

### 3. Sistema de adquisición de imágenes

#### 3.1. Introducción

Tal y como se ha explicado en capítulos anteriores, la obtención del par estéreo se suele realizar con dispositivos estereoscópicos.

En este proyecto se ha optado por un sistema de adquisición formado por (véase fig. 21):

- un robot posicionador
- una cámara digital USB
- un equipo informático (PC)

El robot posicionador dispone de una plataforma giratoria y un brazo móvil en cuyo extremo se coloca la cámara. Tanto la cámara como el robot están conectados al ordenador de manera independiente. El ordenador ejecuta el software encargado de configurar y controlar la cámara digital y el robot posicionador. La comunicación con la cámara digital se realiza a través del puerto USB, el robot posicionador en cambio se controla a través del puerto paralelo. Véase el esquema de la fig. 22.

El proceso de adquisición consiste en grabar un giro de 360° con el fin de obtener las vistas necesarias por el tratamiento posterior, para ello el software se encarga de hacer girar la plataforma del robot y de adquirir las imágenes del objeto a través de la cámara digital.

Una vez adquiridas las imágenes el software las procesa y las almacena en un fichero AVI.

#### 3.2. El robot posicionador

##### 3.2.1. Estructura

El robot posicionador está formado por dos bases:

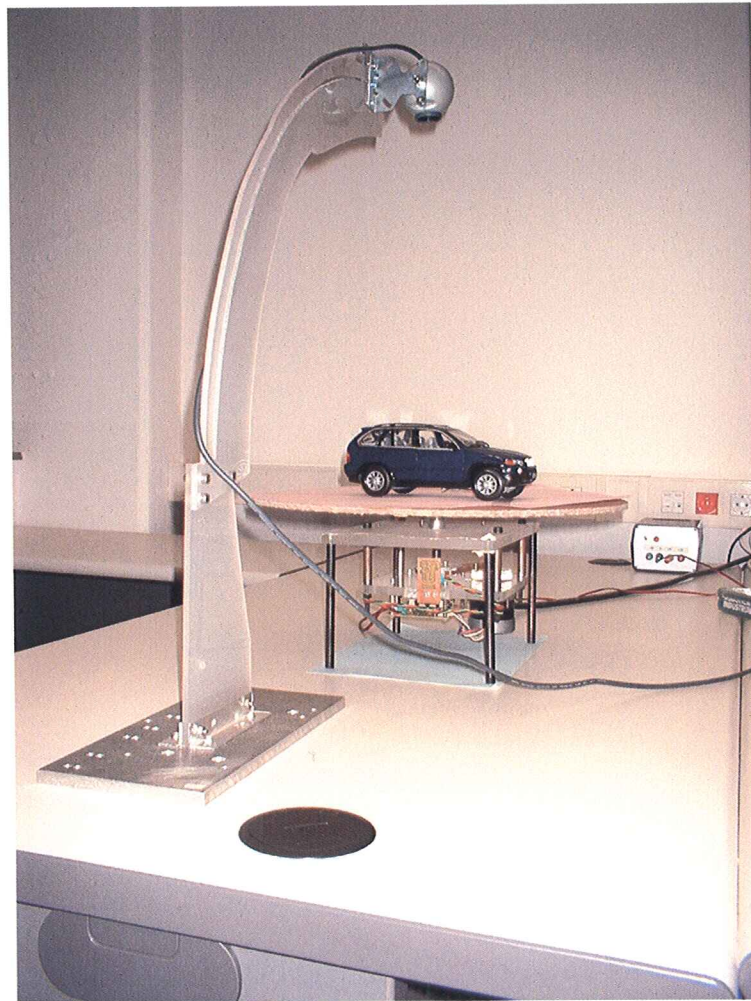


Figura 20: Robot posicionador

- una base superior encargada de mantener la estructura fijada a la superficie mediante soportes atornillados. En ella están atornillados los soportes de la base inferior. Además se encarga de guiar el cilindro posicionador de la plataforma mediante un círculo de diámetro apropiado.
- una base inferior encargada de ubicar el motor, la interfaz electrónica y el sistema de correas. Además hace de base para guiar y rotar sobre su propio eje el cilindro posicionador.

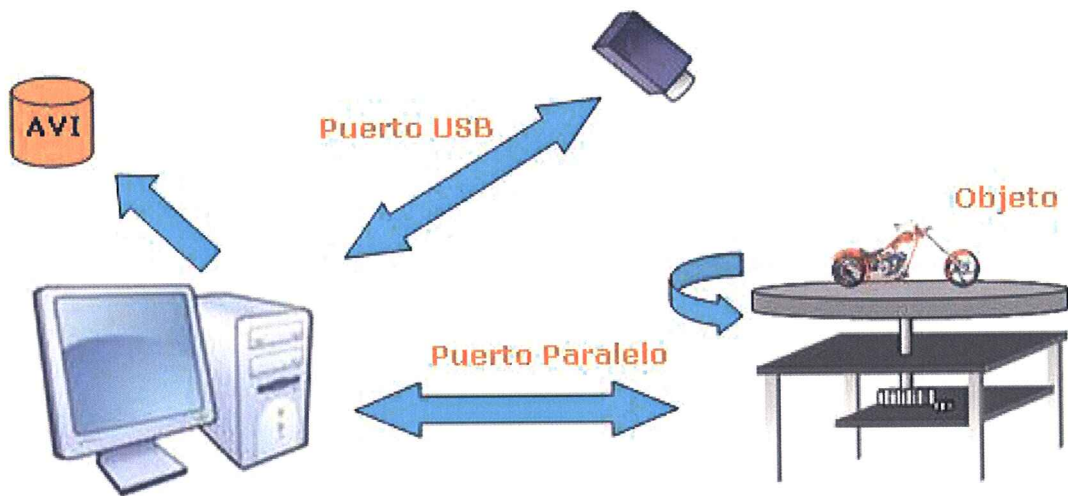


Figura 21: Sistema de adquisición utilizado

### 3.2.2. Elementos mecánicos y eléctricos

**Motor paso a paso** El robot posicionador dispone de dos motores tipo **paso a paso** para mover la plataforma y el brazo móvil. También conocidos como "stepper motor", son ideales para la construcción de mecanismos en los cuales se requiere movimientos muy precisos.

