x_1, x_2) \longrightarrow i
 \end{array}$$

- **Lista de puntos clave o de control:**

Una lista de Puntos Clave (llamados también Puntos de Control o *Landmarks*), no es más que una matriz de $P \times 2$ en la que se almacena la posición x_1, x_2 de P puntos de control. Tal y como se representa en la Ilustración 5.2.

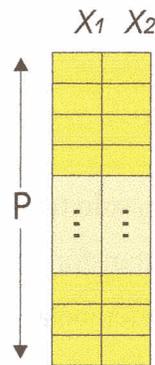


Ilustración 5.2. Matriz de Puntos Clave (PTC).

Algebraicamente, cada punto de control p queda definido por sus coordenadas espaciales x_1, x_2 .



- **Matriz de transformación afín**

Como se vio con anterioridad una transformación 3D afín, podía describirse como:

$$y_i = a_{ij}x_j$$

siendo x_i e y_i las coordenadas iniciales y las coordenadas deformadas respectivamente.

Para el caso de una transformación rígida en 2D, la ecuación está restringida a ser del tipo:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Donde t_i es un vector de traslación arbitrario, y r_{ij} es una matriz 2x2 de rotación definida tal que:

$$r_{ij} = \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix} \quad t_i = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

i.e., r_{ij} rota la imagen un ángulo ϕ .

En el caso de la transformación **afín**, r_{ij} solo tiene como restricción ser no singular.

Cuando operamos con matrices de transformación, con frecuencia es necesario poder expresarlas en diferentes sistemas de referencia.

El cambio de sistema de referencia se puede enunciar como la siguiente descomposición matricial.

$$N^{-1} \cdot T \cdot N$$

Donde N es la matriz de normalización.

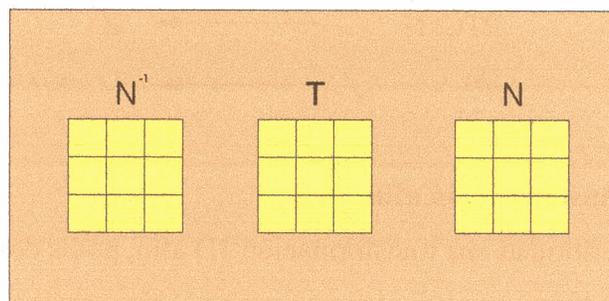


Ilustración 5.3. Matriz de transformación afín como descomposición bajo un cambio de sistema de referencia.

- **Campo de deformación disperso**

El problema del registrado puede contemplarse como la obtención de un campo de deformación que es necesario aplicar a una imagen para llegar a ajustarla a la imagen de referencia. El campo de deformación no es más que un campo vectorial que nos indica la magnitud y dirección del desplazamiento de los puntos o píxeles de una imagen. Se denomina Campo de deformación disperso a aquel campo que indica el desplazamiento de los puntos de control o *landmarks*.

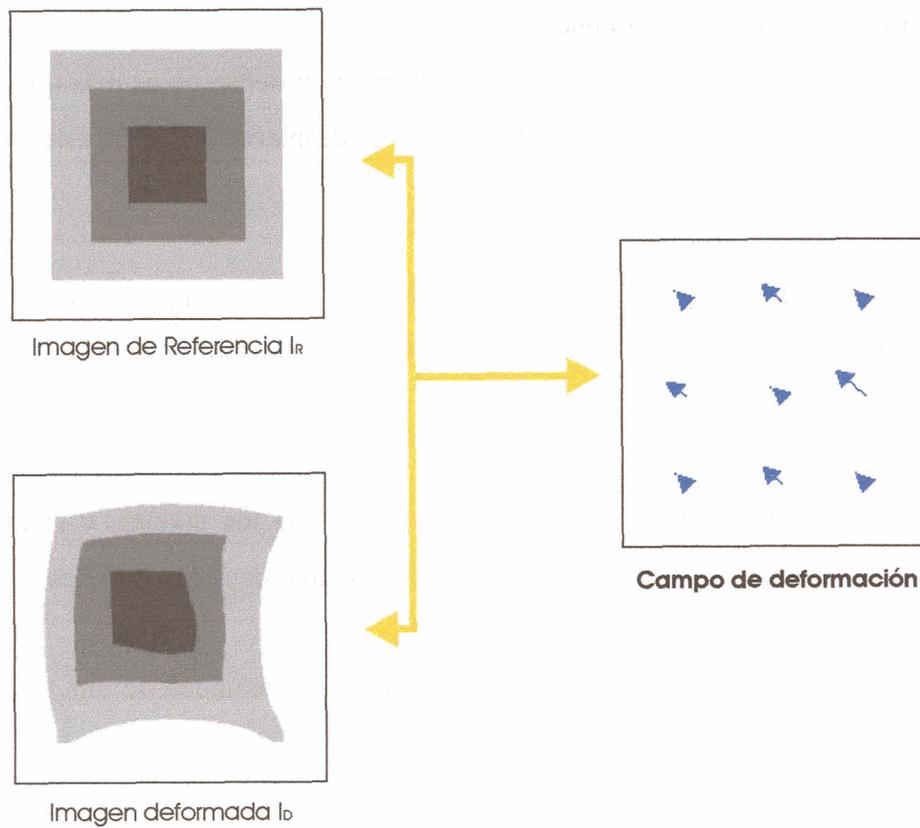


Ilustración 5.4.

El campo de deformación disperso se puede definir como una estructura con dos matrices de $P \times 2$, donde una matriz almacena las posiciones de P puntos de control (Matriz Pos), y la otra almacena los valores del desplazamiento sufrido por cada uno de dichos P puntos (Matriz Dif). En la Ilustración 5.5. se puede ver una representación esquemática de una estructura de campo de deformación disperso.

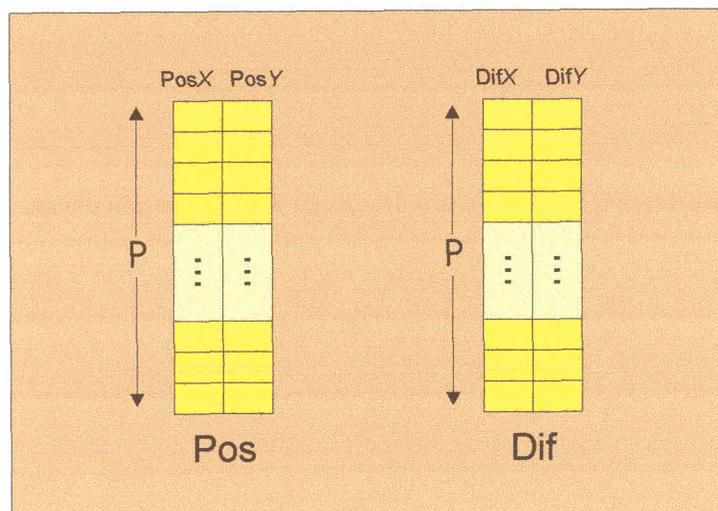


Ilustración 5.5. Estructura del campo de deformación disperso.

- **Campo de deformación denso**

Al igual que en el caso anterior, el campo de deformación denso no es más que un campo vectorial que nos indica la magnitud y dirección del desplazamiento en este caso de los píxeles de la imagen.

El campo vectorial denso se puede categorizar como una estructura constituida por dos matrices de $(M \times N) \times 2$ elementos, siendo $(M \times N)$ la resolución de la imagen (i.e. el número de píxeles que conforman la imagen).

Una de las matrices almacena la posición de cada uno de los píxeles, y la otra matriz contiene los valores del desplazamiento sufrido por cada uno de dichos píxeles.

En la Ilustración 5.6. se puede ver una representación esquemática de una estructura de campo de deformación denso.

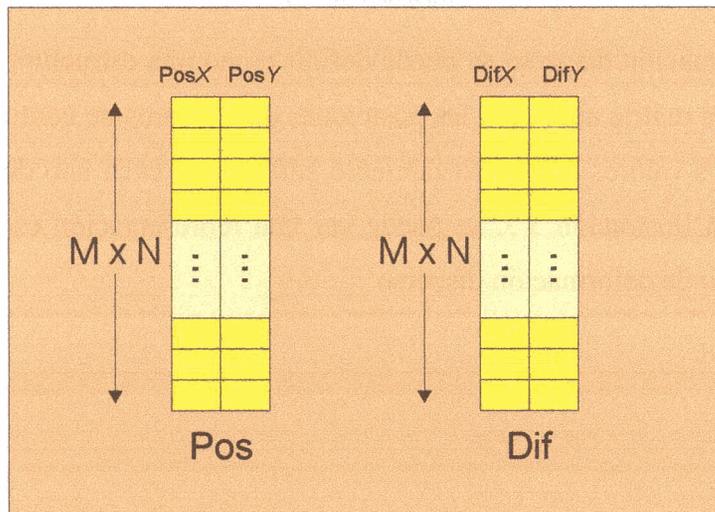


Ilustración 5.6. Estructura del campo de deformación denso.

5.3. Flujo de procesado

En la Tabla 5.2 se muestra a modo de grafo, la manera en que se pueden obtener y manipular los distintos tipos de datos que se utilizan en la aplicación. En la primera columna se sitúan los tipos de datos de origen y en la primera fila el dato resultante de aplicar la operación descrita en la casilla de intersección entre ambas. Las casillas nulas ponen de manifiesto la imposibilidad de pasar de los datos de origen indicados en la columna al dato resultante de la fila.

| | Lista de puntos clave | Matriz de transformación afin | Campo de deformación disperso | Campo de deformación denso | Imagen deformada |
|--|--|--|---|---|---|
| <ul style="list-style-type: none"> Imagen maestra Imagen dependiente | Selector manual de puntos clave | Registrado rígido de imagen | | Registrado no rígido imagen | |
| <ul style="list-style-type: none"> Lista de puntos clave maestra Lista de puntos clave dependiente | Sintonía fina de puntos clave | Registrado rígido de puntos clave | Registrado no rígido de puntos clave | | |
| <ul style="list-style-type: none"> Matriz de transformación afin Imagen dependiente | | | | Obtener campo de deformación denso a partir de transformación afin | Deformar a partir de transformación afin |
| <ul style="list-style-type: none"> Campo de deformación disperso Imagen dependiente | | | | Interpolación | Deformar a partir de campo de deformación disperso |
| <ul style="list-style-type: none"> Campo de deformación denso Imagen dependiente | | | | | Deformar a partir de campo de deformación denso |

Tabla 5.1. Procesos de los distintos tipos de datos.

La Ilustración 5.7. muestra el diagrama de flujo de la Tabla 5.1.

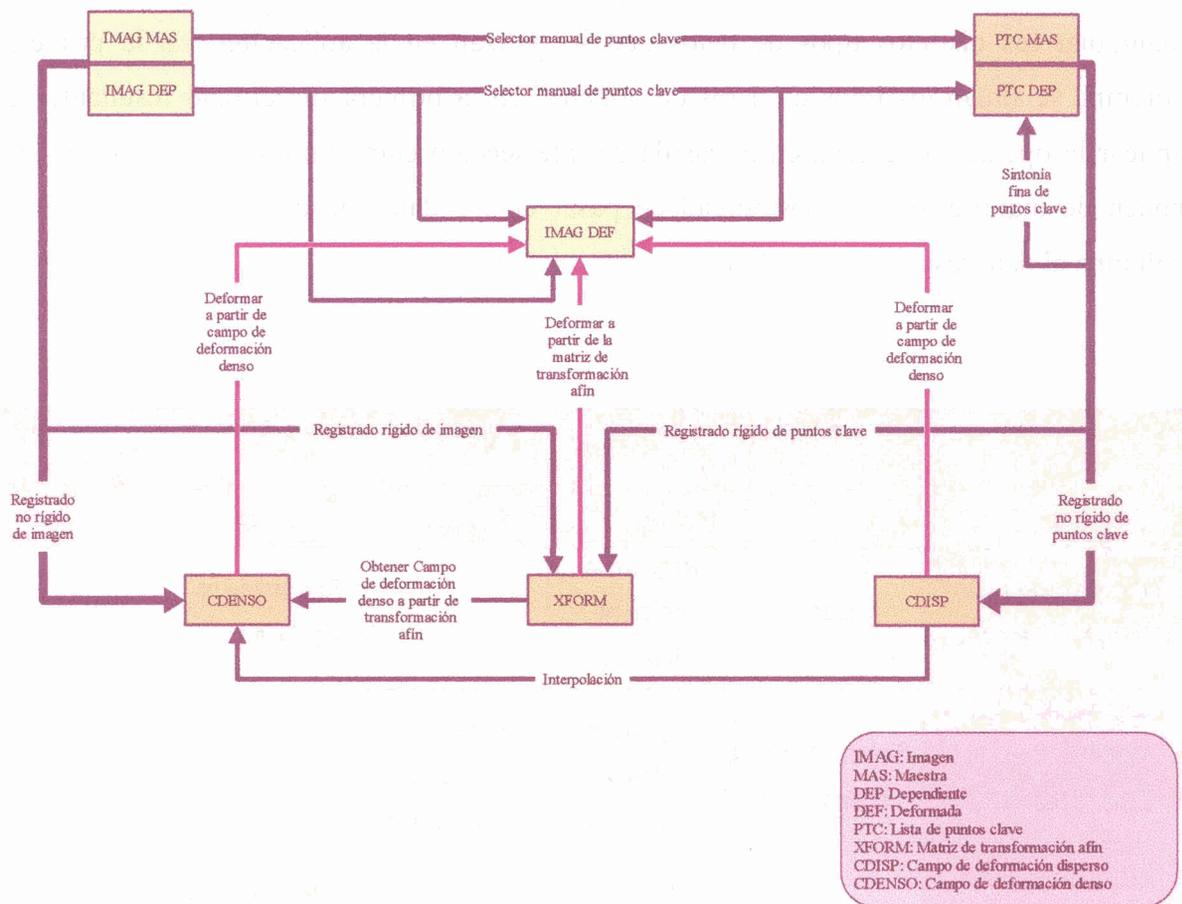


Ilustración 5.7. Diagrama de flujo de la Tabla 5.1.

5.4. Procesos

A continuación se describen todos y cada uno de los procesos enumerados en la Tabla 5.1.

- **Selector manual de puntos clave**

En este proceso es el propio usuario quien se encarga de definir los puntos clave o de control, mediante los cuales se va a realizar el registrado.

Para ello la aplicación proporciona las herramientas necesarias que permitan crear, editar y gestionar las listas de puntos clave, así como visualizar dichos puntos sobre ambas imágenes.

- **Sintonía fina de puntos clave**

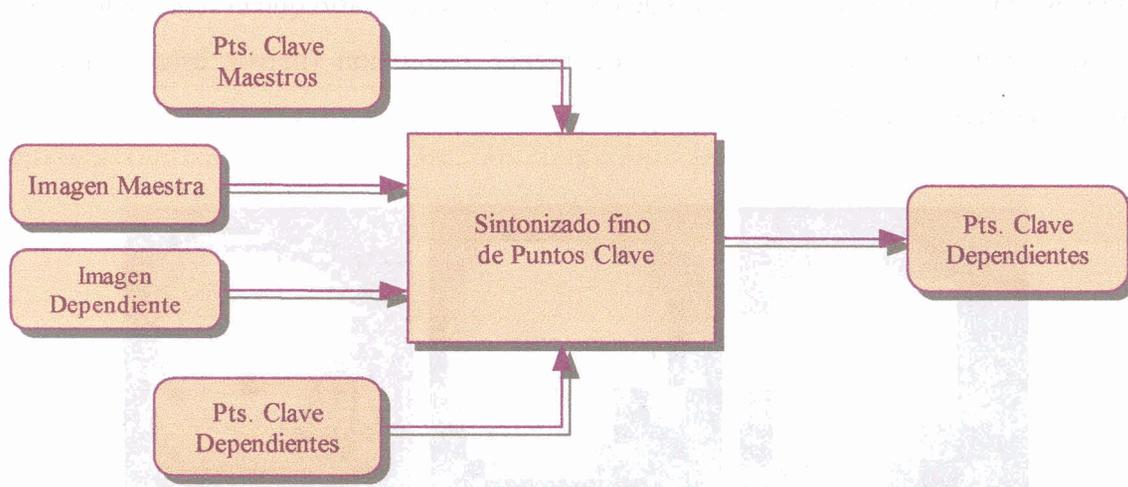


Ilustración 5.8

La herramienta de sintonía fina de puntos clave se utiliza para realizar un mejor ajuste de los puntos de control. Dado que los puntos de control son definidos por el usuario “a ojo” mediante el selector manual de puntos clave, es conveniente el uso de una herramienta automática que mejore dicha selección ajustando los puntos de control. Este ajuste se hará mediante la correlación cruzada, teniendo en cuenta la información de intensidad de gris de las imágenes. La herramienta de sintonía fina de puntos clave requiere como parámetros de entrada, la lista de puntos clave maestra, la lista de puntos clave dependiente, y las imágenes maestra y dependiente propiamente dichas, y devuelve como resultado una nueva lista de puntos clave dependientes ajustados.

- **Registrado rígido de imagen**



Ilustración 5.9

En este apartado vamos a realizar el registro basándonos únicamente en la información existente en las imágenes. Para ello es necesario introducir el concepto de multiescala.

En el procesamiento de imágenes en general, observamos que las estructuras que se aprecian dependen de la escala según la que se estudien. De igual manera sucede en el registrado de imágenes. Por tanto, para el registrado de las mismas es necesario que el registrado se haga en cada nivel, siguiendo un esquema de menor resolución (mayor escala) a mayor resolución (menor escala).

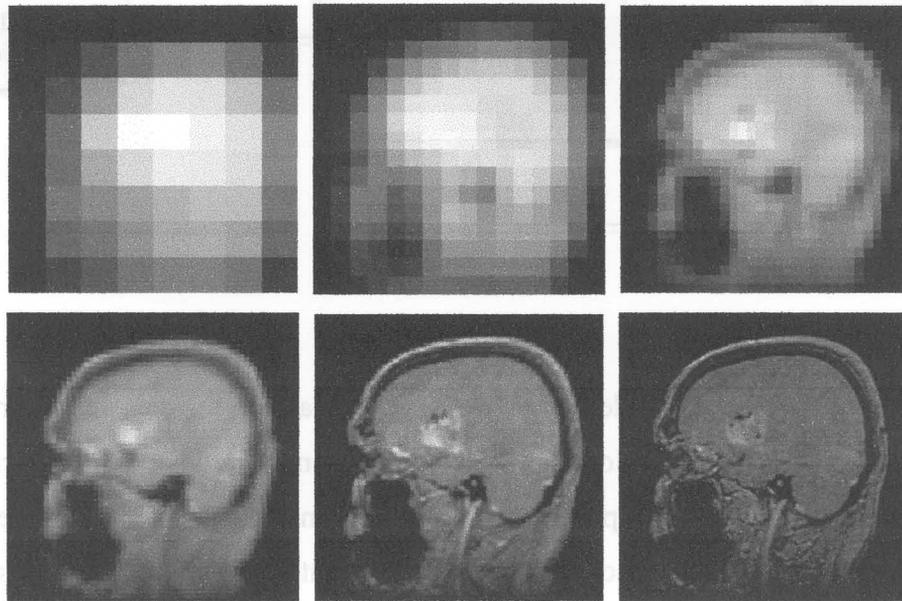


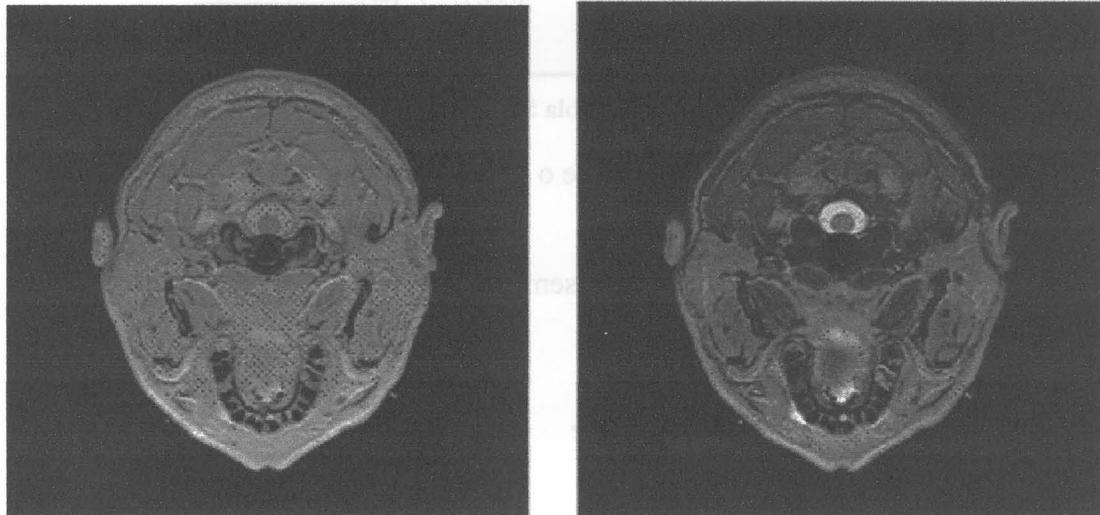
Figura 5.1

En la Figura 5.1. se muestra un corte sagital de la descomposición multiescala de un cerebro.