La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Básicamente estos motores están constituidos normalmente por un rotor sobre el que van aplicados distintos imanes permanentes y por un cierto número de bobinas excitadoras bobinadas en su estator. La excitación de las bobinas deberá ser externamente manejada por un controlador [12].

Para el posicionado se dispone de un motor TEC con cuatro bobinas (fig. 23) lo que implica que serán necesarias cuatro señales para excitar los diferentes estados del motor. Se ha optado por la utilización de puerto paralelo para el control del hardware dada la poca cantidad de señales necesarias para controlar el motor.

**Puerto paralelo** Un puerto paralelo es una interfaz entre un ordenador y un periférico cuya principal característica es que los bits de datos viajan juntos enviando



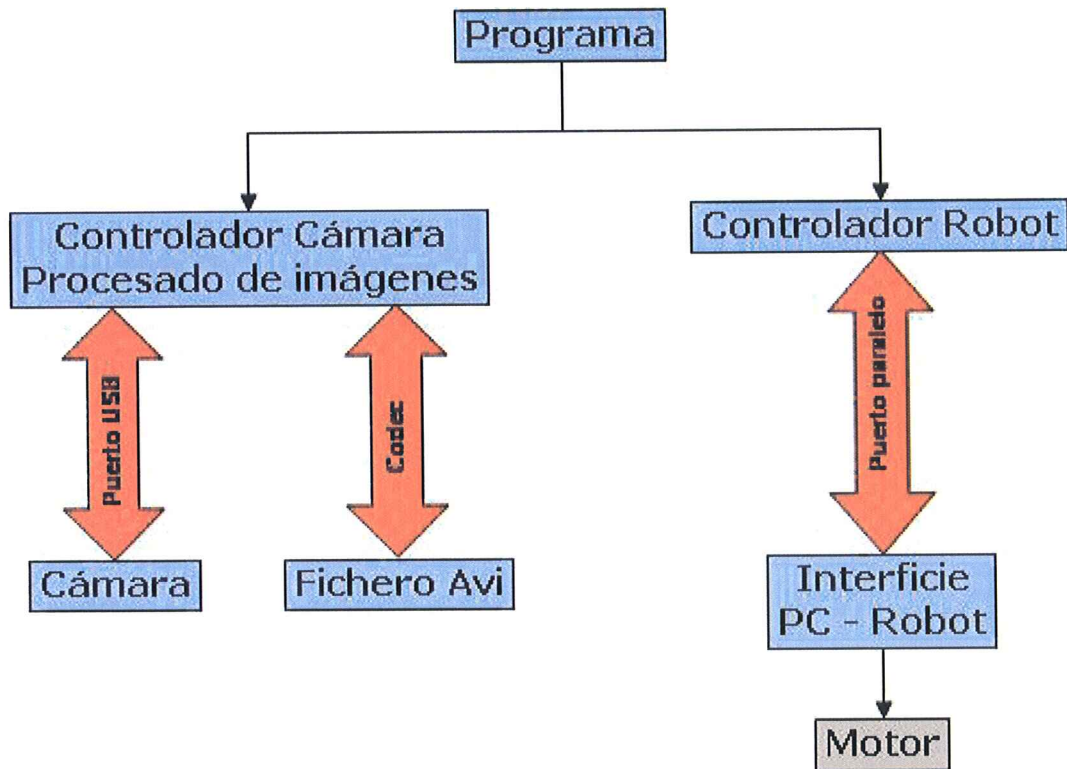


Figura 22: Esquema del sistema de adquisición

un byte completo o más a la vez. Se utiliza generalmente para manejar impresoras, sin embargo, dado que este puerto tiene un conjunto de entradas y salidas digitales, se puede emplear para el control de dispositivos.

En un PC pueden instalarse varios puertos paralelos y el sistema operativo gestiona las interfaces de puerto paralelo con los nombres LPT1, LPT2 y así sucesivamente, las direcciones base de los dos primeros puertos es: - LPT1 = 0x378. - LPT2 = 0x278.

El puerto paralelo está formado por 17 líneas de señales y 8 líneas de tierra. Las líneas de señales están formadas por tres grupos:

- 4 Líneas de control
- 5 Líneas de estado
- 8 Líneas de datos



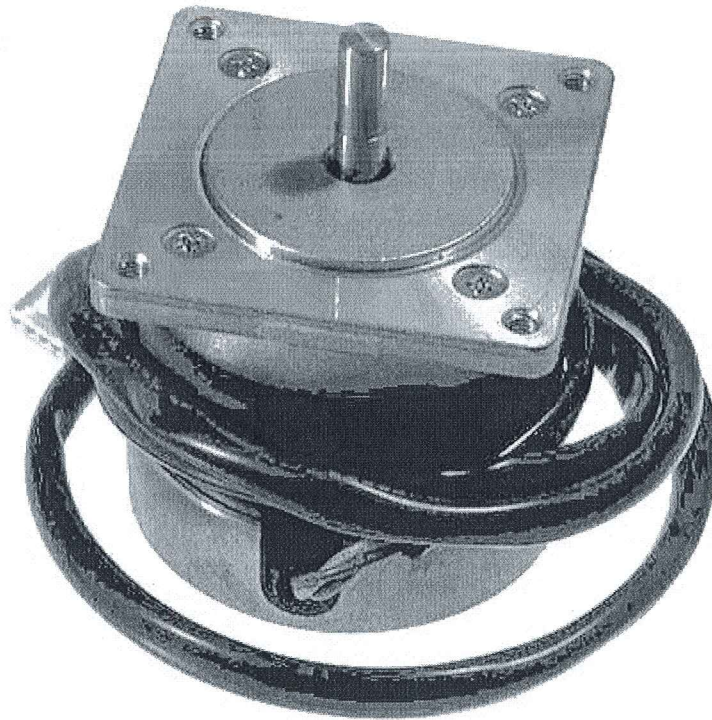


Figura 23: Motor paso a paso

En el diseño original las **líneas de control** son usadas para la interfaz control e intercambio de mensajes desde el PC a la impresora.

Las **líneas de estado** son usadas para intercambio de mensajes, indicadores de estado desde la impresora al PC (falta papel, impresora ocupada, error en la impresora).

Las **líneas de datos** suministran los datos de impresión del PC hacia la impresora y solamente en esa dirección. Las nuevas implementaciones del puerto permiten una comunicación bidireccional mediante estas líneas.

Cada una de estas líneas (control, estado, datos) puede ser referenciada de modo independiente mediante un registro [13].

Cada registro del puerto paralelo es accedido mediante una dirección. El puerto paralelo tiene tres registros:

1. **Registro de datos** encargado mediante el hardware y software necesario del

control del motor paso a paso

2. **Registro de estado** destinado a la lectura del estado de las líneas de interfaz con el sensor
3. **Registro de control** destinado a activar el sensor de posición *home* del motor

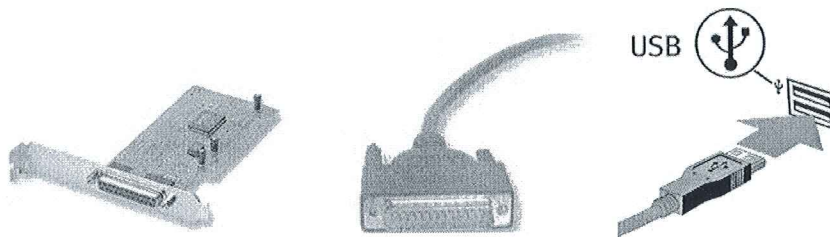


Figura 24: Tarjeta controladora, conector de puerto paralelo y conector USB

### 3.3. La cámara digital

Para este proyecto se ha optado por la utilización de una cámara digital tipo webcam de la marca Logitech, véase fig. 25. Esta cámara genera tanto imágenes estáticas en formato jpeg como secuencias de imágenes en formato AVI. Dispone de un conector USB y una resolución aceptable para los propósitos de este proyecto.

La configuración de la cámara y adquisición de imágenes mediante ésta, se realiza a través del programa desarrollada. Éste utiliza un control ActiveX denominado **VideoOCX** : permite que los programadores integren fácilmente las capacidades de proceso vídeo de la captura y de imagen en usos del software. Además es compatible con la mayoría de los dispositivos de Vídeo-para-Windows (VFW), tales como cámaras fotográficas USB (webcams) y framegrabbers conjuntamente con una cámara fotográfica o el camcorder del CCD.

### 3.4 Almacenamiento de la información fotométrica

Máxima resolución	640 x 480 píxeles
Formato de vídeo digital	AVI
Formato de imagen	Bmp, Jpeg
Velocidad de captura de vídeo	25 frames por segundo (FPS)
Conexión	Puerto USB



Figura 25: Logitech QuickCam Pro3000

### 3.4. Almacenamiento de la información fotométrica

Una vez adquiridas las imágenes es necesario almacenarlas, bien en el equipo informático o en un soporte externo (disco duro extraíble, DVD etc.). El volumen de datos electrónicos que se necesita para almacenar las imágenes puede llegar a ser muy elevado en secuencias relativamente largas, por ejemplo, una imagen en color con una resolución de 640 x 480 píxeles (la ofrecida por la cámara utilizada) ocupa unos 900 kBytes de espacio en disco. Si se desea almacenar las imágenes adquiridas de los 1200 pasos de la plataforma giratoria del robot, el archivo final ocuparía aproximadamente 1054 MBytes. Para reducir el volumen de datos será necesario convertir la secuencia a formato **AVI**. El controlador ActiveX de la aplicación permite utilizar codecs instalados en el sistema para convertir las secuencias de imágenes a formato AVI comprimido o sin comprimir. Con esto se consigue reducir el espacio ocupado hasta en un 98 %, por ejemplo una secuencias de 1200 imágenes puede llegar a ocupar solamente 22 MBytes después de



### **3.4 Almacenamiento de la información fotométrica**

---

haberla comprimido, frente a los 1054 MBytes originales.

### 4. Requerimientos del software

Una vez explicadas las bases teóricas del proyecto se va a analizar los requerimientos de la aplicación desarrollada.

#### 4.1. Análisis de requerimientos

Se analiza cuales son las necesidades reales que debe cubrir la aplicación, para ello es necesario definir los requerimientos de esta.