Basándonos exclusivamente en la información proveniente de la imagen, se podrá efectuar un registrado en cada escala con otra imagen proveniente de otra modalidad. El registrado se lleva a cabo buscando los parámetros de la transformación afin que maximizan el parecido de la imagen maestra con la imagen dependiente transformada. Este parecido se mide mediante una función de semejanza, que describimos a continuación.

Dados dos imágenes X e Y , podemos suponer la generación de las mismas como realizaciones de un vector aleatorio i que nos proporciona la intensidad en cada píxel. El aspecto de la función de densidad de probabilidad de i se representa con un ejemplo en la Figura 5.2.

A partir de la función de densidad de probabilidad existen estadísticos que miden la dependencia estadística de las componentes i_x e i_y de la misma. Usaremos estos estadísticos como medidas de semejanza. Algunos de estos estadísticos se muestran en la Tabla 5.2.



Resonancia Magnética - 1(X)

Resonancia Magnética - 2 (Y)

Figura 5.2

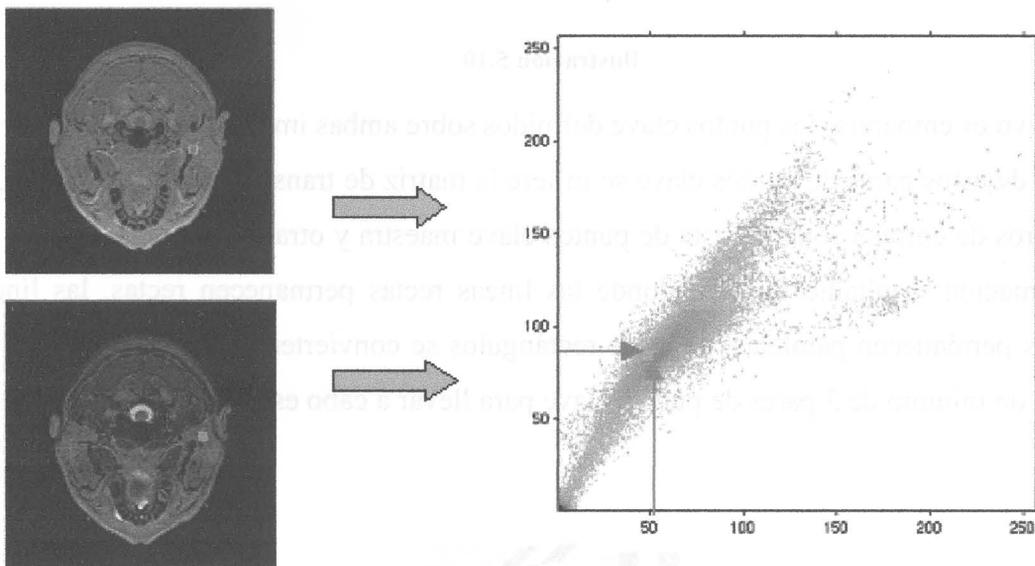


Figura 5.3. Vector aleatorio i a partir de X e Y .

- **Registrado no rígido imagen**



Ilustración 5.12

El registrado rígido nos proporciona una aproximación que en algunos casos puede ser grosera. Para un registrado más preciso es necesario una etapa no rígida. En ella deformamos la posición de los datos punto a punto, hasta conseguir una transformación que relacione ambas imágenes. La gran variedad de métodos existentes en el registrado no rígido han sido expuestos de forma sucinta en el capítulo 4.

- **Registrado no rígido de puntos clave**



Ilustración 5.13

El registrado no rígido de puntos clave, simplemente cuantifica la diferencia de posición entre cada punto clave y su homólogo. El proceso realiza la siguiente operación:

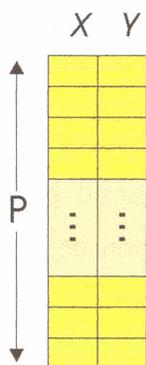


Ilustración 5.14. Puntos Clave Maestros

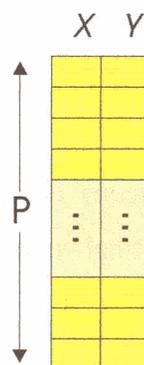


Ilustración 5.15. Puntos Clave Dependientes

- **Registrado no rígido imagen**



Ilustración 5.12

El registrado rígido nos proporciona una aproximación que en algunos casos puede ser grosera. Para un registrado más preciso es necesario una etapa no rígida. En ella deformamos la posición de los datos punto a punto, hasta conseguir una transformación que relacione ambas imágenes. La gran variedad de métodos existentes en el registrado no rígido han sido expuestos de forma sucinta en el capítulo 4.

- **Registrado no rígido de puntos clave**



Ilustración 5.13

El registrado no rígido de puntos clave, simplemente cuantifica la diferencia de posición entre cada punto clave y su homólogo. El proceso realiza la siguiente operación:

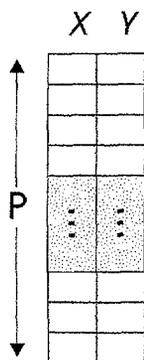


Ilustración 5.14. Puntos Clave Maestros

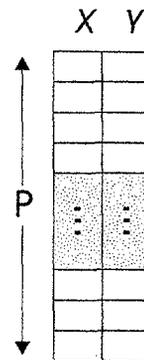


Ilustración 5.15. Puntos Clave Dependientes

Matriz de Posición = Puntos Clave Dependientes.

$$\text{Vector X} = \sum_{i_x=1}^P [\text{Puntos Clave Maestros}(i_x)] - [\text{Puntos Clave Dependientes}(i_x)]$$

$$\text{Vector Y} = \sum_{i_y=1}^P [\text{Puntos Clave Maestros}(i_y)] - [\text{Puntos Clave Dependientes}(i_y)]$$

Matriz de diferencias = [X, Y]

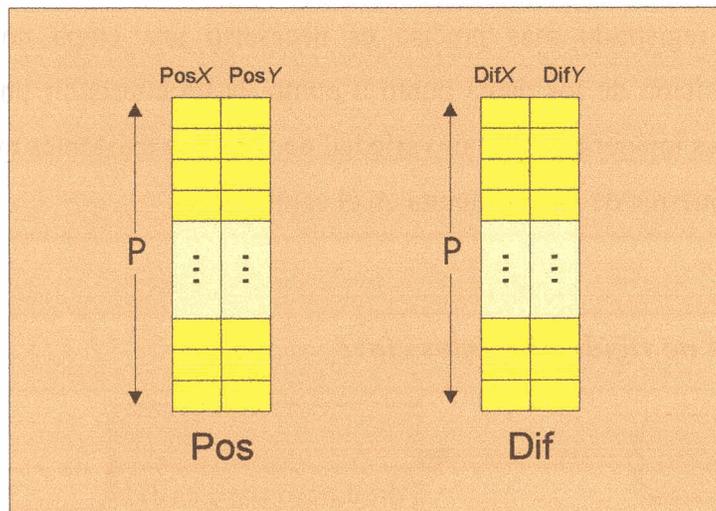


Ilustración 5.16. Campo Disperso.

El Campo disperso es una estructura que consta de dos matrices, una matriz de posiciones (Pos) y una de diferencias (Dif).

En la matriz de posiciones se almacena la posición de los puntos clave antes de realizar el desplazamiento, equivale por tanto a la posición de los puntos clave dependientes.

La matriz de diferencias almacena el desplazamiento de dichos puntos necesario para trasladarlos hasta su posición final, que en este caso no es más que la posición de los puntos clave maestros.

- **Obtener Campo de deformación denso a partir de Transformación afín**

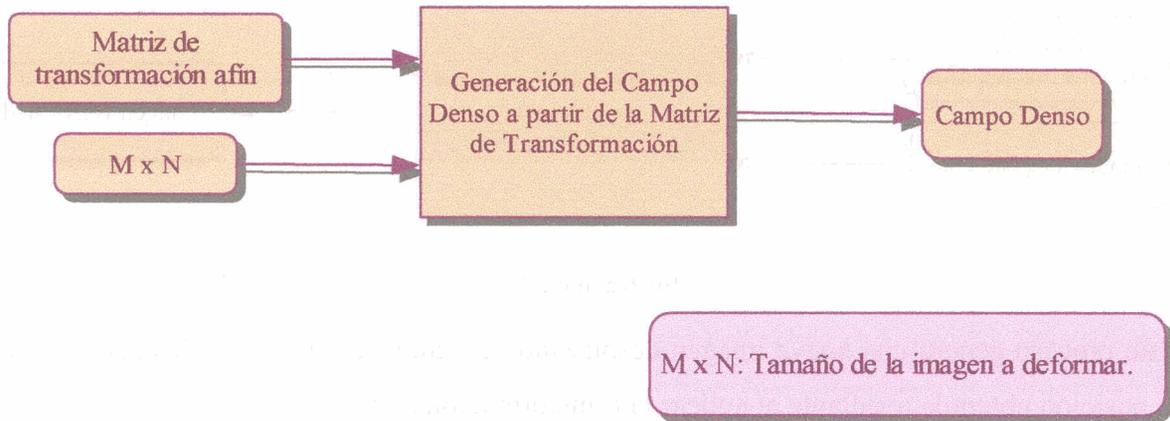


Ilustración 5.17.

Este proceso permite generar el campo denso que se representa el desplazamiento de cada píxel de la imagen, al ser ésta deformada mediante una transformación afín. Para ello basta multiplicar la posición de cada píxel por la matriz de transformación, restándole su posición original para obtener el desplazamiento.

Matriz de Puntos deformados = matriz de posiciones * Matriz de Transformación afín

Matriz de diferencias = matriz de puntos deformados - matriz de posiciones

- **Interpolación**

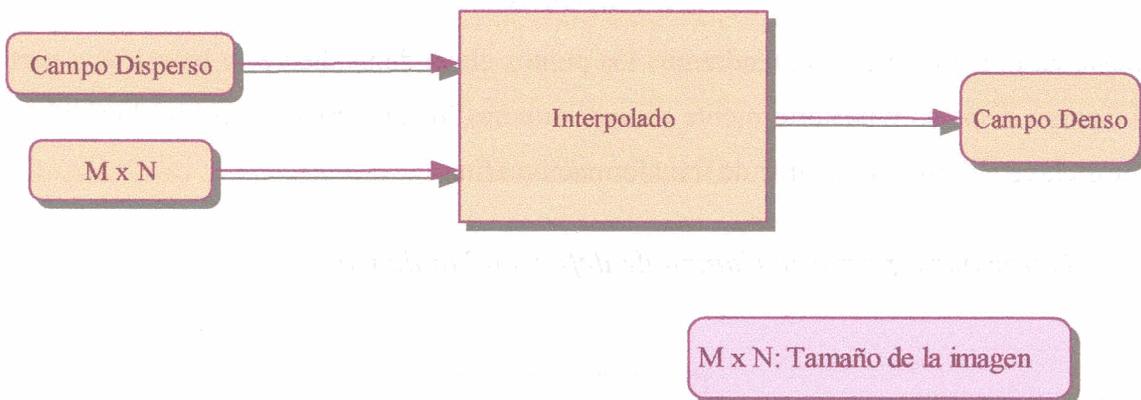


Ilustración 5.18

La interpolación nos va a hacer falta para conocer la deformación en aquellos píxeles en los que no esté definido el campo disperso.

- *Deformar a partir de Transformación afín*



Ilustración 5.19

Este proceso genera una nueva imagen desplazando el valor de intensidad de cada píxel a la posición obtenida resultante al aplicar la transformación afín.

- *Deformar a partir de Campo de deformación disperso*

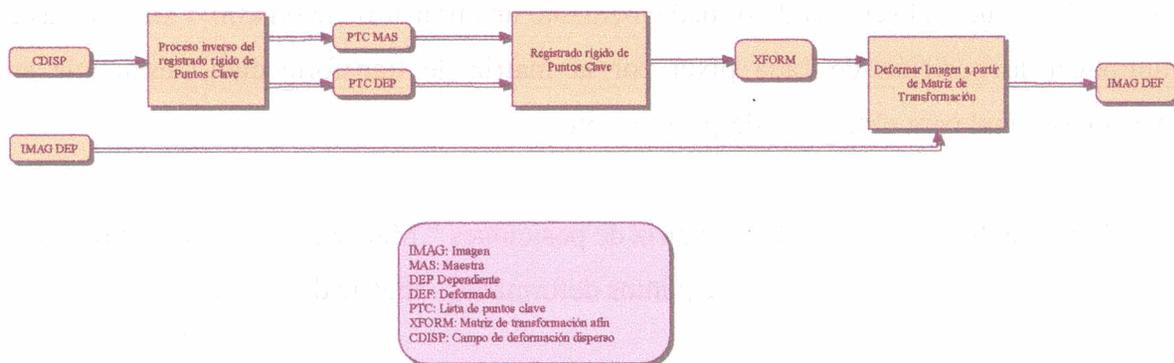


Ilustración 5.20

En este caso, simplemente obtendremos los puntos clave dependientes y maestros a partir del Campo disperso y posteriormente concatenaremos los procesos de registrado rígido de puntos clave y deformar a partir de transformación afín.

- *Deformar a partir de Campo de deformación denso*

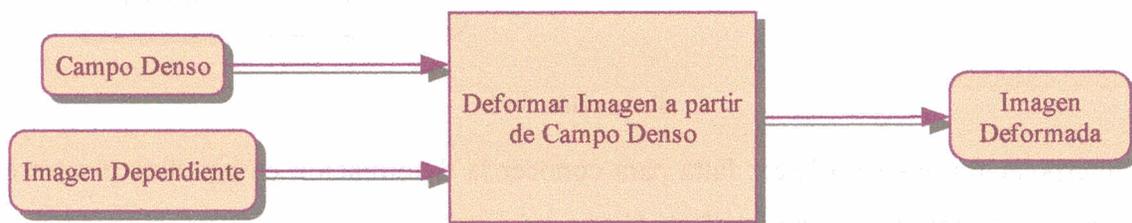


Ilustración 5.21

Al igual que a partir de transformación afin, generaremos una nueva imagen, pero esta vez desplazando según los valores indicados en el campo de deformación denso.

5.5. Plataforma

Los planos del edificio han sido consumados. Ahora es turno de elegir el terreno donde excavar unos cimientos que proporcionen estabilidad y que faciliten la construcción.

Con esta analogía se pretende dar una idea de lo que es la plataforma del *software*, al igual que en su momento se dio la idea del modelado identificándolo con los planos de un proyecto arquitectónico. De la misma manera que no se construye un edificio en cualquier parte, sino que se hace un estudio previo de los posibles terrenos, también es fundamental elegir la plataforma adecuada para que un proyecto *software* se desarrolle satisfactoriamente.

Lo primero que hay que hacer es determinar qué requisitos debe cumplir esta plataforma para que se puedan implementar las exigencias del software diseñado. Pero además de enumerar qué es lo que hace falta, sería conveniente señalar lo que no hace falta o no es requerimiento indispensable, pues de esta manera se abre mucho más el abanico de posibilidades. Una vez realizado este paso, se puede establecer una concurrencia entre las distintas plataformas. La manera de optar por una o por otra, puede hacerse discriminando las que menos cumplen con las necesidades anteriormente anunciadas o decidiendo a favor de las que aporten mayores ventajas.

5.5.1. Requisitos de implementación

Puesto que se trata de un programa basado en interfaces gráficas, es lógico pensar en una plataforma que facilite los métodos para producirlas y manejarlas . Aunque la utilización de imágenes provenientes de datos 3D no forma parte de este proyecto, si formaría parte esencial de una futura ampliación de la herramienta aquí desarrollada, y por lo tanto sería conveniente que la plataforma ofreciera funciones orientadas no solamente al procesado de imágenes sino también al procesado de señal en 3D.

Un requisito del cual no se ha hablado todavía, pero que suele ser muy frecuente y solicitado a la hora de seleccionar una plataforma, es el coste económico de ésta. Generalmente se suele buscar entre los múltiples soportes de código gratuitos que existen actualmente.

Está claro que, entre una plataforma que cumple con todas las exigencias pero es de precio elevado, y otra cuyo coste sea nulo y también cumple con todos los requisitos, aunque tal vez no sea *amigable* es natural inclinarse hacia la segunda opción. No obstante, y es ese el motivo por el cual no se ha mencionado esta cuestión hasta ahora, este proyecto ha sido propuesto para ser utilizado en laboratorios de investigación, para fabricar prototipos y modelos previos a la posible implementación de otro software destinado a ser comercializado.

De hecho, se convierte en una necesidad el crear una herramienta orientada a la creación de prototipos, que permita diseñar y depurar algoritmos de registrado. Por lo tanto no es objetivo principal el vender el producto a particulares con el fin de darle uso como aplicación directa, sino proporcionarlo a laboratorios capacitados con plataformas destinadas al uso científico y que demanden la creación de prototipos. Por esta misma razón, tampoco es obligatorio presentar un programa compilado, sino que basta que sea interpretado por la plataforma correspondiente.

Algo que sí sería digno de consideración, es la necesidad de que el programa pueda ser ejecutado en distintas máquinas, ya sea en Windows, Linux o Solaris. Se habla entonces de una **multiplataforma**. Es decir, es importante que la plataforma pueda funcionar bajo cualquier sistema operativo.

La plataforma finalmente designada debe, de acuerdo con lo escrito, cumplir con las siguientes condiciones:

- Comodidad en el diseño de interfaces gráficas.
- *Software* destinado a los prototipos.
- Disponibilidad de funciones encauzadas al procesado de imágenes y datos en 3D.
- Multiplataforma

5.5.2. Elección adecuada

Una vez estudiado los requisitos previos, ya se pueden poner en la bandeja varias opciones a fin de dar con el soporte adecuado. Se exponen a continuación una serie de alternativas argumentando los pros y los contras en cada una de ellas, hasta dar con la que más se adapta a las pretensiones de este trabajo.

Para empezar existen varias librerías como las MFC (*Microsoft Foundation Classes*) [16], o las GTK (*Gimp Toolkit*) [17], con las que se han desarrollado un gran número de interfaces gráficas. Pero ninguna de las dos es multiplataforma, por lo que no sería posible emplear el *software* en distintos entornos. Sólo se podría trabajar con la herramienta de registro bien en Windows o bien en Linux, dependiendo de si se escogiese MFC o GTK respectivamente.

Una alternativa bastante aceptable es la utilización de VTK (*Visualization Toolkit*) [9], a partir de TCLTK o de C++. De hecho, existen varios paquetes software de registro de datos, implementados con estas soluciones. Pero éste es también el motivo por el que no se ha optado por ellas.