Se desarrollará una aplicación *escritorio* para Windows que realice los siguiente procesos:

- Adquisición y almacenamiento de la información fotométrica mediante un robot posicionador
- Segmentación de imágenes
- Un método de "compresión" de datos
- Reconstrucción de la información fotométrica

Para ello, la aplicación se debe comunicar con un robot posicionador por el puerto paralelo y una cámara digital por el puerto USB. Además debe ser capaz tanto de procesar como de generar una secuencia de imágenes en formato AVI.

A continuación se explica cada requerimiento.

##### 4.1.1. Adquisición y almacenamiento de la información fotométrica

Para poder realizar la adquisición de la información fotométrica del objeto, la aplicación deberá permitir la captura de imágenes tanto mediante el robot posicionador como sin él.

**Adquisición con robot:** el usuario deberá poder controlar la inclinación del brazo móvil así como mover la plataforma giratoria para realizar los ajustes antes de la propia adquisición, para la cual podrá configurar la cantidad de capturas

por vuelta y el tiempo de espera entre los avances de la plataforma. La aplicación se encargará de realizar la secuencia de captura de información de la vuelta completa, es decir, hacer girar la plataforma del robot, capturar la información fotométrica y almacenarla en una base de datos con los parámetros establecidos.

**Adquisición sin robot** la adquisición sin robot se podrá realizar de dos maneras: directa (busrate) o forma periódica.

1. de manera directa: se capturará las imágenes a la velocidad máxima que permita la cámara y el equipo informático.
2. de forma periódica: las imágenes se capturarán cada cierto intervalo de tiempo

En todos los casos, el usuario podrá seleccionar el tipo de códec, resolución y la ruta y nombre del fichero AVI de salida.

### 4.1.2. Segmentación de imágenes

El programa dispondrá de una aplicación para segmentar una secuencia de imágenes, es decir, que aísle un objeto del fondo que lo rodea.

La aplicación cargará la secuencia de imágenes en formato AVI y generará otro fichero en el mismo formato que contendrá únicamente la información fotométrica del objeto.

El usuario deberá disponer de la información necesaria para establecer los parámetros y configurar los métodos que intervienen en el proceso, éstos son:

- color de fondo que se utilizará como referencia
- el valor del umbral
- las etapas de erosionado y dilatado
- fichero de entrada y de salida
- el códec utilizado para la compresión



- resolución de la imagen del fichero generado

Como parámetros opcionales el usuario podrá establecer:

- el área o zona de la imagen que se procesará
- el modelo de color que se utilizará: RGB o HLS
- si se debe aplicar un proceso de etiquetado o *labeling*
- la cantidad de fotogramas a procesar
- el color final de la zona eliminado, por defecto será negro. Esto facilitará un posible posterior procesado como puede ser la introducción de un fondo artificial

La aplicación debe proporcionar al usuario la información necesaria para poder escoger correctamente éstos parámetros.

### 4.1.3. Método de "compresión" de datos

Se implementará un método para reducir considerablemente el volumen de la información fotométrica del objeto que permita la reconstrucción posterior de esta información.

En concreto, a partir de una secuencia de imágenes segmentadas de un objeto, se debe generar un fichero en formato AVI que contenga una determinada cantidad de vistas reales del objeto y sus correspondientes *mapas de disparidad*.

Para el cálculo de los mapas de disparidad, el usuario configurará los parámetros de las diferentes etapas de procesado como son:

- el tipo de correlación (SSD,SSA o XCA)
- el componente para calcular la correlación (H,L o S)
- el tamaño de la ventana o *window*
- el sentido de giro del objeto

- el color del fondo que se ignorará sí lo hay (para tratamientos posteriores)
- cantidad de vistas reales que se debe procesar

Igual que en los puntos anteriores, la aplicación deberá permitir elegir los parámetros del fichero AVI de salida (resolución y códec).

### 4.1.4. Reconstrucción de la información fotométrica

La aplicación deberá generar las vistas virtuales a partir de los datos generados en el punto anterior.

Dado que este proceso se realizaría en un punto remoto y probablemente mediante un *software* enfocado a Internet (como podría ser un *plug-in* para el navegador, este proceso servirá como método de control de la información obtenida.

### 5. Aplicación desarrollada

#### 5.1. Diseño

A continuación se define la forma de implementación de las funcionalidades de las aplicaciones que se han definido durante la anterior etapa y constituyen los objetivos del proyecto.

##### 5.1.1. Definición de la arquitectura

Dadas las diferentes funcionalidades del programa, se ha dividido el programa en tres capas claramente diferenciadas intentando mantener cierto grado de modularidad. Con esto se consigue que las diferentes partes del programa podrán ser implementadas, utilizadas y mantenidas de manera independiente.

En concreto, se ha dividido el programa en estas tres capas o *módulos* (fig. 26):

- Capa de presentación
- Capa de dominio
- Capa de datos o de entrada/salida y procesamiento

**Capa de presentación** Contiene la interfaz y es por tanto la capa que interacción con el usuario. Deberá incluir los elementos necesarios para recibir las peticiones del usuario, ordenar la ejecución de acciones, mostrar al usuario el resultado de estas, controlar las ventanas, menús, botones etc, es decir, gestionará la coordinación de las peticiones del usuario con la capa de dominio.

**Capa de dominio** Es la responsable de la implementación de las funcionalidades del sistema. Por tanto, ejecutará las diferentes funciones del sistema, realizará las comprobaciones necesarias, conocerá los métodos a realizar, pedirá datos y confirmaciones a la capa de gestión de datos y comunicará los resultados a la capa de presentación.



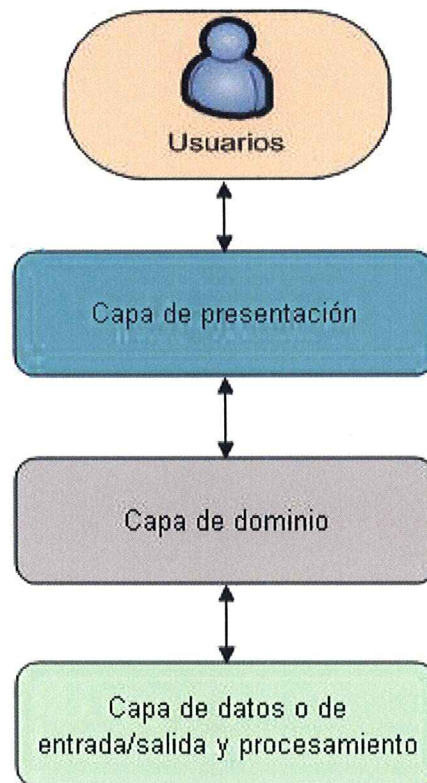


Figura 26: Modelo de capas

**Capa de datos o de entrada/salida y procesamiento** Esta capa se divide en tres partes o *módulos*:

- el controlador de la cámara y archivos AVI: será el responsable de la configuración de la cámara digital y de la adquisición de imágenes tanto a través de ésta como a partir de un fichero AVI existente. Además se encargará de almacenar estas imágenes en un fichero AVI de salida.
- el controlador del Robot: será el responsable de realizar toda la comunicación con el robot posicionador a través del puerto paralelo.
- una clase para el procesamiento de imágenes: esta clase contendrá todas las funciones y métodos necesarios para realizar las funcionalidades de tratamiento de imágenes que se han explicado anteriormente (*segmentación, mapas de disparidad, etc.*)

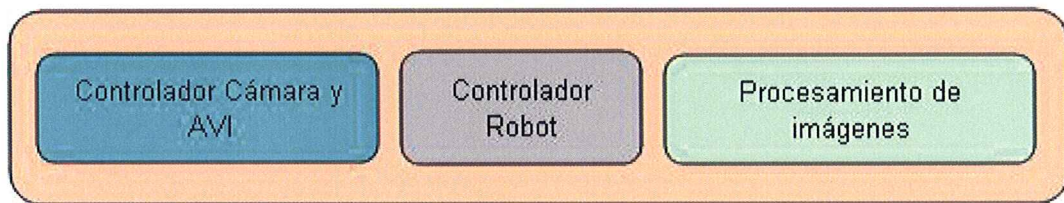


Figura 27: Componentes de la capa de Datos

### 5.1.2. Capa de presentación

La capa de presentación se divide en dos partes:

- El diseño externo o interfaz
- El diseño interno o comunicación entre diseño externo y capa de dominio.

El diseño externo o interfaz es la parte con la cual interactúa el usuario, por lo tanto debe ser amigable e intuitiva.

La capa de interfaz debe incluir los elementos necesarios para recibir las peticiones del usuario, mostrarle el resultado de dichas peticiones y coordinar la resolución de éstas junto con la capa de dominio. El conjunto de estas herramientas constituye el diseño interno.

El diseño interno define la interacción entre el diseño externo y la capa de dominio. Este diseño engloba los mecanismos que recogen, procesan y responden a las peticiones del usuario.

A nivel externo se ha dividido el programa en cinco partes que a nivel de implementación se representan con cinco *cuadros de diálogo* o *formularios*:

- Menú principal
- Un formulario para la adquisición de imágenes
- Un formulario para la segmentación
- Un formulario para la compresión de datos
- Un formulario para la interpolación de vistas

Los elementos que intervienen en el nivel externo son básicamente dos: los *formularios* o *cuadros de diálogo* y los *controles*.

**Formulario** Es la interfaz entre el usuario y la aplicación, también se le denomina *cuadro de diálogo*. Existen dos tipos: *modal* y *no modal*. Un cuadro de diálogo *modal* es un diálogo que bloquea el resto de la aplicación hasta que el usuario responda a lo que se le pide en él. Estos diálogos suelen contener uno o varios *controles* que permiten interactuar con la aplicación.

**Controles** Son elementos que facilita la comunicación entre el usuario y la aplicación normalmente de una forma intuitiva y visual.

**Menú Principal** El menú principal (fig. 52) es el punto de entrada al programa, desde aquí se accederá a las diferentes aplicaciones.

**Diseño externo** En la figura 29 se puede observar la estructura del programa. El menú principal contiene cuatro *botones*, uno para cada aplicación del programa. También se puede acceder a las diferentes aplicaciones a través del menú superior.



Figura 28: Menú Principal

**Diseño interno** El menú principal será un formulario con dos tipos de controles: varios botones y un menú superior. El usuario podrá ejecutar las aplicaciones bien haciendo clic en los botones o bien eligiéndola a través del menú.