Además, en estos entornos la implementación de funciones orientadas al procesamiento de imágenes, es muy compleja, por lo que sería más útil, como ya se comentó anteriormente, desarrollar primero algún prototipo que facilite la posterior implementación en otra plataforma, tal vez éstas.

Tal y como se ha planteado la situación, las vías conducen abiertamente a **Matlab®** (*Matrix Laboratory*). **Matlab®** es un entorno que con la adición de las librerías adecuadas, ofrece gran cantidad de herramientas orientadas al procesamiento de imágenes y datos 3D. Permite la elaboración de interfaces gráficas de usuario y es un sistema multiplataforma, por lo que puede funcionar bajo cualquier sistema operativo. Es por todo ello que **Matlab®** se presenta como la solución adecuada.

5.5.3. **Matlab®**

Matlab® es un entorno de computación y desarrollo de aplicaciones totalmente integrado orientado a llevar a cabo proyectos donde concurren elevados cálculos matemáticos y la visualización gráfica de los mismos.

Dispone de diversos programas de apoyo especializados, denominados *Toolboxes*, que extienden significativamente el número de funciones incorporadas en el programa principal. Estos *Toolboxes* cubren casi todas las áreas principales en el mundo de la ingeniería y la simulación.

Para la realización de este proyecto se ha hecho uso de los siguientes toolbox:

- *Image Processing 3.0*
- *Optimization 1.1*
- *Symbolic Math 2.1.2.*

En cuanto a las interfaces gráficas, **Matlab®** proporciona funciones orientadas a la creación de interfaces gráficas de usuario y que permiten el uso de botones, ejes, *sliders*, iconos, menús y otros elementos clásicos en dichas interfaces.

5.5.4. Decisiones de implementación

5.5.4.1. Sintonía fina de puntos clave

Se define una retícula de 11 x 11 regiones alrededor de cada punto clave de la imagen dependiente, así como para su homólogo en la imagen maestra.

Posteriormente se calcula la correlación entre los valores de cada píxel en la región. Seguidamente se busca la posición en la que el coeficiente de correlación es máximo, y se utiliza ésta como la posición óptima del punto clave.

Cada punto clave dependiente, únicamente puede desplazarse hasta un máximo de 4 píxeles, dependiendo del resultado de la correlación cruzada.

En el caso de que algunos puntos clave no puedan ser correlados, se devuelve el punto clave dependiente en su posición original.

5.5.4.2. Registrado rígido de imagen

En nuestro caso la interfaz hace uso de un algoritmo de registrado rígido de imagen en el que el funcional a optimizar, consiste en la información mutua existente entre las dos imágenes en cada posible posición de registrado. El método de optimización elegido es el BFGS [11].

5.5.4.3. Registrado no rígido imagen

Nuestra interfaz incorpora un algoritmo en el que se hace uso de una técnica conocida como *Block Matching*, ya usada en la codificación de video MPEG2, conjuntamente con un regularizado ponderado por estructura local [12].

5.5.4.4. *Obtener Campo denso a partir de Transformación afín*

Para obtener un campo denso a partir de una transformación afín, en primer lugar, se crea una matriz de posiciones (Pos) de $Q \times 3$ elementos que define Q puntos en dos dimensiones mediante coordenadas homogéneas. Dichos puntos almacenan la posición de los píxeles que conforman la imagen dependiente.

Cada punto queda determinado por la terna (x_1, x_2, x_3) . En nuestro caso, ya que trabajaremos en dos dimensiones, el tercer elemento (x_3) será siempre igual a 1. Q es igual al número de píxeles en la imagen, i.e. $Q = M \times N$.

A continuación se realiza la multiplicación de la matriz de transformación y la matriz anteriormente definida, para obtener la matriz de puntos deformados (Def), que contiene la posición final de los puntos dependientes una vez éstos han sufrido la transformación. Para obtener el Campo denso sólo es necesario obtener la matriz de las diferencias (Dif) entre la posición de los puntos iniciales y finales.

5.5.4.5. *Interpolación*

La interpolación rellena una superficie de la forma $Z = f(X)$ con los datos de vectores (generalmente) no uniformemente espaciados (x, z) . La superficie se interpola en los puntos especificados por x_i para producir z_i , donde x_i puede ser no uniforme.

- X es de dimensión $P \times 2$, representando P puntos en un espacio 2D.
- Z es de dimensión $P \times 1$, representando P valores de la superficie $f(X)$.
- x_i es un vector de dimensión $Q \times 2$, representando a Q puntos en un espacio 2D, cuyo valor de superficie ha de ser rellenado. x_i es generalmente una retícula uniforme.
- z_i es un vector de longitud P aproximando los valores de $f(x_i)$.

La superficie siempre pasa por los puntos (x, z) .

La interpolación es lineal, y está basada en una triangulación Delaunay de datos.

Mosaico de Delaunay

El mosaico de Delaunay (*Delaunay tessellation*) está formado por un conjunto de elementos singulares denominados **simplex**, tales que cada simplex no contiene ningún punto en la región que comprende. Dado que trabajamos en dos dimensiones, un simplex no es más que un triángulo (si trabajásemos en 3D, sería un tetraedro).

5.5.4.6. Deformar a partir de Transformación afín

La transformación espacial en 2D de la imagen dependiente a partir de la matriz de transformación, utiliza un método de remuestreo bilineal para aplicar la transformación espacial a una matriz de entrada (imagen dependiente), para obtener una matriz de salida (imagen deformada).

5.5.4.7. Deformar a partir de Campo de deformación denso

Para deformar la imagen dependiente se realiza la interpolación bilineal a trozos, basada en los valores de los dos puntos más cercanos en cada dimensión para los puntos del campo denso que sufren una traslación.

5.5.5. Empaquetado

El empaquetado de datos es la manera de distribuir el código en distintos archivos que conforman el programa. Para elaborar un código fluido es conveniente separar cada proceso en distintos módulos o funciones que son llamados unos a otros.

El software debe realizarse de manera modular, es decir, compuesto por distintos módulos independientes con el objeto de poder efectuar ampliaciones, modificaciones o actualizaciones futuras.

El esquema seguido a la hora de clasificar los distintos módulos lleva a dividir los ficheros en tres grandes grupos: ficheros que controlan datos, ficheros que controlan procesos y ficheros que controlan la interfaz.

Mediante los botones, listas desplegables, menús, etc., de la interfaz se activan las llamadas a las diferentes funciones mediante el *callback* i.e., la función que el usuario define para cada objeto de la interfaz.

El *callback* se activa cuando, una vez el programa esta ejecutándose, se selecciona mediante el puntero del ratón o el teclado.

Se empaqueta el software de la forma más modular posible, construyendo numerosos ficheros para cada acción. A continuación se da una idea acerca de qué tipos de ficheros deberían crearse.

5.5.5.1. *Ficheros de Datos*

Los ficheros de datos serán aquellos que se encargarán de abrir, guardar o cerrar algún tipo de dato. Las funciones de abrir deberán de lanzar una ventana de dialogo que permite al usuario elegir el dato a abrir, cargarlo en la interfaz y proceder a su correcta visualización. Las funciones de guardar permitirán salvar los datos en un archivo para un uso posterior. Los ficheros de datos deberán realizar las siguientes funciones:

- Abrir y visualizar una imagen (maestra o dependiente)..
- Abrir y visualizar una lista de puntos clave (maestra o dependiente)..
- Abrir y visualizar una matriz de transformación afin.
- Abrir y visualizar un campo de deformación disperso.
- Abrir y visualizar un campo de deformación denso.
- Guardar una imagen deformada.
- Guardar una lista de puntos clave (maestra o dependiente).
- Guardar una matriz de transformación afin.
- Guardar un campo de deformación disperso.
- Guardar un campo de deformación denso.
- Cerrar una imagen (maestra, dependiente, deformada).
- Cerrar una lista de puntos clave (maestra o dependiente).
- Cerrar una matriz de transformación afin.
- Cerrar un campo de deformación disperso.
- Cerrar un campo de deformación denso.

5.5.5.2. *Ficheros de Procesado*

Pertenecerán a este grupo todos aquellos archivos que impliquen algún proceso relacionado con los datos, como puede ser la modificación, o la generación de nuevos datos a partir de éstos.

Cada función se corresponderá con una acción diferente, de manera que no existirá ningún fichero que permita al usuario ejecutar dos procesos distintos, de esta manera se pretende conseguir aumentar la modularidad de la aplicación.

Los ficheros de Procesado deberían realizar las siguientes funciones:

- Sintonizado fino de puntos clave.
- Registrado rígido de imagen.
- Registrado rígido de puntos clave.
- Registrado no rígido de imagen.
- Registrado no rígido de puntos clave.
- Obtener campo de deformación denso a partir de transformación afin.
- Interpolación.
- Deformar a partir de transformación afin.
- Deformar a partir de campo disperso.
- Deformar a partir de campo denso.

5.5.5.3. *Ficheros de interfaz*

Bajo este epígrafe se englobarán las funciones relacionadas con la gestión de la interfaz gráfica de usuario, incluyendo las que se encargarán de la visualización de los distintos tipos de datos. Seguidamente se señalan algunas de estas funciones.

- Mostrar una imagen (maestra, dependiente, deformada).
- Mostrar una lista de puntos clave (maestra, dependiente).
- Mostrar la matriz de transformación afin.
- Mostrar el campo de deformación disperso.
- Mostrar el campo de deformación denso.
- Añadir un punto clave.
- Modificar un punto clave.
- Eliminar un punto clave.
- Indicar en la barra de estado, la posición y brillo de un píxel determinado.
- Controlar la visibilidad de la barra de estado.
- Controlar la visibilidad de la barra de herramientas.
- Controlar la visibilidad de los ejes de las imágenes.
- Modificar opciones de visualización de las imágenes.
- Proporcionar una herramienta de visualización tipo Zoom.
- Seleccionar el idioma de la interfaz.
- Controlar dinámicamente el tamaño de la ventana de la interfaz.
- Imprimir la interfaz.
- Lanzar un documento de ayuda.

6. MANUAL DE USUARIO



6.1. Introducción

La **interfaz para el registrado de imágenes médicas, REGIM**, es un programa orientado al diseño de prototipos de algoritmos de registrado y ha sido implementado bajo el entorno de programación de **Matlab® 6.0**.

Se pretende con este bloque procurar al usuario una guía que le ayude a manejar el programa.

Se informa ampliamente de las posibilidades de esta interfaz así como de sus características. Se ha ilustrado cada opción del programa con un ejemplo, y al final de este manual, se expone un caso sencillo de registrado, explicado paso a paso para que el usuario se familiarice de una manera rápida con ésta interfaz.

6.2. Instalación y ejecución de la interfaz

Para comenzar a utilizar el programa, será necesario instalar el Software **Matlab® 6.0** o versiones superiores, junto a los Toolbox de **Image Processing 3.0**, **Optimization 1.1** y **Symbolic Math 2.1.2**. Asimismo es necesario añadir el directorio **TENSORTOOLBOX** contenido en el directorio **REGIM**, al "search path" o trayecto de búsqueda de **Matlab®**, ello puede hacerse simplemente tecleando `addpath('tensortoolbox');` desde el *prompt* del **Matlab®**.

Dependiendo de la plataforma donde se haya instalado **Matlab®**, la manera de ejecutar la interfaz variará.

- **Windows 98, 2000, XP:**

- Sitúese en el directorio donde se hayan instalado los archivos de **REGIM** y haga doble clic con el ratón sobre el icono de **REGIM**.
- Alternativamente, puede ejecutarse el **Matlab®**, para ello bastará con hacer doble clic con el cursor del ratón sobre su correspondiente icono, y desde el *prompt* del **Matlab®**, llamar al programa tecleando `regim`, será necesario situarse previamente en el directorio donde se ubiquen los archivos de **REGIM**.

- **Linux, Solaris:**

- Abra el **Matlab®** tecleando `matlab` desde un terminal. Sitúese en el directorio donde se encuentra **REGIM** y teclee `regim`.

6.3. Entorno de trabajo

La ventana inicial, representada en la Figura 6.1, contiene una barra de estado, un menú de opciones, una barra de herramientas, y el área de trabajo. Las opciones de los menús y los botones de la barra de herramientas serán habilitadas o inhabilitadas en función de que su uso sea posible a lo largo del programa.

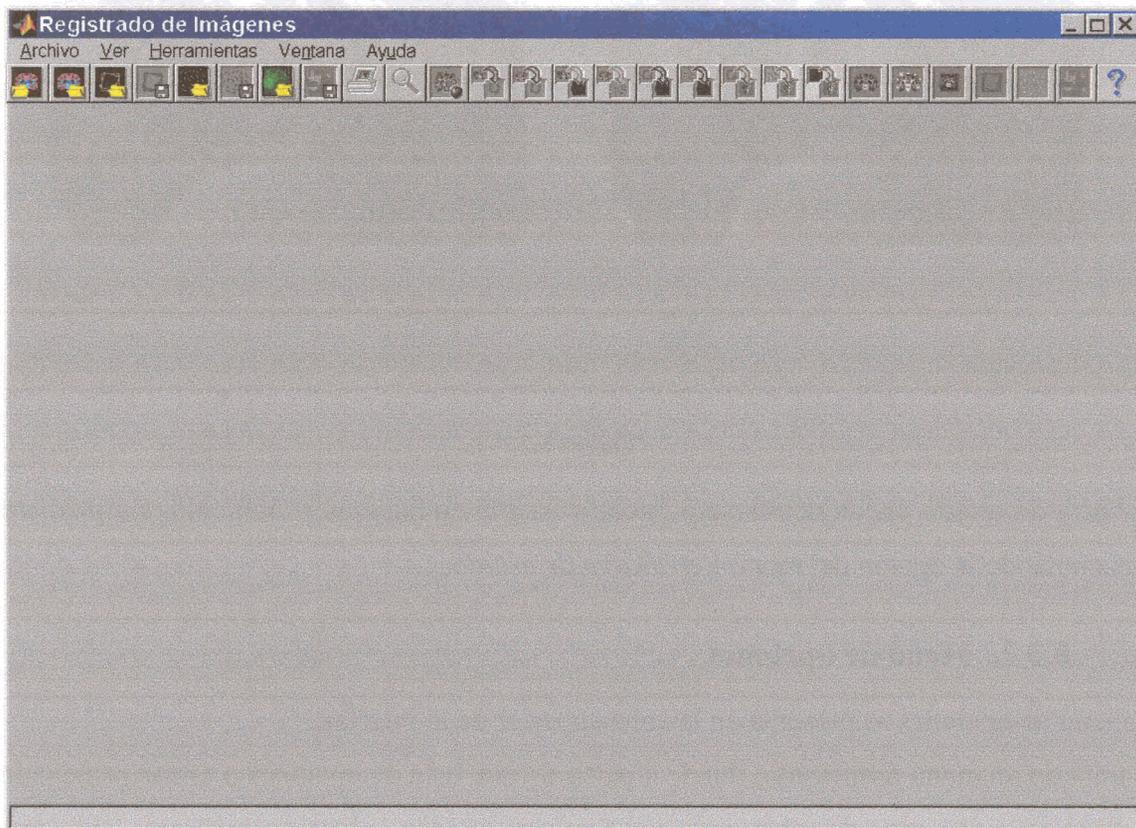


Figura 6.1

6.3.1. Barra de estado

Tanto en el modo [Ventana Imagen](#), como [Ventana imagen deformada](#), cuando el cursor del ratón se sitúa sobre una imagen visible; la barra de estado, situada en la parte inferior de la interfaz, muestra la posición (x, y) así como la intensidad del píxel sobre el que esta situado dicho cursor, tal y como muestra la Figura 6.2.

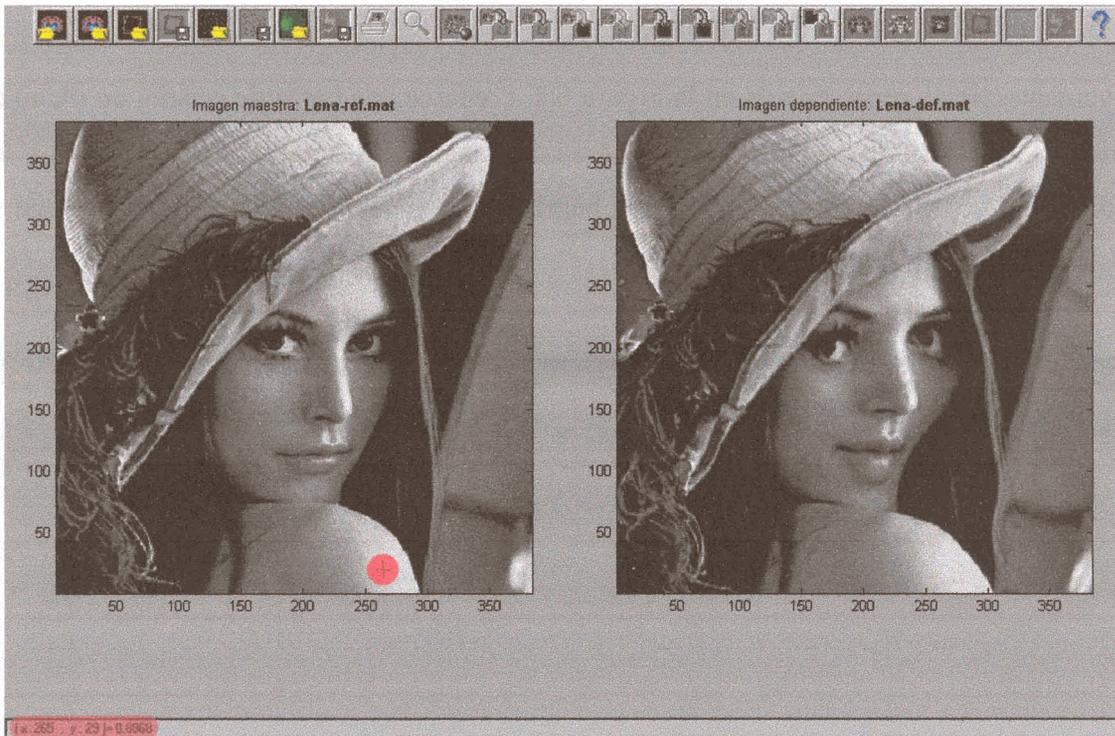


Figura 6.2

La barra de estado puede ocultarse o hacerse visible en cualquier momento, simplemente seleccionando la opción del menú *Ver/ Barra de estado*.

6.3.2. Menú de opciones

El menú de opciones se presenta en la zona superior de la interfaz.

Se trata de un menú ramificado, donde a partir de una lista de opciones o ramas principales se despliegan una serie de submenús, que a su vez, pueden volver a ramificarse, siguiendo una estructura de árbol.

El menú principal consta de cinco opciones; *Archivo*, *Ver*, *Herramientas*, *Ventana* y *Ayuda*.

Al iniciar la aplicación, la mayoría de las opciones del menú estarán desactivadas, a excepción de *Ayuda*, *Archivo/ Idioma*, *Ver/ Barra de herramientas*, *Ver/ Barra de estado*, *Archivo/ Salir*, y de aquellas opciones que permitan abrir algún tipo de dato, e.g. *Archivo/ Abrir/ Imagen Maestra*, *Archivo/ Abrir/ Campo Disperso*, etc.

Menú Archivo:

Bajo este menú se engloban las opciones relacionadas con la administración de datos agrupadas en tres menús secundarios o submenús, *Abrir*, *Guardar* y *Cerrar*; así como la opción *Imprimir*, un submenú de *Preferencias* que permite elegir el idioma de la interfaz y la opción *Salir* que permite terminar la aplicación.

El submenú *Abrir* se expande dando acceso a las opciones que permiten abrir y cargar imágenes, listas de puntos clave, matrices de transformación afin, campos vectoriales dispersos y campos vectoriales densos, como se puede ver en la Figura 6.3.