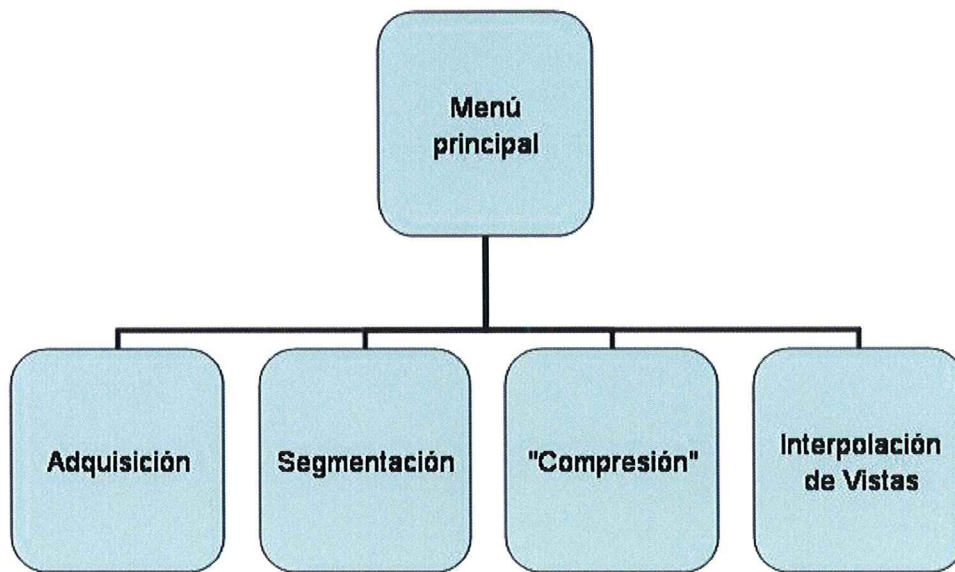


Figura 29: Organigrama del menú principal

En el momento de realizar cualquiera de estas acciones, el programa ejecuta el siguiente procedimiento:

- se ejecuta la función que captura el evento
- ésta crea un objeto del tipo de *formulario* que corresponde a la aplicación que el usuario quiera ejecutar
- se muestra este formulario de forma *modal*
- una vez concluida la ejecución de la aplicación iniciado, se destruye el formulario o *cuadro de diálogo*

**Formulario para la adquisición de imágenes** La aplicación permitirá realizar la adquisición de imágenes tanto con el robot posicionador como sin él. Este cuadro de diálogo permite seleccionar la opción deseada y establecer los parámetros necesarios para cada caso.

**Diseño externo** El formulario dispondrá de:



- un grupo de controles para el tipo de *Adquisición*: dos *radio-button* para seleccionar la grabación **con** el robot posicionador o **sin** él
- un grupo de controles para la *Configuración del Robot*: dos cajas de texto para introducir el número de pasos de cada avance del plato y el tiempo de espera en *ms* entre cada uno
- un grupo de controles para la *Configuración de la Adquisición sin Robot*: dos *radio-button* para seleccionar entre la adquisición a velocidad máxima o *Busrate* y la adquisición periódica. Para este último caso existirá una caja de texto para introducir la frecuencia de captura en *milisegundos*

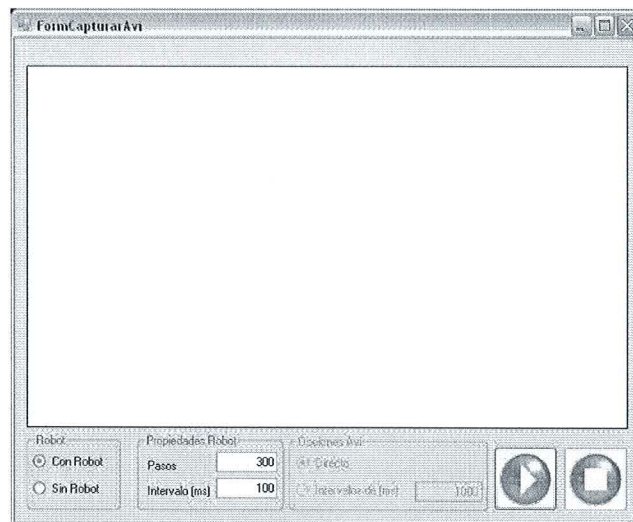


Figura 30: Adquisición con robot

- un control ActiveX que muestra la imagen capturada en tiempo real
- un botón para comenzar la adquisición
- un botón para parar la adquisición

**Diseño interno** Los controles del formulario facilitarán al usuario la configuración del proceso, por lo tanto, debe tener un comportamiento lógico en función de las opciones seleccionadas. Esto se consigue habilitando los controles que in-

tervienen en el proceso y deshabilitando los que no influyen. Los controles deshabilitados no permiten ser seleccionados o que el usuario introduzca datos.

La figura 31 muestra las opciones del proceso (azul) y sus parámetros (en verde). El comportamiento de la aplicación será el siguiente:

- al principio todos los controles menos el grupo para el tipo de *Adquisición* estarán deshabilitados.
- si el usuario selecciona la adquisición con robot se habilita el grupo de controles para la *Configuración del Robot*
  - las dos cajas de texto controlarán que los datos introducidos sean válidos, es decir, que sean valores numéricos. Si se cumple esta condición se habilita el botón para iniciar la adquisición
- si el usuario selecciona la adquisición sin robot se habilita el grupo de controles para la *Configuración de la Adquisición sin Robot*, con lo cual, el usuario podrá elegir entre la adquisición a velocidad de *Busrate* y la periódica:
  - en el primer caso se habilita directamente el botón para iniciar la adquisición
  - en el segundo caso se habilita la caja de texto para la frecuencia de captura. Una vez introducida esta frecuencia la aplicación comprueba que tenga el formato correcto y, en caso afirmativo, habilita el botón para iniciar la adquisición
- una vez iniciada la adquisición, se deshabilita todos los controles menos el botón para interrumpir la adquisición
- al finalizar la adquisición se vuelve al estado inicial