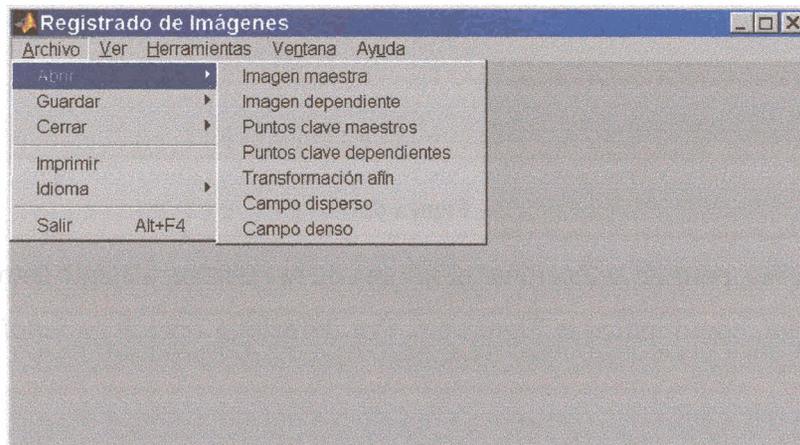


Figura 6.3

El submenú *Guardar*, da acceso a las opciones que permiten guardar listas de puntos clave, matrices de transformación afin, campos vectoriales dispersos y campos vectoriales densos, como muestra la Figura 6.4.

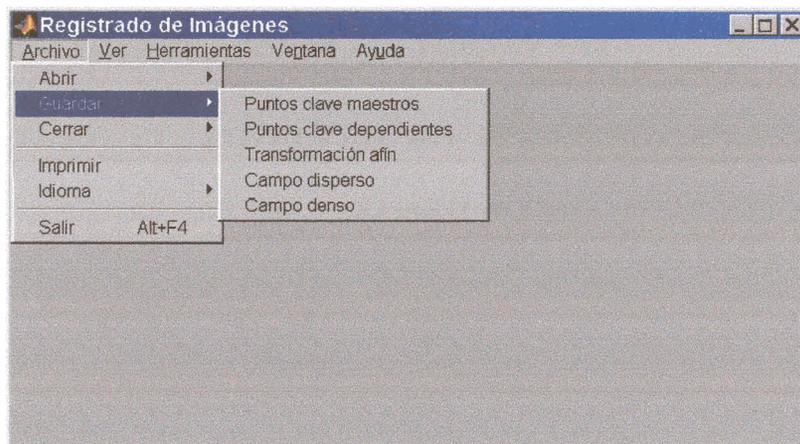


Figura 6.4

El submenú *Cerrar*, se expande mostrando las opciones que posibilitan cerrar los mismos tipos de datos que pueden ser abiertos, i.e. imágenes, listas de puntos clave, matrices de transformación afin, campos vectoriales dispersos y campos vectoriales densos, tal y como se observa en la Figura 6.5.

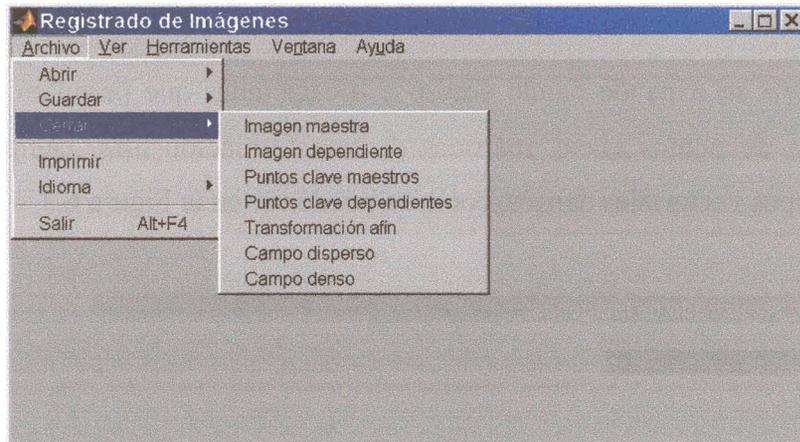


Figura 6.5

El submenú *Idioma*, permite seleccionar el idioma de la interfaz, a elegir entre las opciones *English* y *Español*, como indica la Figura 6.6. Por defecto la opción *Español* se encontrará seleccionada.

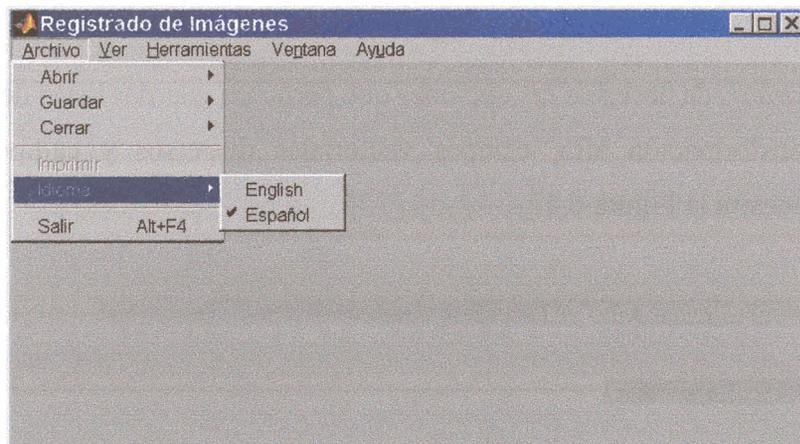


Figura 6.6



Menú Ver:

El menú *Ver*, representado en la Figura 6.7, comprende aquellas opciones que afectan al tipo de visualización, y permite al usuario hacer visibles u ocultar partes de la interfaz, a conveniencia.

De esta forma, las opciones *Ver/ Barra de herramientas* y *Ver/ Barra de estado* ocultarán o harán visible la barra de herramientas y la barra de estado respectivamente, según se encuentren seleccionadas o no.

Las opciones *Ver/ Datos maestros* y *Ver/ Datos dependientes* permiten (tanto en los modos de *Ventana Imagen* y *Ventana Puntos Clave*, como en el de *Ventana Imagen Deformada*), alternar entre la visualización singular de los datos (ya sean maestros o dependientes), y la visualización conjunta de ambos, según sean seleccionadas o no. Es posible activar o desactivar estas opciones utilizando métodos abreviados de teclado, sin necesidad de usar el ratón, tan sólo con pulsar simultáneamente la tecla `Ctrl` más el carácter que se indica.

La opción *Ver/ Ejes* permite hacer visibles u ocultar los ejes, según sea seleccionada o no.

La opción *Ver/ Zoom* activa o desactiva el zoom. Esta opción también dispone de un método abreviado de teclado.

El submenú *Ver/ Mapa de color*, permite escoger entre los diferentes mapas de colores, que se utilizarán a la hora de mostrar las imágenes. Por defecto se utiliza el *Gris*, que direcciona 64 niveles de grises.

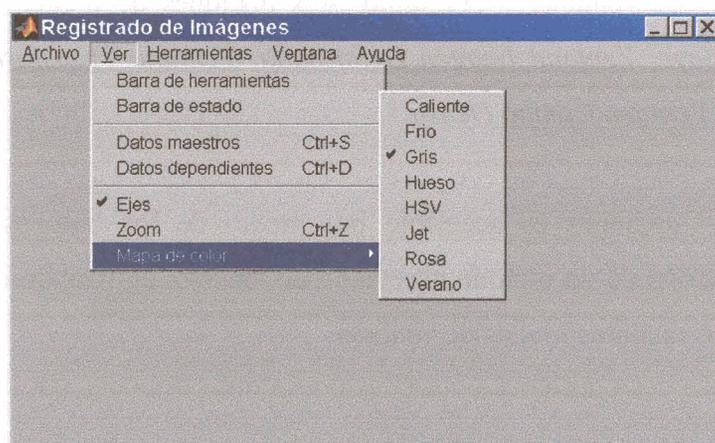


Figura 6.7



Menú Herramientas:

El menú *Herramientas*, abarca todas las opciones del menú que transforman o procesan los datos cargados de alguna manera. Hay diez herramientas como se puede ver en la Figura 6.8.

Cada herramienta es accesible también desde la barra de herramientas, tal y como muestra la Figura 6.11, la descripción de la acción que realiza cada uno de los botones de la barra de herramientas se puede ver en la en la Tabla 5.1..

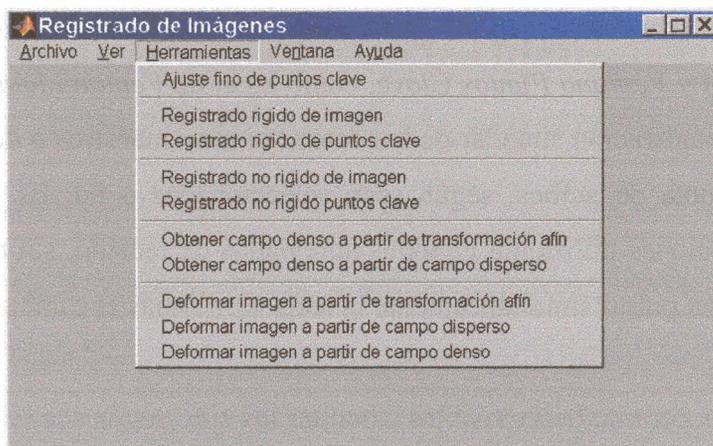


Figura 6.8

Menú Ventana:

Las opciones del menú *Ventana*, véase Figura 6.9, permiten alternar entre los distintos modos de ventana de la interfaz.

Existen seis modos de ventana, y por tanto, seis opciones de menú; *Ventana Imagen*, *Ventana Puntos Clave*, *Ventana Imagen Deformada*, *Ventana Transformación*, *Ventana Campo Disperso* y *Ventana Campo Denso*.

Cada una de estas opciones es accesible por medio de los botones de la barra de herramientas, y dispone de un método abreviado de teclado, agilizando de este manera la navegación entre los distintos modos de ventana.

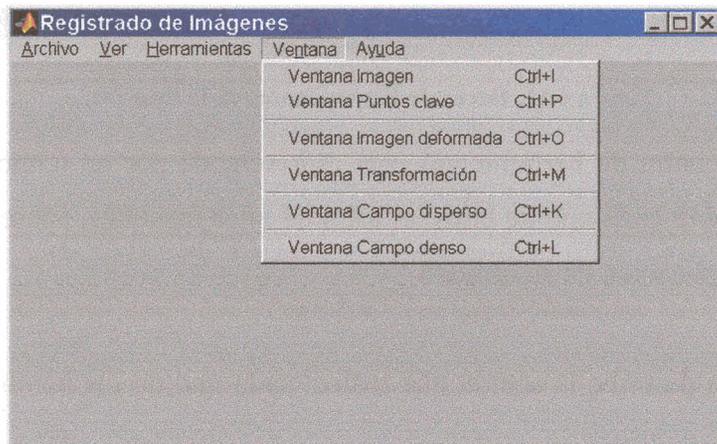


Figura 6.9

Menú Ayuda:

El menú *Ayuda*, ilustrado en la Figura 6.10, da acceso al manual de usuario a través de *Contenido*, y proporciona información sobre la interfaz pulsando en *Acerca de ...*

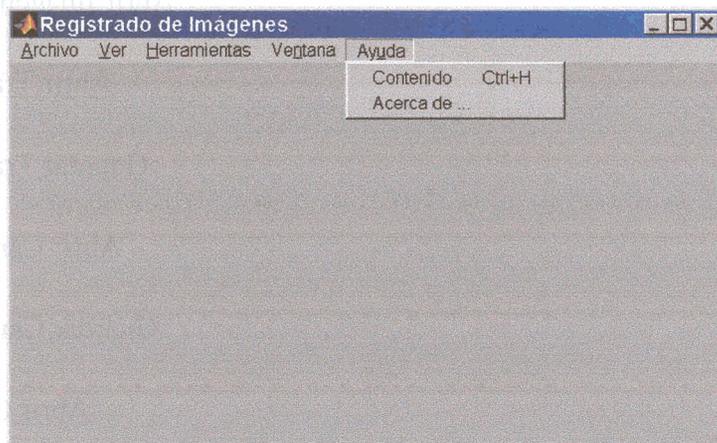


Figura 6.10

6.3.3. Barra de herramientas

Con el fin de hacer la interfaz más amigable y ágil, se ha creado la barra de herramientas, tal y como se puede ver en la Figura 6.11. La barra de herramientas permite seleccionar de manera sencilla, aquellas opciones del menú de uso más frecuente. La barra de herramientas puede ocultarse o hacerse visible en cualquier momento, simplemente seleccionando la opción del menú *Ver/ Barra de herramientas*.



Figura 6.11. Barra de herramientas de la interfaz.

Cada acción de la barra de herramientas está representada por un icono, y si se sitúa el cursor del ratón sobre cada uno de ellos, al cabo de unos instantes, aparecerá una etiqueta de ayuda con la explicación de su utilidad.

En la Tabla 5.1. se describe la acción que realiza cada uno de los botones de la barra de herramientas.

Tabla 6.1. Botones de la barra de herramientas.

| ICONO | ACCIÓN |
|-------|--|
| | <u>Abrir Imagen Maestra</u> |
| | <u>Abrir Imagen Dependiente</u> |
| | <u>Abrir Transformación</u> |
| | <u>Guardar Transformación</u> |
| | <u>Abrir Campo Disperso</u> |
| | <u>Guardar Campo Disperso</u> |
| | <u>Abrir Campo Denso</u> |
| | <u>Guardar Campo Denso</u> |
| | <u>Imprimir</u> |
| | <u>Zoom</u> |
| | <u>Ajuste Fino de Puntos Clave</u> |
| | <u>Registrado Rígido de Imagen</u> |
| | <u>Registrado Rígido de Puntos Clave</u> |
| | <u>Registrado No Rígido de Imagen</u> |

| | |
|---|---|
|  | <u>Registrado No Rígido de Puntos Clave</u> |
|  | <u>Obtener Campo Denso a partir de Transformación</u> |
|  | <u>Obtener Campo Denso a partir de Campo Disperso</u> |
|  | <u>Deformar a partir de Transformación</u> |
|  | <u>Deformar a partir de Campo Disperso</u> |
|  | <u>Deformar a partir de Campo Denso</u> |
|  | <u>Ventana Imagen</u> |
|  | <u>Ventana Puntos Clave</u> |
|  | <u>Ventana Imagen Deformada</u> |
|  | <u>Ventana Transformación</u> |
|  | <u>Ventana Campo Disperso</u> |
|  | <u>Ventana Campo Denso</u> |
|  | <u>Contenido de la Ayuda</u> |

| | | |
|---|---|---|
|  |  | ABRIR IMAGEN MAESTRA; ABRIR IMAGEN DEPENDIENTE |
|---|---|---|

Para cargar la imagen **maestra**, pulse con el cursor del ratón en el botón de la barra de herramientas Abrir Imagen maestra, o seleccione la opción de menú *Archivo/ Abrir/ Imagen Maestra*.

Para cargar la imagen **dependiente**, pulse con el cursor del ratón en el botón de la barra de herramientas Abrir Imagen dependiente, o seleccione la opción de menú *Archivo/ Abrir/ Imagen Dependiente*.

En ambos casos se abrirá una ventana de diálogo como la de la Figura 6.12, que permite al usuario seleccionar el archivo de imagen que quiere abrir.



Figura 6.12



ABRIR TRANSFORMACIÓN

Para cargar la matriz de transformación afín previamente guardada, pulse con el cursor del ratón en el botón de la barra de herramientas **Abrir Transformación**, o seleccione la opción de menú *Archivo/ Abrir/ Transformación*.

A continuación se abrirá una ventana de diálogo como la de la Figura 6.12, que permite al usuario seleccionar el archivo donde se ha almacenado la matriz de Transformación afín.



GUARDAR TRANSFORMACIÓN

Para guardar la matriz de transformación afín, pulse con el cursor del ratón sobre el botón de la barra de herramientas **Guardar Transformación**, o seleccione la opción de menú *Archivo/ Guardar/ Transformación*.

A continuación se abrirá una ventana de diálogo como la de la Figura 6.13, que permite al usuario asignar el nombre con el que quiere guardar el archivo. El sistema propondrá un nombre por defecto.



Figura 6.13



ABRIR CAMPO DISPERSO

Para cargar un campo vectorial disperso previamente guardado, pulse con el cursor del ratón en el botón de la barra de herramientas Abrir Campo Disperso, o seleccione la opción de menú *Archivo/ Abrir/ Campo Disperso*.

A continuación se abrirá una ventana de diálogo como la de la Figura 6.12, que permite al usuario seleccionar el archivo donde se ha almacenado el campo vectorial disperso.



GUARDAR CAMPO DISPERSO

Para guardar el campo vectorial disperso, pulse con el cursor del ratón sobre el botón de la barra de herramientas Guardar Campo Disperso, o seleccione la opción de menú *Archivo/ Guardar/ Campo Disperso*.

A continuación se abrirá una ventana de diálogo como la de la Figura 6.13, que permite al usuario asignar el nombre con el que quiere guardar el archivo. El sistema propondrá un nombre por defecto.



ABRIR CAMPO DENSO

Para cargar un campo vectorial denso previamente guardado, pulse con el cursor del ratón en el botón de la barra de herramientas Abrir Campo Denso, o seleccione la opción de menú *Archivo/ Abrir/ Campo Denso*.

A continuación se abrirá una ventana de diálogo como la de la Figura 6.12, que permite al usuario seleccionar el archivo donde se ha almacenado el campo vectorial denso.



GUARDAR CAMPO DENSO

Para guardar el campo vectorial denso, pulse con el cursor del ratón sobre el botón de la barra de herramientas Guardar Campo Denso, o seleccione la opción de menú *Archivo/ Guardar/ Campo Denso*.

A continuación se abrirá una ventana de diálogo como la de la Figura 6.13, que permite al usuario asignar el nombre con el que quiere guardar el archivo. El sistema propondrá un nombre por defecto.



IMPRIMIR

Para imprimir la interfaz, pulse con el cursor del ratón sobre el botón de la barra de herramientas Imprimir, o seleccione la opción de menú *Archivo/ Imprimir*.

A continuación se abrirá la ventana de dialogo de imprimir, tal y como se muestra en la Figura 6.14.

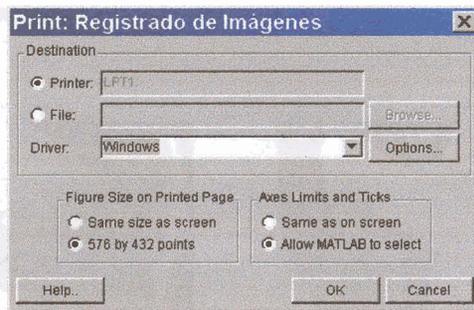


Figura 6.14



ZOOM

El zoom puede ser activado de varias maneras, bien pulsando sobre su icono en la barra de herramientas, bien seleccionando la opción *Ver/ Zoom* del menú, o bien presionado simultáneamente las teclas `Ctrl + Z` del teclado. Para desactivar el zoom, bastará con volver a realizar alguna de las acciones anteriores; [añadir un punto clave](#) o [editar un punto clave](#) también desactiva el zoom.

Existen dos maneras de hacer **zoom in** (ampliar detalle).

La primera consiste simplemente en pulsar una sola vez con el botón izquierdo del ratón (hacer clic) sobre la zona de la imagen que se quiere ampliar.

La segunda permite definir mediante una ventana rectangular la zona a aumentar, tal y como se muestra en Figura 6.15. Para ello bastará con pulsar el botón izquierdo del ratón, anclando así una esquina de la ventana, y mientras se mantiene pulsado dicho botón, arrastrar el cursor sobre la imagen hasta definir el tamaño de la ventana, en la Figura 6.16 se muestra la zona ya ampliada.