### Formulario para la segmentación

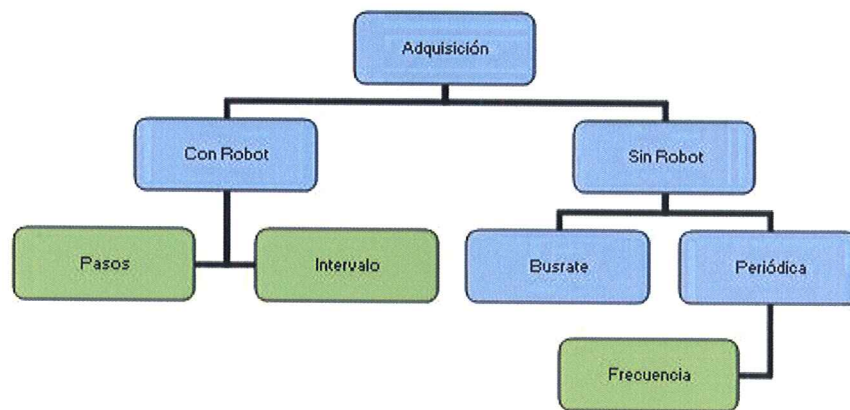


Figura 31: Opciones y parámetros de la adquisición

**Diseño externo** Como se puede observar en el siguiente listado, en el proceso de segmentación interviene gran cantidad de parámetros necesarios para el procesado. Es obvio que la elección de estos parámetros incide directamente en la calidad del resultado final, con lo que se ha añadido una serie de herramientas y funcionalidades que proporcionarán al usuario la información necesaria para establecer estos parámetros correctamente.

**Selección del color de fondo** Dado que la técnica del *Croma Key* admite diferentes colores de fondo, será necesario que el usuario indique el color del fondo de la grabación. El usuario podrá seleccionar este color indicándolo directamente en la primera imagen de la secuencia.

**Umbral o *Threshold*** El valor del umbral es el que indica si un píxel pertenece al fondo o bien al objeto, por lo tanto, es muy importante elegir un valor lo suficientemente alto como para descartar los píxeles cuyo color sea cercano al color del fondo sin descartar los píxeles que realmente pertenecen al objeto. La aplicación permite introducir este valor bien a través de un control de tipo *Numeric Up Down* o bien mediante una barra deslizadora (fig. 33).

Para facilitar esta labor, la aplicación contará con un módulo (fig. 34) que muestra información acerca del color de fondo elegido (valores *RGB* y *HLS*). Este módulo contará además con la posibilidad de obtener la diferencia entre



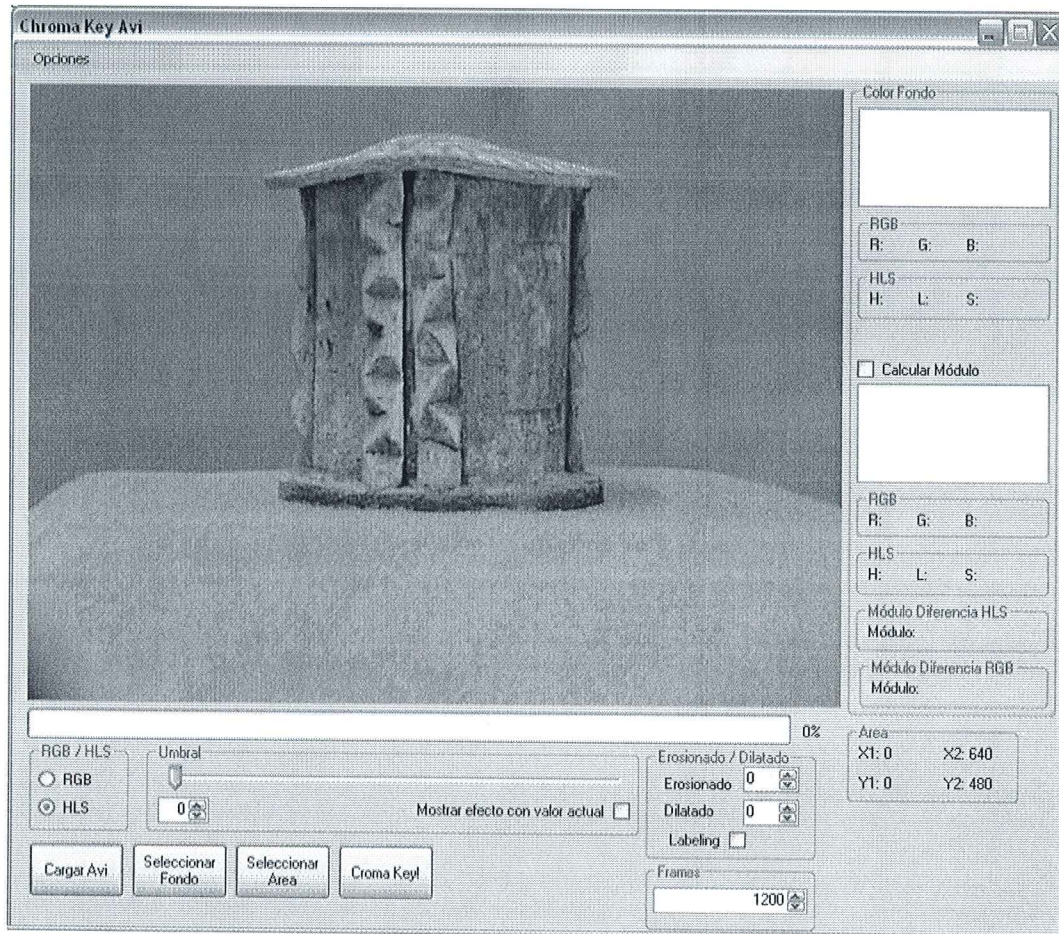


Figura 32: Cuadro de diálogo para la segmentación de imágenes

el color de fondo elegido y el color de los píxeles situando el ratón encima de la imagen.