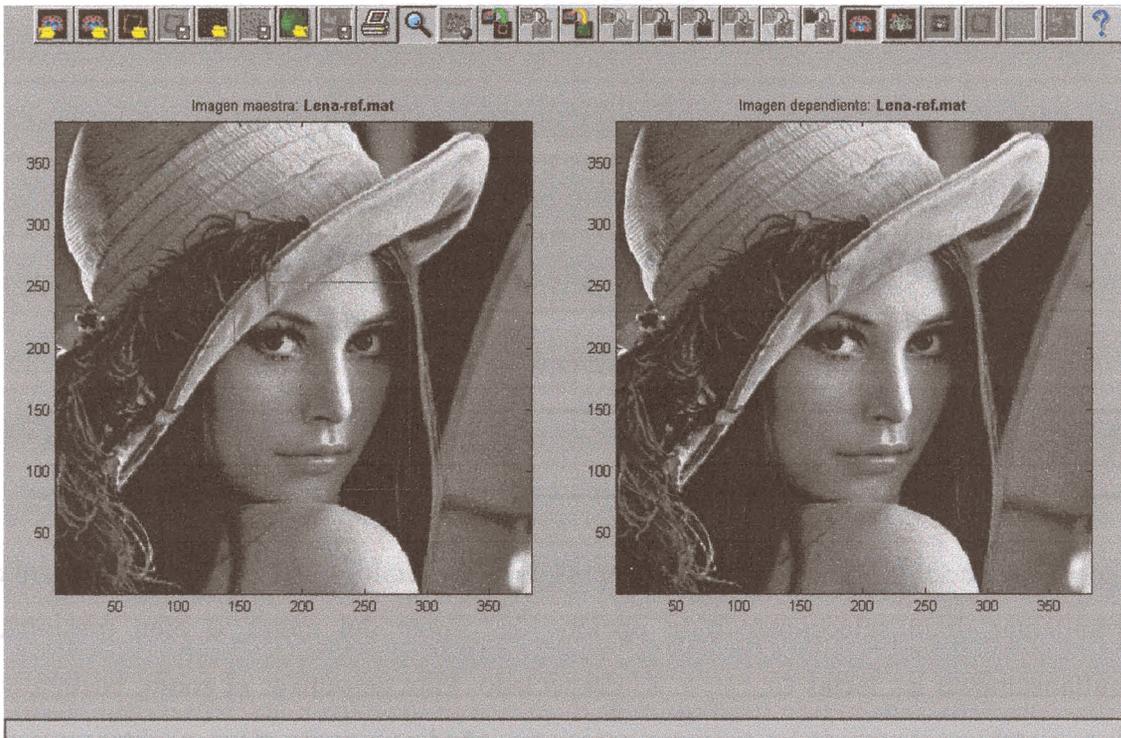


Figura 6.15 Definimos con una ventana la zona a ampliar mediante el zoom.

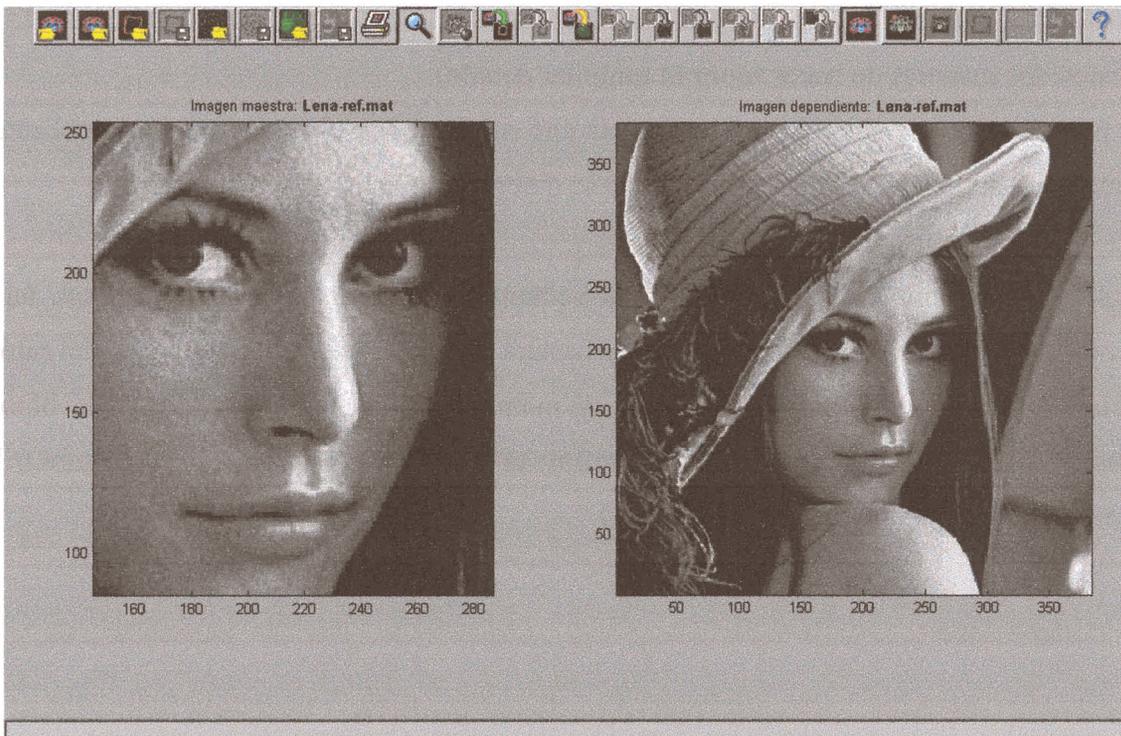


Figura 6.16 La zona dentro de la ventana es ampliada.

Se puede hacer **zoom in** varios niveles.

Para hacer **zoom out** un nivel, basta con hacer clic con el botón derecho del ratón.

Para regresar a la imagen original, pulsar dos veces el botón (hacer doble clic), sobre cualquier punto de la imagen.



AJUSTE FINO DE PUNTOS CLAVE

El Ajuste Fino de Puntos Clave, accesible tanto desde el botón homónimo de la barra de herramientas, como desde la opción de menú *Herramientas/ Ajuste Fino de Puntos Clave*, ajusta cada par de puntos clave por medio de la correlación cruzada normalizada de ambas imágenes.

Como resultado de esta operación se obtiene una nueva lista de puntos clave dependiente. En caso de que la anterior no hubiese sido guardada con anterioridad, aparecerá una ventana de diálogo que avisará al usuario de ésta situación y le permitirá salvarla.

Esta herramienta sólo se encontrará habilitada cuando coexistan tanto la lista de puntos clave maestra, como la lista de puntos clave dependiente y ambas contengan el mismo número de puntos. A su vez es necesario que exista también la imagen dependiente.



REGISTRADO RÍGIDO DE IMAGEN

Para realizar el Registrado Rígido de Imagen, pulse con el cursor del ratón sobre el botón homónimo de la barra de herramientas, o bien, seleccione la opción de menú *Herramientas/ Registrado Rígido de Imagen*.

Esta herramienta sólo se encontrará habilitada cuando coexistan tanto la imagen maestra, como la imagen dependiente.

Como fruto de esta operación se obtendrá la matriz de transformación afín, pasando la interfaz al modo [Ventana Transformación](#).

Una vez pulsado el Registrado Rígido de Imagen, se abrirá una ventana de diálogo como la de la Figura 6.17, que permite al usuario seleccionar el nivel de la pirámide que se usará

para los cálculos. Cuanto menor sea dicho nivel, menor será el tiempo que se tardará en realizar los cálculos necesarios; pero al mismo tiempo, menos preciso será el resultado final. Por defecto se propone un nivel que supone un equilibrio entre ambos factores. También se advierte al usuario que el proceso puede llevar varios minutos, permitiéndole elegir si desea continuar o no, tal y como se muestra en la Figura 6.18.

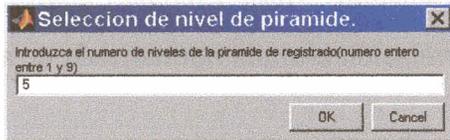


Figura 6.17

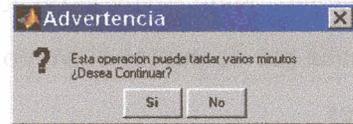


Figura 6.18



REGISTRADO RÍGIDO DE PUNTOS CLAVE

Para realizar el Registrado Rígido de Puntos Clave, pulse con el cursor del ratón sobre el botón homónimo de la barra de herramientas, o bien, seleccione la opción de menú *Herramientas/ Registrado Rígido de Puntos Clave*.

Esta herramienta sólo se encontrará habilitada cuando coexistan, tanto la lista de puntos clave maestra, como la lista de puntos clave dependiente y ambas contengan el mismo número de puntos. A su vez es necesario que exista también la imagen dependiente. Como fruto de ésta operación se obtendrá la matriz de transformación afin, pasando la interfaz al modo [Ventana Transformación](#).



REGISTRADO NO RÍGIDO DE IMAGEN

Para realizar el Registrado No Rígido de Imagen, deberá pulsar con el cursor del ratón sobre el botón homónimo de la barra de herramientas, o bien, seleccionar la opción de menú *Herramientas/ Registrado No Rígido de Imagen*.

Esta herramienta sólo se encontrará habilitada cuando coexistan, tanto la imagen maestra, como la imagen dependiente.

El resultado de ésta operación será un campo vectorial denso que se mostrará en el modo [Ventana Campo Denso](#).

Una vez pulsado el Registrado No Rígido de Imagen, se abrirá una ventana de diálogo como la de la Figura 6.17, que permite al usuario seleccionar el nivel de la pirámide que se usará para los cálculos. Cuanto menor sea dicho nivel, menor será el tiempo que se tardará en realizar los cálculos necesarios; pero al mismo tiempo, menos preciso será el resultado final. Por defecto se propone un nivel que supone un equilibrio entre ambos factores.

También se advierte al usuario que el proceso puede llevar varios minutos, permitiéndole elegir si desea continuar o no, tal y como se muestra en la Figura 6.18.



REGISTRADO NO RÍGIDO DE PUNTOS CLAVE

Para realizar el Registrado No Rígido de Puntos Clave, pulse con el cursor del ratón sobre el botón homónimo de la barra de herramientas, o bien, seleccione la opción de menú *Herramientas/ Registrado No Rígido de Puntos Clave*.

Esta herramienta sólo se encontrará habilitada cuando coexistan, tanto la lista de puntos clave maestra, como la lista de puntos clave dependiente y ambas contengan el mismo número de puntos.

El resultado de esta operación será un campo vectorial disperso que se mostrará en el modo [Ventana Campo Disperso](#).



OBTENER CAMPO DENSO A PARTIR DE TRANSFORMACIÓN

Para Obtener el Campo Denso a partir de la Transformación, deberá pulsar con el cursor del ratón sobre el botón homónimo de la barra de herramientas, o bien, seleccionar la opción de menú *Herramientas/ Obtener Campo Denso a partir de Transformación*.

Esta herramienta se encontrará habilitada cuando existan tanto la imagen dependiente como la matriz de Transformación.

El resultado de esta operación será un campo vectorial denso que se mostrará en el modo [Ventana Campo Denso](#).



OBTENER CAMPO DENSO A PARTIR DE CAMPO DISPERSO

Para Obtener el Campo Denso a partir del Campo Disperso, deberá pulsar con el cursor del ratón sobre el botón homónimo de la barra de herramientas, o bien, seleccionar la opción de menú *Herramientas/ Obtener Campo Denso a partir del Campo Disperso*.

Esta herramienta se encontrará habilitada cuando existan tanto la imagen dependiente como el campo vectorial disperso.

El resultado de esta operación será un campo vectorial denso que se mostrará en el modo [Ventana Campo Denso](#).



DEFORMAR A PARTIR DE TRANSFORMACIÓN

Para obtener la imagen deformada a partir de la transformación, deberá pulsar con el cursor del ratón sobre el botón Deformar a partir de Transformación de la barra de herramientas, o bien, seleccionar la opción de menú *Herramientas/ Deformar a partir de Transformación*.

Deformar a partir de Transformación, se habilitará cuando existan tanto la imagen dependiente como la matriz de Transformación.

Esta herramienta aplica las transformaciones contenidas en la matriz de transformación afin a la imagen dependiente, proporcionando así la imagen deformada, que se mostrará en el modo [Ventana Imagen Deformada](#).



DEFORMAR A PARTIR DE CAMPO DISPERSO

Para obtener la imagen deformada a partir de un campo disperso, deberá pulsar con el cursor del ratón sobre el botón Deformar a partir de Campo Disperso de la barra de herramientas, o bien, seleccionar la opción de menú *Herramientas/ Deformar a partir de Campo Disperso*.

Deformar a partir de Campo Disperso, se habilitará cuando existan tanto la imagen dependiente como el campo vectorial disperso.

Esta herramienta, en primer lugar, genera la matriz de transformación a partir del campo disperso, habilitando el modo [Ventana Transformación](#). Posteriormente aplica las transformaciones calculadas a la imagen dependiente, proporcionando así la imagen deformada, que pasará a mostrarse en el modo [Ventana Imagen Deformada](#).



DEFORMAR A PARTIR DE CAMPO DENSO

Para obtener la imagen deformada a partir de un campo denso, deberá pulsar con el cursor del ratón sobre el botón Deformar a partir de Campo Denso de la barra de herramientas, o bien, seleccionar la opción de menú *Herramientas/ Deformar a partir de Campo Denso*.

Deformar a partir de Campo Denso, se habilitará cuando existan tanto la imagen dependiente como el campo vectorial denso.

Esta herramienta, genera la imagen deformada, aplicando píxel a píxel, los desplazamientos contenidos en el campo vectorial denso, dicha imagen deformada se mostrará en el modo [Ventana Imagen Deformada](#).



Pulsar con el cursor del ratón en el botón de la barra de herramientas Contenido de la Ayuda, seleccionar la opción de menú *Ayuda/ Contenido*, o bien pulsar simultáneamente las teclas **Ctrl + H**, muestra el manual de usuario en formato html.

6.3.4. Área de trabajo

El área de trabajo abarca el espacio entre la barra de herramientas y la barra de estado, y en ella se mostraran los distintos modos de trabajo o modos de ventana.

Existen seis modos de ventana, es posible moverse entre los distintos modos de ventana, bien haciendo uso del menú de opciones, de la barra de herramientas o de los métodos abreviados de teclado.



6.3.4.1. Modo Ventana Imagen

El modo Ventana Imagen se habilita y activa al [abrir una imagen](#), tanto si se trata de la imagen maestra, como de la dependiente, permitiendo visualizar hasta dos imágenes al mismo tiempo.

En el eje situado a la izquierda de la interfaz, se mostrará la imagen **maestra**, que será la imagen de referencia a la hora de hacer el registrado, y en el eje de la derecha, se mostrará la imagen **dependiente**, es decir, aquella imagen que queremos transformar, tal y como se observa en la Figura 6.19.

El modo Ventana Imagen es compatible con el modo [Ventana Puntos Clave](#), esto es, los dos pueden estar activos al mismo tiempo.

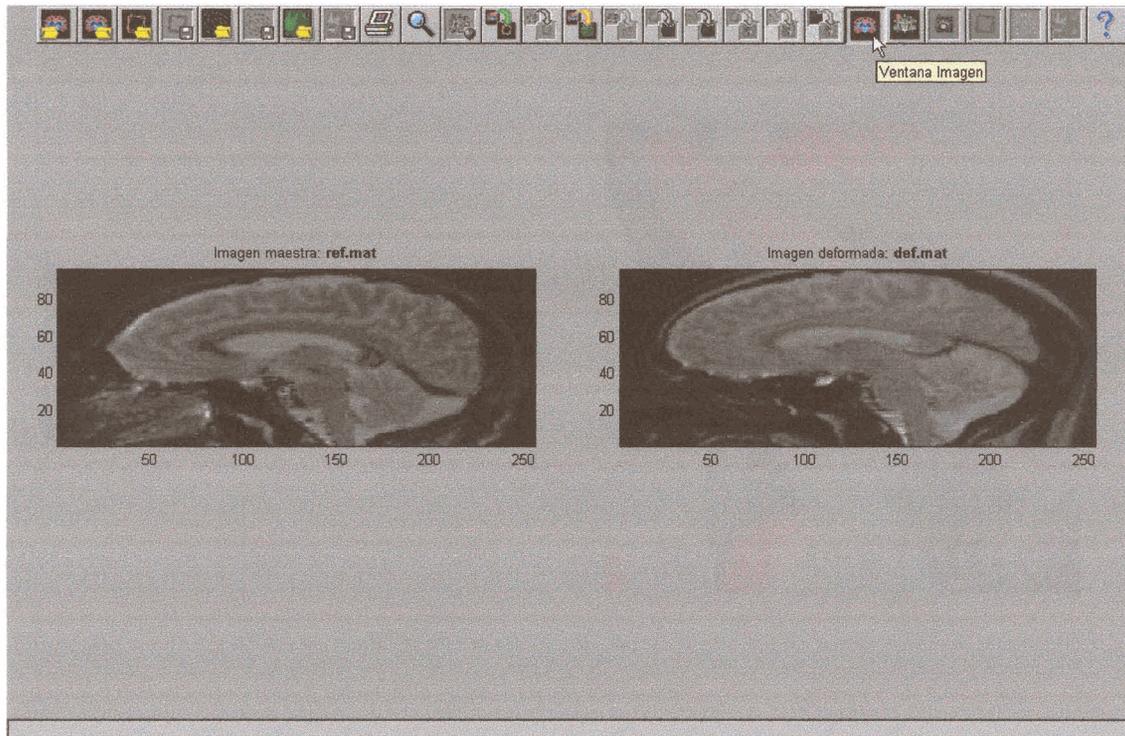


Figura 6.19

Al situar el cursor del ratón sobre cualquier píxel de una imagen, éste se convierte en un cursor de precisión al tiempo que en la barra de estado se indica posición x e y del píxel, así como su valor de intensidad, tal y como se observaba en la Figura 6.2.

Mediante las opciones del menú *Ver/ Datos Maestros* y *Ver/ Datos Dependientes*, es posible ocultar una de las dos imágenes (la imagen maestra o la dependiente), aunque no las dos a la vez. Esto permite ampliar la imagen visible de manera que se aproveche mejor el área de trabajo de la interfaz.

Otra manera de ver mejor los detalles de una imagen es haciendo uso del [zoom](#), utilidad que es habilitada desde el momento en el que se abre una imagen, véase la Figura 6.16.

En *Ver / Mapa de color*, se puede cambiar la paleta de colores con la que se muestran las imágenes; por defecto está activado el *Gris*. En la Figura 6.20 se muestran las imágenes usando el mapa de color *Jet*.

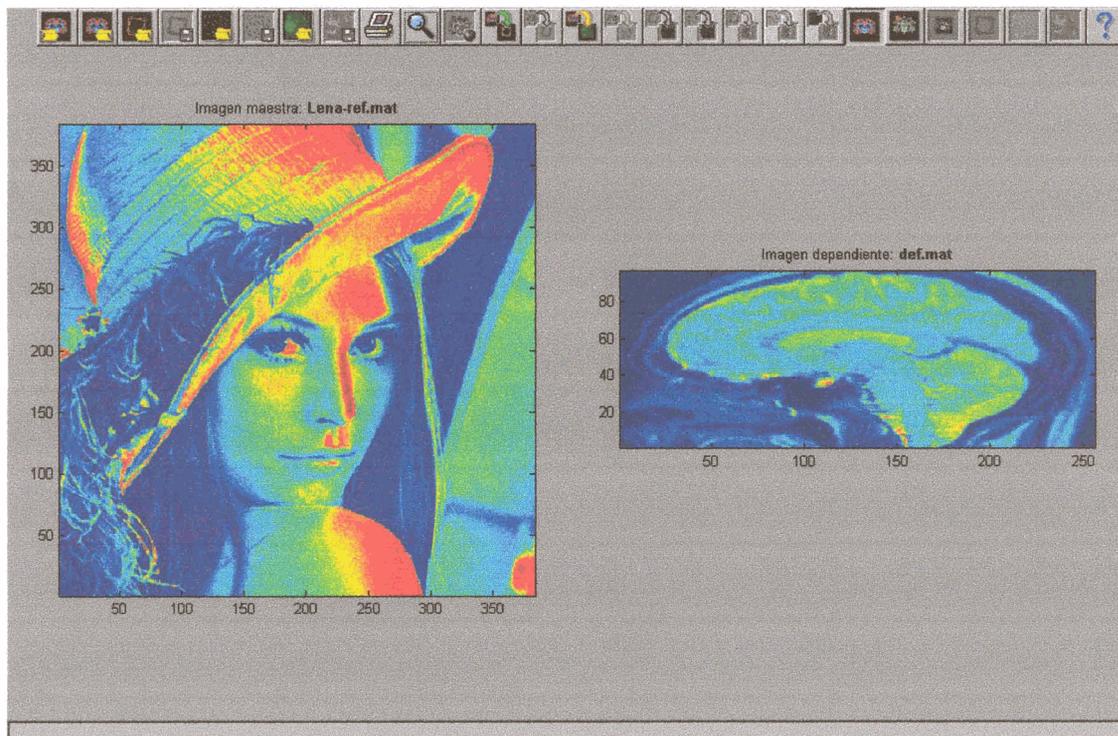


Figura 6.20

El método abreviado de teclado de ventana imagen es `Ctrl + I`.

Desde el momento en el que se hayan cargado las dos imágenes, se habilitarán las herramientas de [Registrado Rígido de Imagen](#) y [Registrado No Rígido de Imagen](#).



VENTANA PUNTOS CLAVE

6.3.4.2. Modo Ventana Puntos Clave

El modo Ventana Puntos Clave se habilita y activa, bien al [abrir una imagen](#), bien al [abrir una lista de puntos clave](#). Este modo sirve para mostrar listas de puntos clave, al tiempo que proporciona el llamado **selector manual de puntos clave**, es decir una serie de herramientas, que permiten la creación y gestión de listas de puntos clave.

Los modos [Ventana Imagen](#) y Ventana Puntos Clave son compatibles, y se activan ambos por defecto al abrir una imagen, tal y como indica la Figura 6.21. La razón de esto, es que sólo se puede añadir o [editar un punto clave](#), si se hace sobre una imagen visible. Al fin y al cabo, no tiene sentido [añadir un punto clave](#) si no es identificándolo con un elemento de la imagen.

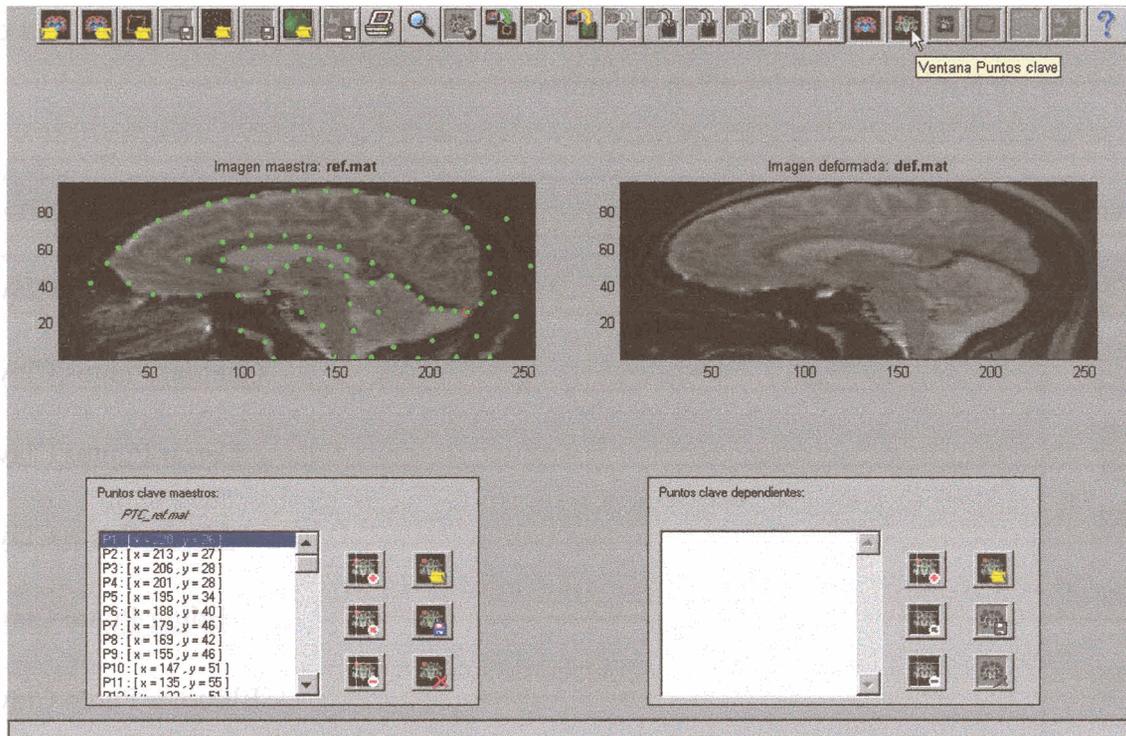


Figura 6.21

Se pueden mostrar hasta dos listas de Puntos Clave al mismo tiempo (la maestra y la dependiente).

A su vez, es posible ver solamente una de ellas, para ello bastará con deseleccionar, mediante las opciones del menú *Ver/ Datos Maestros* y *Ver/ Datos Dependientes*, cuál de las dos listas de Puntos Clave abiertas queremos ocultar.

El método abreviado de teclado de Ventana Puntos Clave es **Ctrl + P**.

Desde el momento en el que se hayan cargado o creado dos listas de puntos clave con igual número de puntos, se habilitará la herramienta [Registrado Rígido de Puntos Clave](#), si a su vez también existe la imagen dependiente, se habilitarán las herramientas [Ajuste Fino de Puntos Clave](#), y [Registrado No Rígido de Puntos Clave](#).

- **Selector manual de puntos clave**

El Selector manual de puntos clave está formado por una serie de botones y una lista de selección, dichos botones se describen a continuación:

| ICONO | ACCIÓN |
|---|---|
|  | <u>Abrir Puntos Clave</u> |
|  | <u>Guardar Puntos Clave</u> |
|  | <u>Cerrar Puntos Clave</u> |
|  | <u>Añadir Punto Clave</u> |
|  | <u>Editar Punto Clave</u> |
|  | <u>Eliminar Punto Clave</u> |

La lista de selección identifica cada punto clave por medio de una P seguida de su número de orden en dicha lista, al tiempo que muestra la posición (x, y) de cada uno.

Para seleccionar un punto, basta con hacer clic sobre dicho punto en la lista, una vez seleccionado, una arandela de color rojo lo identificará sobre la imagen.

Hacer doble clic sobre un punto en una lista de selección, no sólo selecciona dicho punto, sino que también selecciona el punto de igual orden de la otra lista, si es que existe.

| | |
|---|---------------------------|
|  | ABRIR PUNTOS CLAVE |
|---|---------------------------|

Para cargar la lista de puntos clave **maestra**, pulse con el cursor del ratón en el botón Abrir Puntos Clave situado en el marco de Puntos clave maestros, o seleccione la opción de menú *Archivo/ Abrir/ Puntos clave maestros*.

Para cargar la lista de puntos clave **dependiente**, pulse con el cursor del ratón en el botón Abrir Puntos Clave situado en el marco de Puntos clave dependientes, o seleccione la opción de menú *Archivo/ Abrir/ Puntos clave dependientes*.

En ambos casos se abrirá una ventana de diálogo que permite al usuario seleccionar el archivo donde se encuentra almacenada la lista de puntos clave que se quiere abrir.



GUARDAR PUNTOS CLAVES

Para guardar la lista de puntos clave **maestra**, pulse con el cursor del ratón en el botón Guardar Puntos Clave situado en el marco de Puntos clave maestros, o seleccione la opción de menú *Archivo/ Guardar/ Puntos clave maestros*.

Para guardar la lista de puntos clave **dependiente**, pulse con el cursor del ratón en el botón Guardar Puntos Clave situado en el marco de Puntos clave dependientes, o seleccione la opción de menú *Archivo/ Guardar/ Puntos clave dependientes*.

A continuación se abrirá una ventana de diálogo que permite al usuario asignar el nombre con el que quiere guardar el archivo. El sistema propondrá un nombre por defecto.



CERRAR PUNTOS CLAVE

Para cerrar la lista de puntos clave **maestra**, pulse con el cursor del ratón en el botón Cerrar Puntos Clave situado en el marco de Puntos clave maestros, o seleccione la opción de menú *Archivo/ Cerrar/ Puntos clave maestros*.

Para cerrar la lista de puntos clave **dependiente**, pulse con el cursor del ratón en el botón Cerrar Puntos Clave situado en el marco de Puntos clave dependientes, o seleccione la opción de menú *Archivo/ Cerrar/ Puntos clave dependientes*.

En el caso de que la lista de puntos clave que se va a cerrar no estuviese guardada, aparecerá una ventana de diálogo que avisará al usuario de esta circunstancia, al tiempo que le permitirá guardarla antes de continuar.



AÑADIR PUNTO CLAVE

Para añadir un punto clave, en primer lugar, presione el botón correspondiente, seguidamente sitúe el cursor del ratón sobre la imagen. Notará que el cursor se transforma en una cruz de precisión, finalmente pulse el botón izquierdo del ratón.

El nuevo punto añadido se mostrará sobre la imagen como un punto verde rodeado de una arandela roja. Dicha arandela nos indica que el punto recién añadido es el punto seleccionado. Asimismo dicho punto aparece seleccionado en la lista de selección donde se le identifica con un número y se nos muestra su posición.

Si desea salir del modo añadir punto clave sin añadir ningún punto, bastará con que pulse de nuevo el botón de añadir punto clave.

El modo añadir punto clave desactiva el [zoom](#).



EDITAR PUNTO CLAVE

Este botón sólo se encuentra habilitado si existe algún punto en la lista.

Para editar un punto clave, en primer lugar, seleccione el punto que desea editar mediante la lista de selección. El punto seleccionado se muestra sobre la imagen rodeado por una arandela de color rojo.

Posteriormente, presione el botón editar punto clave, a continuación sitúe el cursor del ratón sobre la imagen. Notará que el cursor se transforma en una cruz de precisión, finalmente pulse el botón izquierdo del ratón.

Si se ha editado el punto con éxito, se mostrará su nueva posición sobre la imagen, al tiempo que se borrará la antigua. El punto sigue siendo el punto seleccionado.

Si desea salir del modo editar punto clave, sin editar ningún punto, bastará con que pulse de nuevo el botón de editar punto clave.

El modo editar punto clave desactiva el [zoom](#).



ELIMINAR PUNTO CLAVE

Este botón sólo se encuentra habilitado si existe algún punto en la lista.

Para eliminar un punto clave, en primer lugar, seleccione el punto que desea eliminar mediante la lista de selección, el punto seleccionado se muestra sobre la imagen rodeado por una arandela de color rojo. Finalmente pulse sobre el botón eliminar punto clave.



VENTANA IMAGEN DEFORMADA

6.3.4.3. Modo Ventana Imagen Deformada

El modo Ventana Imagen Deformada se habilita y activa al deformar la imagen dependiente, ya sea mediante la herramienta [Deformar a partir de Transformación](#), de [Deformar a partir del Campo Disperso](#), o de la herramienta [Deformar a partir del Campo Denso](#).

El modo Ventana Imagen Deformada permite visualizar al mismo tiempo, tanto la imagen maestra como la imagen deformada, posibilitando el que se puedan comparar ambas imágenes, tal y como se observa en la Figura 6.22. A su vez, es posible ver solamente una imagen. Para ello bastará con deseleccionar, mediante las opciones del menú *Ver/ Datos Maestros* y *Ver/ Datos Dependientes*, cuál de las dos imágenes queremos ocultar (en este caso la imagen deformada es el dato dependiente).

El método abreviado de teclado de Ventana Imagen Deformada es `Ctrl + O`.

Tanto en el caso de la deformación obtenida a partir de la matriz de transformación afin, como en el de la obtenida a partir del campo vectorial disperso, junto a la imagen deformada se muestran también unos marcos que ayudan a la comprensión de la deformación que ha sufrido la imagen dependiente.

En el caso de la deformación obtenida a partir del campo vectorial denso no se muestran dichos marcos.

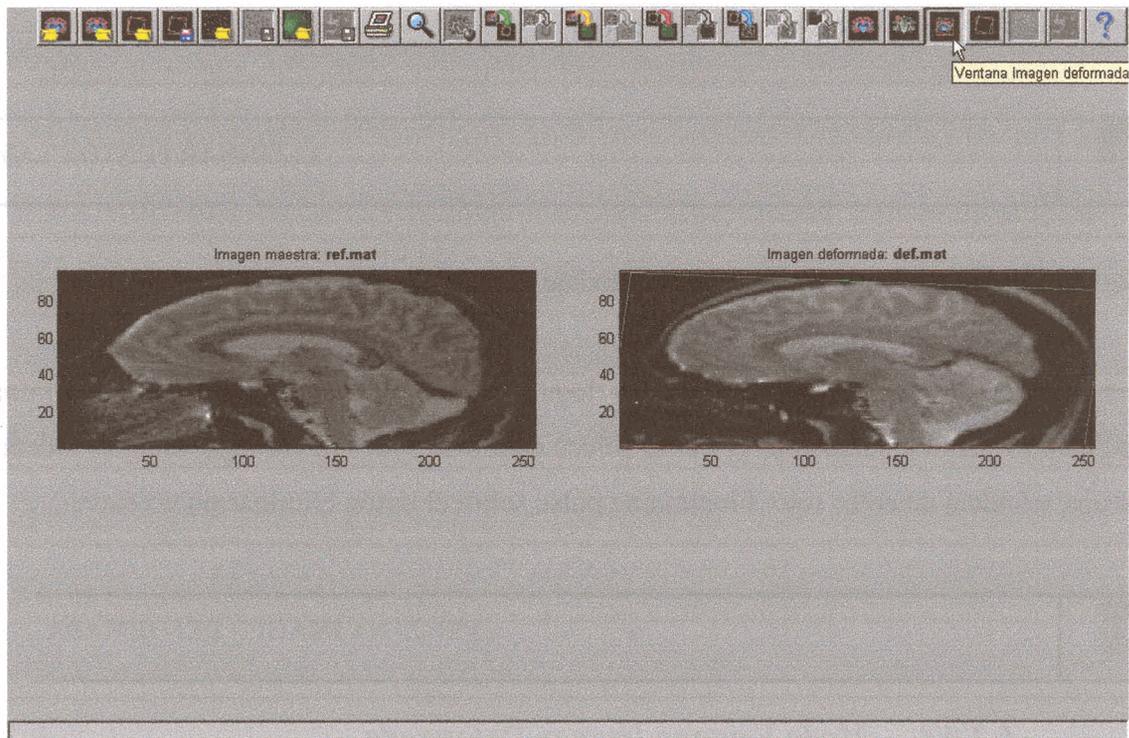


Figura 6.22



VENTANA TRANSFORMACIÓN

6.3.4.4. Modo Ventana Transformación

El modo Ventana Transformación se habilita y activa al [abrir una matriz de transformación](#) o cada vez que se obtenga una a lo largo del programa.

El modo Ventana Transformación, muestra la deformación (mediante un marco verde) que sufriría el marco de la imagen dependiente (marco de origen, en color rojo), al serle aplicada la transformación afín. Véase Figura 6.23.

Al abrir una transformación, se mostrará la transformación normalizada, utilizando como marco de origen el marco normalizado entre $\{-1 \text{ y } 1\}$.

El método abreviado de teclado de Ventana Transformación es `Ctrl + M`.

Desde el momento en el que se haya cargado o creado la transformación afin, y si a su vez existe la imagen dependiente, se habilitarán las herramientas [Obtener Campo denso a partir de Transformación](#) y [Deformar a partir de Transformación](#).

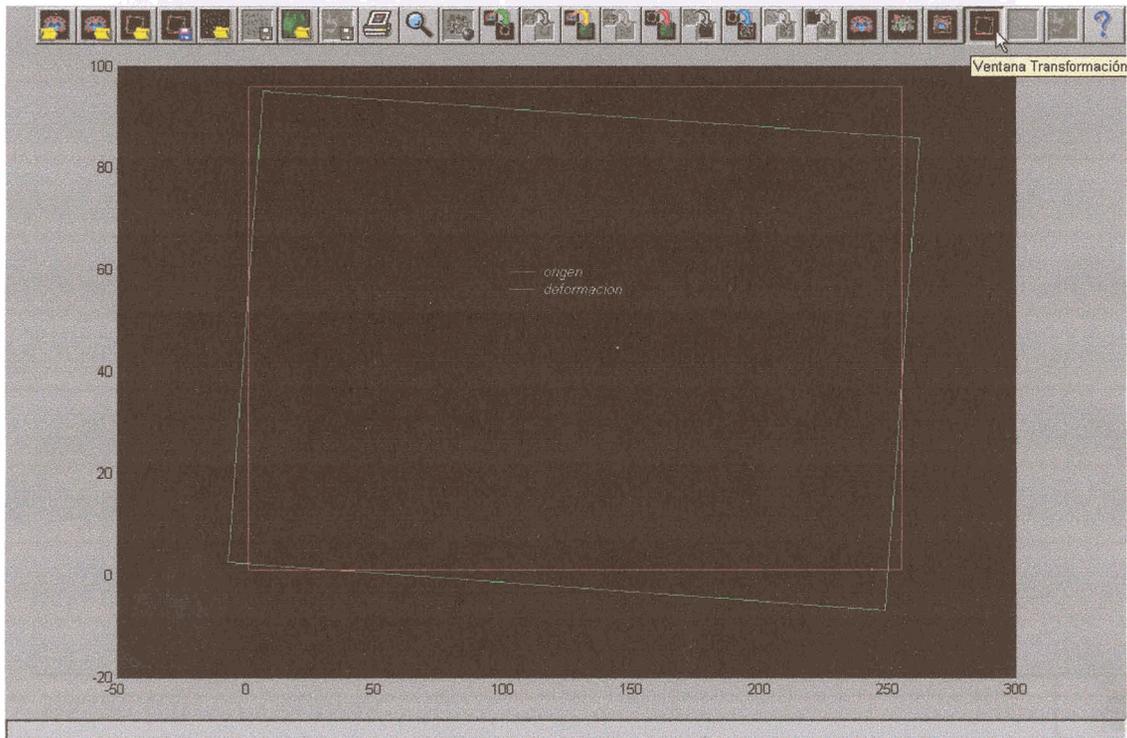


Figura 6.23



VENTANA CAMPO DISPERSO

6.3.4.5. Modo Ventana Campo Disperso

El modo Ventana Campo Disperso se habilita y activa al [abrir un campo vectorial disperso](#), o cada vez que se obtiene uno a lo largo del programa. En este modo se muestra dicho campo vectorial tal y como puede apreciarse en la Figura 6.24.

El método abreviado de teclado de Ventana Transformación es `Ctrl + K`.

Desde el momento en el que se haya cargado o creado el campo vectorial disperso, y si a su vez existe la imagen dependiente, se habilitarán las herramientas [Obtener Campo denso a partir de Campo Disperso](#) y [Deformar a partir de Campo Disperso](#).

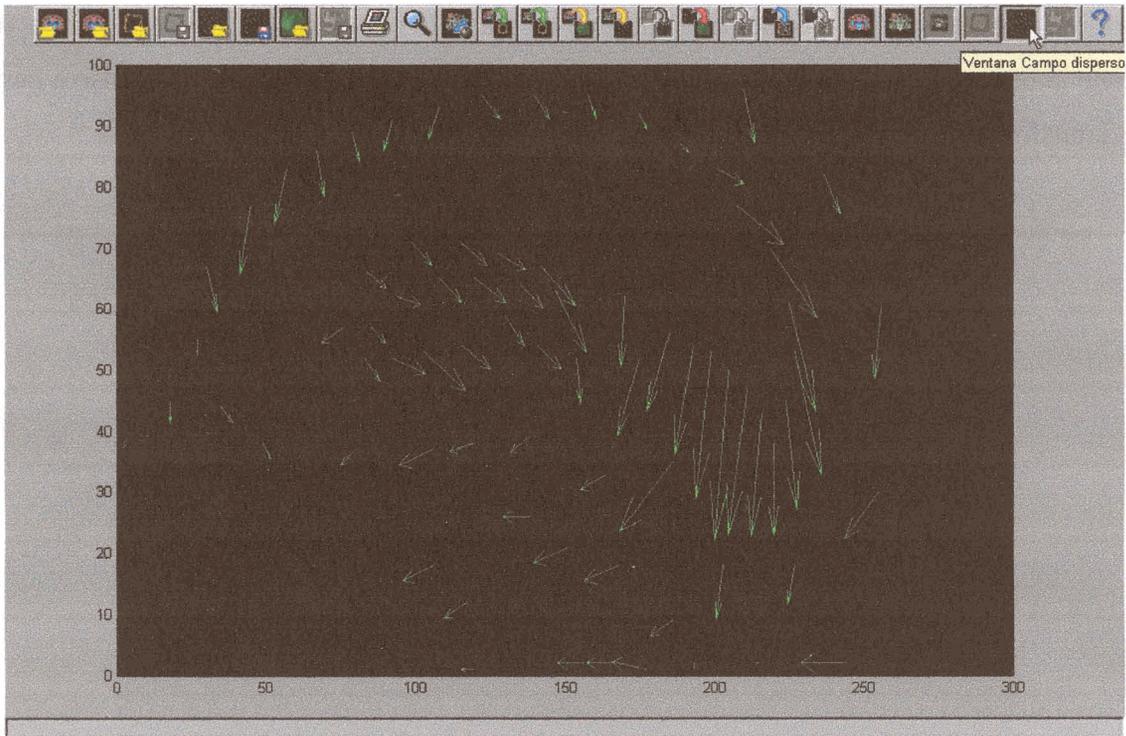


Figura 6.24



VENTANA CAMPO DENSO

6.3.4.6. Modo Ventana Campo Denso

El modo Ventana Campo Denso se habilita y activa al [abrir un campo vectorial denso](#), o cada vez que se obtiene uno a lo largo del programa. En este modo se muestra dicho campo vectorial tal y como puede apreciarse en la Figura 6.25.

El método abreviado de teclado de Ventana Transformación es Ctrl + L.

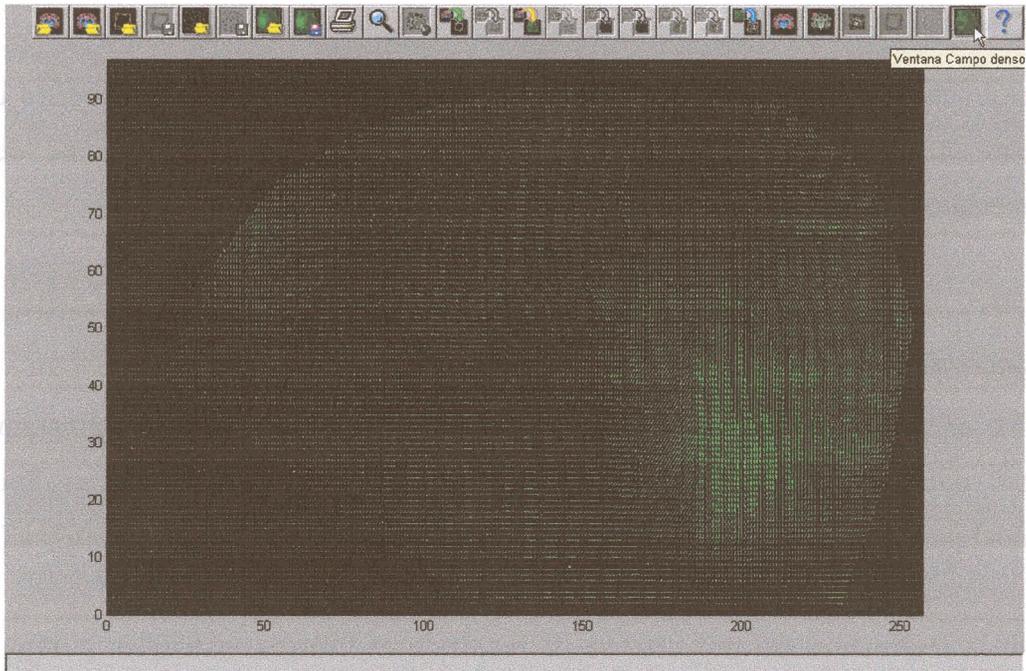


Figura 6.25

Desde el momento en el que se haya cargado o creado el campo vectorial disperso, y si a su vez existe la imagen dependiente, se habilitará la herramienta [Deformar a partir de Campo Denso](#). El uso del [zoom](#) en Ventana Campo Denso puede ser útil para comprender mejor el campo tal y como se representa en la Figura 6.26.

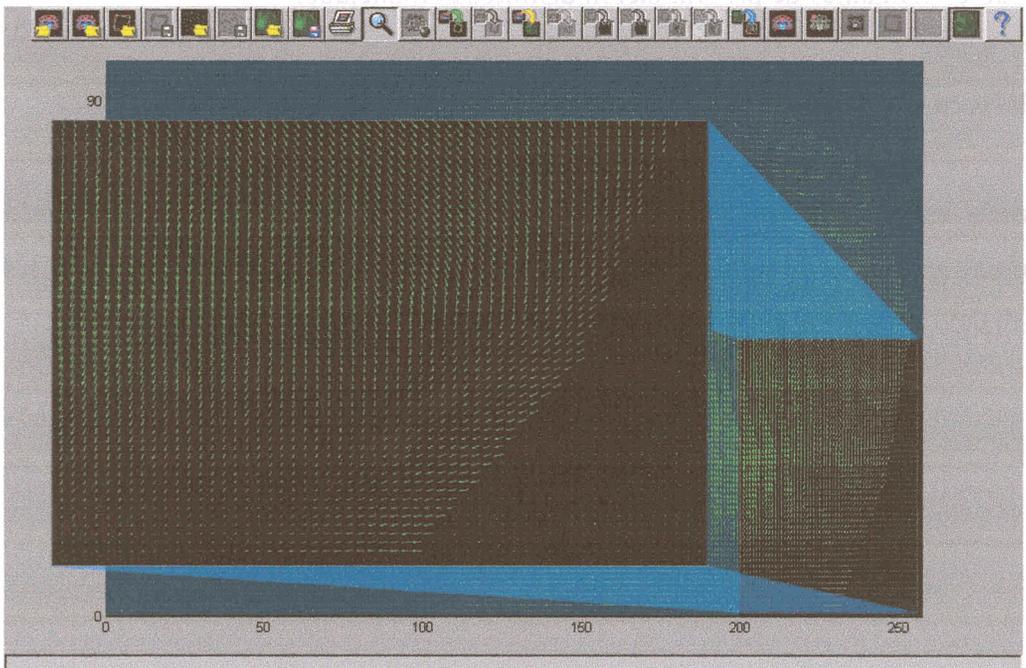


Figura 6.26

7. CONCLUSIONES

En este trabajo se ha efectuado primeramente una exposición acerca del mundo de las imágenes médicas y después se ha explicado el diseño y funcionamiento de una herramienta para el registrado de imágenes médicas en 2 dimensiones.

En el CD que se adjunta se suministra **REGIM**, la interfaz gráfica para el **registrado de imágenes médicas**.

REGIM es una aplicación software desarrollada bajo el entorno de programación Matlab®, dotándola de la capacidad de ser ejecutada en diferentes plataformas, lo que la hace **ideal para ser integrada en un entorno de trabajo orientado a la investigación**.

Durante la realización del proyecto se ha recurrido a diversos materiales y algoritmos normalmente utilizados en el propio entorno de trabajo del software. La interfaz se ha desarrollado orientada a los prototipos, es útil para **validar distintos métodos de registrado de imágenes médicas** y cumple con las características de modularidad deseadas, permitiendo la **incorporación de nuevos algoritmos**, y la **reutilización de los suyos propios en otras interfaces**. Debido a que el *software*, antes de ser implementado, ha sido diseñado, es posible utilizar el diseño para implementar la herramienta de registrado en lenguajes de programación distintos al manejado.

Se satisfacen de esta manera los objetivos que se trazaron en un principio.

Alcanzado este punto, se culmina una pequeña aportación al avance de las nuevas tecnologías aplicadas al campo de la medicina y concretamente al desarrollo del software especializado en el procesado de imágenes biomédicas.

8. LÍNEAS FUTURAS

La creación de esta herramienta para el registrado de imágenes biomédicas no es una cuestión cerrada, sino que al igual que la mayoría del software, sería deseable que fuera actualizado, adaptado y ampliado con nuevas características que permitiesen una notable evolución de la aplicación.

A continuación se realizan algunas sugerencias que pretender bosquejar algunas líneas futuras de actuación.

La aplicación desarrollada está enfocada exclusivamente al registrado de imágenes en dos dimensiones, luego es evidente que el siguiente paso evolutivo ha de ser introducir herramientas para el registrado 3D así como para el registrado 2D-3D.

Además se pueden introducir herramientas que permitan la deformación local de la imagen, para una comparación más interactiva.

Es en cuanto a la visualización de estos nuevos datos donde se habrían de realizar cambios más significativos. Por ejemplo, gestionar una imagen tridimensional (volumen de datos), obligaría a introducir un sistema de navegación entre las diferentes secciones o *slices* del volumen. Esto se podría conseguir gracias a una barra de deslizamiento (*slider*) que el usuario podría manejar indistintamente mediante el desplazamiento del cursor del ratón, o a través de la introducción de valores numéricos desde el teclado.

Sería deseable también permitir diferentes opciones de visualización de los volúmenes de datos, como podrían ser opciones de alternar entre las distintas secciones ortogonales de los datos, esto es, **axial**, **sagital** y **coronal**, e incluir una opción de visualización en perspectiva 3D de las intersecciones de diferentes planos.

Tanto para el campo de deformación disperso como para el campo de deformación denso, se podría utilizar un sistema de representación mediante conos en 3D.

Asimismo, se podrían implementar una serie de mejoras en el *selector manual de puntos*, que hicieran de ésta una herramienta más amigable e intuitiva para el usuario.

Algunas de estas mejoras podrían ser por ejemplo, permitir el reajuste de la posición de un punto (lo que en *Regim* se denomina *editar punto*) mediante el procedimiento de *arrastrar y soltar*, i.e. mantener pulsado el botón del ratón mientras se desplaza el cursor (y con este el punto) por la pantalla.

Se podría introducir también el uso de un menú contextual (menú que se despliega al pulsar con el botón derecho del ratón sobre un objeto) para cada punto clave que permitiese, definir un nombre para cada punto, eliminar el punto o incluso acceder a un cuadro de diálogo de opciones, en el que se podrían variar parámetros tales como el color del punto o su forma, dando de esta manera mayor control al usuario para personalizar la apariencia de la interfaz.

En cuanto a los algoritmos de procesado, lo mejor sería que estos trabajasen siempre con datos 3D y que sencillamente contemplasen los datos 2D como un caso especial y simplificado de datos 3D. Es importante señalar que el paso de dos a tres dimensiones conllevará un incremento tremendamente significativo del coste computacional en los algoritmos, escenificándose éste a través de un notable aumento de los tiempos de cálculo, que ya de por sí son altos en 2D.

Asimismo, las exigencias a la hora de visualizar datos en 3D son mayores. Por todo ello se concluye que será necesario incrementar sensiblemente los requerimientos hardware para desarrollar y ejecutar la futura herramienta.

APÉNDICES

APÉNDICES

I. INFORMACIÓN PARA DESARROLLADORES

En este apéndice se informa extensamente de los detalles del software, con vistas a ser adaptado, expandido o modificado en un futuro.

Con el objetivo de facilitar tales actividades, el software se ha desarrollado de manera modular. Cada herramienta, cada opción, está implementada en un fichero independiente. De esta manera se pueden agregar o quitar opciones con sencillez.

Una de las propiedades de los objetos de Matlab® es el *callback*. En ella se indica la función que el programa tiene que llamar cuando se hace clic mediante el cursor del ratón, sobre ese objeto.

Los objetos son creados en el programa principal, `regim.m`.

En los siguientes puntos se comenta la estructura de **handles** utilizada en el programa `regim`, y se hace una descripción más detallada de los ficheros en los que ha sido empaquetado el software.

1. Estructura

La estructura principal o estructura de *handles* utilizada en el programa `regim` se denomina **estH**.

Esta estructura se utiliza para almacenar tanto los datos y variables manejadas por el programa, como los *handles* manejados por los objetos gráficos de la interfaz.

Matlab® utiliza el concepto de **handle**, que no es más que un **identificador único** para cada objeto gráfico, i.e. desde un botón, a un eje, una opción de menú o un punto en una gráfica.

La estructura **estH** se almacena en la propiedad *userdata* de la figura.

A continuación se muestra la estructura **estH**, con todos sus campos, y posteriormente se comentan los más importantes.

```

estH = struct('hFig', [], ...
...                                     %%% Uimenu %%%
    'hMnArchivo', [], ...
        'hSmnAbrir', [], ...
            'hSmnImgMas', [], ...
            'hSmnImgDep', [], ...
            'hSmnPtMas', [], ...
            'hSmnPtDep', [], ...
            'hSmnXform', [], ...
            'hSmnCDisp', [], ...
            'hSmnCDenso', [], ...
        'hSmnGuardar', [], ...
            'hSmnGPtMas', [], ...
            'hSmnGPtDep', [], ...
            'hSmnGXform', [], ...
            'hSmnGImgDef', [], ...
            'hSmnGCDisp', [], ...
            'hSmnGCDenso', [], ...
        'hSmnCerrar', [], ...
            'hSmnCerrarImgMas', [], ...
            'hSmnCerrarImgDep', [], ...
            'hSmnCerrarPtMas', [], ...
            'hSmnCerrarPtDep', [], ...
            'hSmnCerrarXform', [], ...
            'hSmnCerrarCDisp', [], ...
            'hSmnCerrarCDenso', [], ...
        'hSmnImpr', [], ...
        'hSmnIdiom', [], ...
            'hSmnEng', [], ...
            'hSmnEsp', [], ...
        'hSmnSalir', [], ...

```

```

'hMnVer', [], ...
    'hSmnBH', [], ...
    'hSmnBE', [], ...
    'hSmnMas', [], ...
    'hSmnDep', [], ...
    'hSmnEjes', [], ...
    'hSmnZoom', [], ...
    'hSmnMColor', [], ...
    'hSmnAv', [], ...
'hMnH', [], ...
    'hSmnAFino', [], ...
        'hSmnHR_Img', [], ...
        'hSmnHR_Pt', [], ...
        'hSmnHnR_Img', [], ...
        'hSmnHnR_Pt', [], ...
    'hSmn_Xform', [], ...
    'hSmn_CDisp', [], ...
    'hSmnH_Xform', [], ...
    'hSmnH_CDisp', [], ...
    'hSmnH_CDenso', [], ...
'hMnVentana', [], ...
    'hSmnV_Img', [], ...
    'hSmnV_Pt', [], ...
    'hSmnV_ImgDef', [], ...
    'hSmnV_Xform', [], ...
    'hSmnV_CDisp', [], ...
    'hSmnV_CDenso', [], ...
'hMnAyuda', [], ...
    'hSmnCont', [], ...
    'hSmnAcerca', [], ...
...
                                     %%% Axes %%%
'hAxeBH', [], 'hBotonesBH1', [], ...
'hAxeMas', [], 'hAxeDep', [], ...
'hAxeXform', [], 'hAxeAux', [], ...

```

```

...                               %%% Uicontrol   %%%
'hFrmMas', [], 'hFrmDep', [], ...
'hTxtMas', [], 'hTxtTitleMas', [], ...
'hTxtDep', [], 'hTxtTitleDep', [], ...
'hPbtMasAdd', [], 'hPbtDepAdd', [], ...
'hPbtMasSup', [], 'hPbtDepSup', [], ...
'hPbtMasEdt', [], 'hPbtDepEdt', [], ...
'hPbtMasAbrir', [], 'hPbtDepAbrir', [], ...
'hPbtMasGuardar', [], 'hPbtDepGuardar', [], ...
'hPbtMasCerrar', [], 'hPbtDepCerrar', [], ...
'hLbxMas', [], 'hLbxDep', [], ...
'hEdtBE1', [], ...
...                               %%% Otros   %%%
'BotonesCData', [], ...
'estTickMas', [], 'estTickDep', [], ...
'estTickAux', [], ...
'hBoton', [], ...
'estPtCMas', [], 'estPtCDep', [], ...
'estXform', [], 'estCDisp', [], 'estCDenso', [], ...
'hImagMas', [], 'hImagDep', [], 'hImagDef', [], ...
'hPlotPtCMas', [], 'hPlotPtCDep', [], ...
'hPlotImgDef', [], ...
'hPlotXform', [], ...
'hQuivCDisp', [], ...
'hQuivCDenso', [], ...
'hLegenda', []]);

```

En la primera sección (uimenu) se definen aquellos campos que almacenaran los *handles* de todas las opciones de menú. En la siguiente sección (axes), se almacenan los *handles* de todos los ejes utilizados en la aplicación desarrollada.

La sección posterior (uicontrol), almacena los *handles* concernientes a los objetos de control tales como botones, cuadros de texto, cajas de texto editables, cajas de selección, etc.

Por ultimo en la sección final (otros), se definen campos que almacenan variables de muy distintos tipos, entre las que caben destacar:

| | |
|-------------|---|
| 'estPtCMas' | almacena la lista de puntos clave maestros. |
| 'estPtCDep' | almacena la lista de puntos clave dependientes. |
| 'estXform' | almacena la matriz de transformación. |
| 'estCDisp' | almacena el campo de deformación disperso. |
| 'estCDenso' | almacena el campo de deformación denso. |

2. Ficheros

Tal y como se había hecho en el capítulo 5, los ficheros se pueden clasificar en ficheros de datos, ficheros de procesado y ficheros de interfaz. A continuación se enumeran dichos ficheros y en aquellos casos más relevantes se identifica su función, señalándose también los parámetros de entrada y salida.

2.1. Ficheros de Datos

- abrirImagen.m
- abrirPtC.m
- abrirXform.m
- abrirCDisp.m
- abrirCDenso.m
- guardarPtC.m
- guardarXform.m
- guardarImgDef.m
- guardarCDisp.m
- guardarCDenso.m
- cerrarImagen.m
- cerrarPtC.m
- cerrarXform.m
- cerrarCDisp.m
- cerrarCDenso.m

2.2. Ficheros de Procesado

- `ajusteFinoPtC.m`

ajusteFinoPtC(estH, estIdiom);

Función que realiza el sintonizado fino de puntos clave.

La función evalúa si existen los puntos clave maestros y dependientes, así como que existan la imagen maestra y la dependiente. En caso de éxito, la nueva lista de puntos clave maestros se almacena en el campo 'estPtCDep' de la estructura `estH`. Donde:

'estH', es la estructura donde guardamos todos los *handles*. (prop. 'UserData' de la figura).

'estIdiom' es la estructura donde guardamos todas las etiquetas correspondientes al idioma.

- `deformador.m`

[IMAGEN_DEF] = deformador(T, IMAGEN);

Deforma la imagen dependiente (IMAGEN), en función de la matriz de transformación T, y devuelve la imagen deformada en IMAGEN_DEF. Donde:

'T' Matriz de transformación afín en coordenadas homogéneas.

'IMAGEN' Es la imagen que va a ser deformada (matriz contenida en la propiedad 'CData' de la imagen).

- `deformadorCampoDenso.m`

[IMAGEN_DEF] = deformadorCampoDenso(pos, dif, IMAGEN);

Deforma la imagen dependiente (IMAGEN) en función del campo denso de deformación denso (pos, dif), y devuelve la imagen deformada en IMAGEN_DEF. Donde:

'pos' matriz que indica la posición del origen de los vectores del campo denso.

'dif' matriz que indica la dirección y magnitud de los vectores del campo denso.

'IMAGEN' Es la imagen que va a ser deformada (matriz contenida en la propiedad 'CData' de la imagen).

- `regRigImagen.m`

regRigImagen(estH, estIdiom, colorOrg, colorDef);

Registrado rígido de Imagen, crea una transformación afín. Internamente evalúa la existencia de las imágenes maestra y dependiente y si la función tiene éxito almacena la transformación afín en el campo 'estXform' de **estH**. Donde:

'estH', estructura donde guardamos todos los *handles*. (prop. 'UserData' de la figura).

'estIdiom' estructura donde guardamos todas las etiquetas correspondientes al idioma.

'colorOrg' *string* que indica el color del marco de origen(sin deformar).

'colorDef' *string* que indica el color del marco de la deformación.

- `regRigPtC.m`

regRigPtC(estH, estIdiom, colorOrg, colorDef);

Registrado rígido de puntos clave, crea una transformación afín. Internamente evalúa la existencia de las listas de puntos clave maestra y dependiente y si la función tiene éxito almacena la transformación afín en el campo 'estXform' de **estH**.

- `regNoRigImagen.m`

regNoRigImagen(estH, estIdiom, colorflecha);

Realiza el registrado no rígido entre la imagen maestra y la dependiente, generando un campo de deformación denso. Internamente evalúa la existencia de las imágenes maestra y dependiente y si la función tiene éxito almacena el campo denso en el campo 'estCDenso' de **estH**. Donde:

'estH', estructura donde guardamos todos los *handles*. (prop. 'UserData' de la figura).

'estIdiom' estructura donde guardamos todas las etiquetas correspondientes al idioma.

'colorflecha', *string* que indica el color de las flechas del campo.

- `regNoRigPtC.m`

regNoRigPtC(estH, estIdiom, colorflecha);

Realiza el registrado no rígido entre los puntos clave maestros y dependientes, generando un campo de deformación disperso.

Internamente evalúa la existencia de las listas de puntos clave maestros y dependientes, y si la función tiene éxito, almacena el campo disperso en el campo 'estCDisp' de estH.

- `registradorDiferencias.m`

[POS,DIF] = registradorDiferencias(PTCMAS, PTCDEP);

Registrador de diferencias entre los puntos clave maestros y dependientes, retorna 2 matrices.

La matriz de posición (POS) de los puntos del campo disperso.

La matriz de diferencias(DIF) que indica la dirección y magnitud de los vectores de dicho campo. Donde:

'PTCMAS' Matriz de Nx2, que indica las posiciones de los N puntos clave maestros.

'PTCDEP' Matriz de Nx2, que indica las posiciones de los N puntos clave dependientes.

- `registradorMSE.m`

[estXform] = registradorMSE(PTCMAS, PTCDEP, TAMyx),

Registrador rígido de puntos clave, que retorna la transformación afín, en una estructura del tipo *tform*, en la que se incluye también la matriz de normalización. Donde:

'PTCMAS' Matriz de Nx2, que indica las posiciones de los N puntos clave maestros.

'PTCDEP' Matriz de Nx2, que indica las posiciones de los N puntos clave dependientes.

'TAMyx' es la resolución de la imagen a deformar.

- `interpolador.m`

[POS, DIF] = interpolador(pos, dif, tam);

Interpola el campo disperso para obtener el campo denso. Se retornan 2 matrices:

La matriz de POS, que indica la posición del origen de los vectores del campo denso.

La matriz de diferencias DIF, que indica la dirección y magnitud de los vectores del campo denso. Donde:

'pos' Matriz de Nx3, que indica la posición del origen de los N vectores del campo disperso.

'dif' Matriz de Nx3, que indica la dirección y magnitud de los N vectores del campo disperso.

'tam' es la resolución de la imagen a deformar.

- obtenerCDenso.m

obtenerCDenso(estH, estIdiom, tipo, colorflecha);

Obtiene el campo denso a partir de la transformación afin, o del campo disperso. Donde:

'estH', estructura donde guardamos todos los *handles*. (prop. 'UserData' de la figura).

'estIdiom' estructura donde guardamos todas las etiquetas correspondientes al idioma.

'tipo' puede ser '*xform*', '*cdisp*'.

'colorflecha', *string* que indica el color de las flechas del campo.

- private\registradorONR.m

[POS, DIF] = registradorONR(MAS, DEP, levels);

Registrador no rígido de imagen, retorna 2 matrices del Campo denso.

La matriz de POS, que indica la posición del origen de los vectores del campo denso.

La matriz de diferencias DIF, que indica la dirección y magnitud de los vectores del campo denso. Donde:

'MAS' Imagen Maestra.

'DEP' Imagen Dependiente.

'levels' Niveles de la pirámide.

- private\registradorOR.m

[estXform] = registradorOR(MAS, DEP);

Registrador rígido de imagen, retorna la matriz de transformación afin en una estructura del tipo *tform* llamada '*estXform*', en la que se incluye también la matriz de normalización. Donde:

'MAS' Imagen Maestra.

'DEP' Imagen Dependiente.

'levels' Niveles de la pirámide.

2.3. Ficheros de interfaz

- `regim.m`

fig = regim(),

Programa para el registrado de imágenes.

La función crea la figura de la interfaz gráfica de usuario(GUI) del programa.

Todos los 'Callback' producidos por los *uicontrol*, los *uimenu* y los *Axes* de la figura, llaman a la función `gesCB.m`, que se encargará de gestionarlos.

- `gesCB.m`

gesCB()

Gestiona todos los *Callback* producidos desde la figura de la interfaz creada por `regim.m`.

Asimismo se encarga de cambiar las etiquetas correspondientes a cada idioma en caso necesario.

- `idioma.m`

[estIdiom] = idioma(idiom);

Retorna en la estructura **`estIdiom`**, que almacena todas las etiquetas de la interfaz en el idioma seleccionado.

donde:

'idiom' puede ser 'English' o 'Español'.

- `mostrarImagen.m`
- `mostrarPtC.m`
- `dibujaXform.m`
- `dibujaImgDef.m`
- `dibujaCDisp.m`
- `dibujaCDenso.m`
- `deseleccionarSmnV.m`
- `pulsadoListbox.m`
- `redimensionaEjes.m`
- `redimensiona_figura.m`

- reescribeTitle.m
- dlgGuardar.m
- eliminaPto.m
- enableBoton.m
- eval_mi2.m
- gesCursor.m
- gesPbtAddEdt.m
- gesPbtSup.m
- resetearPlots.m
- getTick.m
- resetearPtC.m
- algunaVentana.m
- selectBoton.m
- axes2pix.m
- selectLevels.m
- borrarArandela.m
- startup.m
- borrarTick.m
- cellString.m
- habilitar.m
- ventanaCDisp.m
- ventanaImagen.m
- ventanaCDenso.m
- ventanaImgDef.m
- ventanaPtC.m
- ventanaXform.m
- imprimir.m
- clickIzquierdo.m
- listBox.m
- verBarraEstado.m
- comprobarPbt.m

- verBarraHerramientas.m
- continuar.m
- mi_cp2tform.m
- verDatos.m
- controlSmnArchivo.m
- verEjes.m
- controlSmnHerramientas.m
- verZoom.m
- presentacion.m

3. Acoplamiento de un nuevo módulo

En el caso de querer añadir un nuevo módulo a la interfaz, en primer lugar se habrá de definir una nueva opción de menú dentro del submenú de herramientas.

Para ello se introducirá el código que define una nueva opción de menú en la función `regim.m`.

Concretamente en la sección de definición de los menús, se añadirá lo siguiente:

```
% Submenu Herramientas/ nombre del nuevo modulo.
h2 = uimenu('Parent',h1, ...
    'Label',estIdiom.sSmn_identificador_del_nuevo_módulo, ...
    'Callback','gesCB', ...
    'Separator','on', ...
    'Enable','off', ...'on', ...
    'Tag','Smn_identificador_del_nuevo_módulo');
estH.hSmn_identificador_del_nuevo_módulo = h2;
```

El campo de la estructura de *handles* (**estH**), que almacenará el *handle* del nuevo módulo es `'hSmn_identificador_del_nuevo_módulo'`.

En la función `idioma.m`, en la estructura de idioma (**estIdiom**) será necesario añadir el nombre de la nueva función en los idiomas correspondientes. Para ello se añade el campo `'sSmn_identificador_del_nuevo_módulo'`.

Por último en la función que gestiona todas las llamadas o *callbacks* que hace cada objeto de la interfaz, `gesCB.m` es necesario añadir dentro del múltiple `switch- case`, en la sección que se corresponde con las opciones del submenú herramientas, lo siguiente:

```
case estH.hSmn_identificador_del_nuevo_módulo,  
    [variables_de_salida]=nuevo_modulo(estH, estIdiom, variables_de_entrada);
```

donde *nuevo_modulo* es el nombre de la función que contiene el nuevo módulo añadido, *variables_de_entrada* representa todas las variables de entrada necesarias y *variables_de_salida* representa todas las variables de salida necesarias.

Y en la subfunción interna de esta llamada `Cambiaidioma` añadir lo siguiente:

```
set(estH.hSmn_identificador_del_nuevo_módulo,'Label',...  
estIdiom.sSmn_identificador_del_nuevo_módulo);
```

II. CONTENIDO DEL CD

La herramienta software para el registrado de imágenes médicas (**REGIM**) se adjunta en el CD suministrado con este documento, así como toda la documentación de ayuda al usuario necesaria. El CD consta de los siguientes directorios:



En el directorio *Regim* se encuentran los archivos principales de la aplicación, un archivo llamado *leeme.txt* en el que se explica la manera de instalar y ejecutar el programa, así como un archivo llamado *ayuda.htm* que muestra en página web el manual de usuario.

La carpeta *Ayuda_archivos*, contiene elementos gráficos necesarios para la correcta presentación de *ayuda.htm*.

La carpeta *Campo denso*, contiene ejemplos de Campos densos guardados.

La carpeta *Campo disperso*, contiene ejemplos de Campos dispersos guardados.

En la carpeta *iconos* se almacenan las imágenes que se muestran en los iconos de la barra de herramientas.

En la carpeta *Imagenes*, se almacenan algunas imágenes de ejemplo con las que empezar a hacer operaciones de registrado.

La carpeta *private* contiene archivos fundamentales para la aplicación relacionados con los algoritmos para el registrado rígido y no rígido de imágenes.

La carpeta *Puntos Clave*, contiene ejemplos de Listas de puntos clave guardados.

La carpeta *tensorToolbox* contiene archivos que conforman una librería o *toolbox* propia, necesaria para la correcta ejecución de esta aplicación.

La carpeta *Transformacion*, contiene ejemplos de Transformaciones afines guardadas.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

1. *Graphics and GUI's with Matlab.*
Patrick Marchand.
Edt. CRC Press. 1999.
2. *Digital Image Processing.*
William k. Pratt.
Edt. John Wiley & Sons. 1991.
3. *Fundamentals of three-Dimensional computers graphics.*
Alan Watt .
Edt. Addison - Wesley. 1990.
4. *MATLAB.*
The Language of Technical Computing
Using MATLAB Graphics, Version 6.
Edt. The Math Works Inc. 2000.
5. *Image Processing Toolbox.*
For use with MATLAB.
User's Guide, Version 3
Edt. The Math Works Inc. 2001
6. *Diagnóstico por imagen.*
César S. Pedrosa.
Edt. Interamericana/McGraw Hill. 1992.
7. *Handbook of Medical Imaging*
Processing and Analysis.
Isaac N. Bankman.
Edt. Academic Press. 2000.

8. *Mastering MATLAB6.*
A Comprehensive Tutorial and Reference.
Duane C. Hanselman & Bruce Littlefield.
Edt. Prentice Hall. 2001.
9. *The Visualization Toolkit.*
An Object-Oriented Approach to 3D Graphics.
Will Schröder, Ken Martin, Bill Lorensen.
Edt. Prentice Hall. 1997.

Artículos consultados.

10. *A survey of Medical Image Registration.*
J. B. Antoine Maintz and Max. A. Viergever.
Medical Image Analysis, volumen 2, numero 1, paginas 1-37, 1998.
11. *Numerical Recipes in C : The Art of Scientific Computing.*
William H. Press, Brian P. Flannery, Saul A. Teukolsky, William T. Vetterling
Publisher: Cambridge University Press; ISBN: 0521431085;
2nd edition (January 1993)
12. *Nonrigid registration using regularized matching weighted by local structure.*
Eduardo Suárez, Carl-Fredrik Westin, Eduardo Rovaris, and Juan Ruiz-Alzola.
In Proceedings of the Fifth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI '02), volume 2 of Lecture Notes in Computer Science, pages 581-589, Tokyo, Japan, September 2002.
13. *2D-3D Rigid Registration of X-Ray Fluoroscopy and CT Images Using Mutual Information and Sparsely Sampled Histogram Estimators.*
L. Zöllei, E. Grimson, A. Norbash, M. Wells
Artificial Intelligence Laboratory, MIT. Cambridge, MA.
Dept. of Radiology, Brigham and Women's Hospital, Boston. MA.

14. *Plataforma abierta para el procesamiento y gestión de imágenes biomédicas.*

Carlos Platero. Kristof Verbiest. Alejandro Ubeda. Angeles Trillo. Jaime Gonsalvez. Javier Bartolomé.

15. *La Base de la Interfaz Grafica: Usuario Computadora.*

Mario Flores Cabrera.

Direcciones web consultadas vigentes a 15 de Noviembre de 2002.

16. <http://www.msdn.microsoft.com/library/default.asp?url=/library/en-us/vcmfc98/html/mfchm.asp>

17. <http://www.gtk.org>

18. CISG. Computational Imaging Science Group.

King's College. London.

<http://www.image-registration.com/>

19. Analyze 4.0.

<http://www.analyzedirect.com/Analyze4.0/enhancements.asp>

20. 3D Slicer.

MIT Artificial Intelligence lab and Surgical Planning lab at Brigham & Women's Hospital.

Harvard Medical School.

<http://www.slicer.org>

21. Amira. Visualization and 3D-Reconstruction.

<http://www.amiravis.com/usersguide/index.html>

22. SPM. Statistical Parametric Mapping.

<http://www.fil.ion.ucl.ac.uk/spm/>

23. MNI_AutoReg Software Package.

http://www.bic.mni.mcgill.ca/users/louis/MNI_AUTOREG_home/readme/readme.html

24. AIR. Automated Image registration.

<http://bishopw.loni.ucla.edu/AIR5/index.html>

25. Project on Image Guided Surgery.

http://splweb.bwh.harvard.edu:8000/pages/papers/image_guided_surg3/surgical_navigation.html#regis

26. http://www.dpi.inpe.br/spring/usuario_spa/c_filtro.htm

27. <http://www.bodyscan.md/>

**PLIEGO DE
CONDICIONES**

PLIEGO DE CONDICIONES

El sistema para ejecutar la aplicación Regim (Interfaz para el registrado de imágenes biomédicas) consta de:

● SOFTWARE

- Sistema Operativo Windows 9x/NT, Linux, Solaris
- Matlab® 6.0 y los Toolbox de Image Processing 3.0, Optimization 1.1 y Symbolic Math 2.1.2.

● HARDWARE

- *Configuración mínima necesaria*
 - Procesador a 333 MHz
 - 128 MB de memoria RAM
 - Tarjeta de vídeo de 8 MB, con una resolución de 800 x 600
- *Configuración recomendada.*
 - Procesador a 800 MHz o superior
 - 256 MB de memoria RAM, o superior
 - Tarjeta de video de 32 MB, con una resolución de 800 x 600, o superior

PRESUPUESTO

PRESUPUESTO

Este presupuesto se ha realizado según lo establecido en la Junta General del COITT (Colegio Oficial de Ingenieros Técnicos de Telecomunicación) para el año 2002.

1. Trabajo tarifado por tiempo establecido

Se establece la siguiente fórmula para definir los honorarios a presupuestar por el trabajo realizado:

$$H = C * (58 * H_n + 63 * H_e) \text{ €}$$

Donde:

H_n son las horas normales. Si se tiene en cuenta que la duración del proyecto ha sido de 19 meses (descontados fines de semana y días festivos un total de 416 días) a razón de 5 horas diarias, resulta $H_n = 2080$ horas.

H_e son las horas especiales. Dado que siempre se ha trabajado en horario normal resulta $H_e = 0$ horas.

C es un coeficiente de reducción en función del número de horas trabajadas. E COITT establece que para más de 1080 horas $C = 0,4$.

Por lo tanto,

$$H = 0,4 * (58 * 2080) = 48.256 \text{ €}$$

El coste de la mano de obra asciende a **cuarenta y ocho mil doscientos cincuenta y seis euros**.

2. Coste de las licencias del software

El desglose de precios del software requerido para la realización del proyecto estimando una amortización a 3 años, atiende a la siguiente tabla:

| Concepto | Precio (€) | Amortización a 3 años (€) |
|------------------------------|---------------|---------------------------|
| Windows NT Workstation | 240,40 | 80,13 |
| Microsoft Office 97 | 90,15 | 30,05 |
| Matlab® 6.0. | 70,00 | 23,33 |
| Image Processing Toolbox 3.0 | 30,00 | 10,00 |
| Optimization Toolbox 1.1 | 30,00 | 10,00 |
| Symbolic Math Toolbox 2.1.2. | 30,00 | 10,00 |
| S. O. Windows 98 S.E. | 0,00 | 0,00 |
| Internet explorer 5.0 | 0,00 | 0,00 |
| Total | 490.55 | 163.51 |

El coste por las licencias del software asciende a **ciento sesenta y tres euros con cincuenta y un céntimos**.

3. Coste del hardware

Se presenta a continuación el precio del material hardware, estimando una amortización a 3 años:

| Concepto | Precio (€) | Amortización a 3 años (€) |
|--|-----------------|---------------------------|
| Ordenador Personal Pentium III MMX a 800 MHz. | 901,52 | 300,51 |
| Impresora Epson Stylus Color 760 | 901,52 | 300,51 |
| Total | 1.803,04 | 601,02 |

El coste completo del hardware asciende a **seiscientos un euro con dos céntimos**.

4. Derechos de visado del COITT

Los gastos de visado del COITT se obtienen de la siguiente fórmula:

$$V = 0,007 * P$$

Donde, P es el presupuesto de ejecución del material. Para el cálculo de P se suman las cantidades obtenidas en los puntos 1, 2 y 3.

| | Coste (€) |
|--------------------------|------------------|
| Coste de la mano de obra | 48.256,00 |
| Coste del software | 163,51 |
| Coste del hardware | 601,02 |
| Total | 49.020,53 |

Por lo tanto,

$$V = 0,007 * 49.020,53 = \mathbf{343,14 \text{ €}}$$

Los derechos de visado ascienden a **trescientos cuarenta y tres euros con catorce céntimos**.

5. Gastos de administración y envío

Los gastos de administración y envío son fijos y para una tramitación normal (sin suplemento por urgencia ni mensajería) se estipulan en **9 €, nueve euros**.

6. Coste total del proyecto

El coste total del proyecto fin de carrera realizado se desglosa en la siguiente tabla:

| | Coste Total (€) |
|---|------------------------|
| Honorarios | 48.256,00 |
| Licencias software amortizadas a 3 años | 163,51 |
| Coste del hardware amortizado a 3 años | 601,02 |
| Derechos de visado del COITT | 343,14 |
| Gastos de administración y envío | 9,00 |
| Subtotal sin IGIC | 49372,67 |
| IGIC (5%) | 2468,63 |
| Total | 51.841,30 |

El presupuesto total, incluidos impuestos, asciende a la cantidad de **cincuenta y un mil ochocientos cuarenta y un euros con treinta céntimos.**

Las Palmas de Gran Canaria, a 27 de noviembre de 2002.

El Autor del Proyecto

Fdo. José Carlos Rodríguez Pérez.