**Erosionado, Dilatado y Etiquetado** Dependiendo de las características de la imagen es necesario aplicar una o varias veces un erosionado y/o dilatado. El usuario podrá seleccionar la cantidad de veces que quiera aplicar estos métodos así como si se debe aplicar un etiquetado de la imagen o no (fig. 35).

**Cantidad de fotogramas a procesar** El usuario tendrá la oportunidad de indicar la cantidad de imágenes del archivo AVI que se debe procesar, en caso de querer tratar solamente una parte de la secuencia de imágenes (fig. 36).

**Modelo de color** La aplicación permite trabajar con los dos modelos de color *RGB*





Figura 33: Selección del umbral

y *HLS*. La aplicación contará con un control para que el usuario pueda indicar cual prefiere utilizar (fig. 37).

**Nuevo color de fondo** Los píxeles identificados como parte del fondo por defecto se cambiarán a color negro. Si el usuario desea utilizar otro color podrá seleccionar mediante un control de selección de color (fig. 38).

**Selección del área del objeto** La aplicación permite limitar el proceso de segmentación a un determinado área disminuyendo así la cantidad de información a procesar y el coste computacional. Esto provoca que la zona excluida se considere directamente parte del fondo (fig. 39).

**Presentación de resultados** El formulario tendrá un control ActiveX que muestra en cada momento la imagen procesada.

**Carga del fichero AVI e inicio del proceso** Se ha agregado un *botón* para iniciar la carga del fichero AVI y un botón para iniciar el proceso de segmentación

Dado que el proceso de segmentación puede ser un proceso más o menos largo, el usuario podrá ver el resultado de la segmentación para el primer fotograma antes de procesar la secuencia entera. Esto le facilitará la elección de los parámetros. Además se ha incorporado una barra de progreso para que el usuario disponga de la información del tiempo que resta para finalizar la aplicación.

**Diseño interno** Al inicio de la aplicación, puesto que no se ha cargado la secuencia de imágenes sobre la que se va a trabajar, todos los controles estarán deshabilitados.

Para cargar el archivo AVI, el usuario hará clic en el botón correspondiente con lo que se abrirá un *cuadro de diálogo* para indicar la ruta y el nombre del fichero.

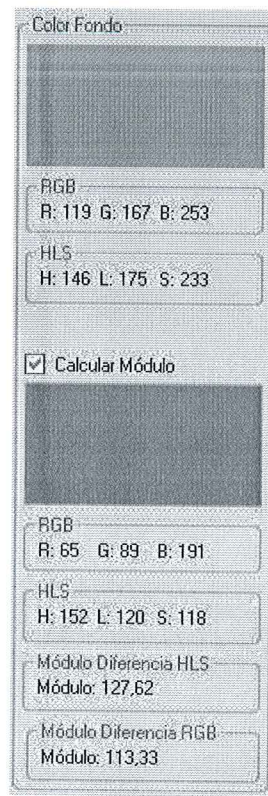


Figura 34: Módulo de información acerca del color de fondo elegido

Una vez cargado el fichero, se habilita el resto de controles.

A continuación se explica las acciones más complejas que puede realizar el usuario:

**Seleccionar el color de fondo** Para seleccionar el color de fondo, el usuario acciona el botón correspondiente y se deshabilita todos los controles del formulario. La selección del color se realiza situando el puntero del ratón encima del control ActiveX y la aplicación se encarga de obtener el color del píxel que está debajo del puntero. Este color así como sus valores RGB y HLS se mostrarán en una zona dedicada a ello. Asimismo, habrá una zona que puede ser habilitada por el usuario y que mostrará el color actual debajo del puntero y la diferencia entre este color y el color de fondo que se ha elegido.

**Seleccionar área a procesar** Si el usuario desea procesar solamente una determinada zona de la imagen, puede indicarla dibujando un rectángulo directa-



Figura 35: Controles para establecer el erosionado, dilatado y etiquetado de la imagen



Figura 36: Control para seleccionar la cantidad de fotogramas a procesar

mente sobre el control ActiveX. Para ello debe hacer clic en el botón correspondiente, lo que provoca que se desactive el resto de controles. A continuación selecciona el punto de una esquina del rectángulo y después la esquina opuesta. El área contenido en este rectángulo será el que se procesará durante la segmentación.

**Presentación del resultado antes del proceso de segmentación** El usuario podrá habilitar la presentación del resultado de la segmentación antes de comenzar el procesado de la secuencia completa. Esto le permite ver el resultado del proceso mientras ajusta los parámetros. La aplicación deberá procesar y mostrar la primera imagen de la secuencia con los parámetros que se han seleccionado en ese momento y refrescarla en el momento de cambiar algún parámetro.

El resto de parámetros se controla a través de controles de tipo *radio-button*, *checkbox* etc.

**Formulario para la compresión de datos** Esta aplicación realiza la "compresión" de la información fotométrica del objeto calculando el mapa de disparidad.





Figura 37: Controles para seleccionar el modelo de color

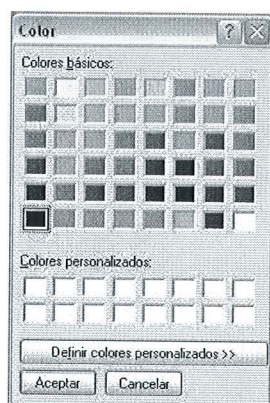


Figura 38: Cuadro de diálogo para seleccionar el nuevo color de fondo

El usuario podrá establecer todos los parámetros que intervienen en los procesos necesarios para generar el mapa de disparidad. Estos son el tipo de correlación, el componente HLS para realizar la correlación, el área de búsqueda y el tamaño de la ventana, además podrá indicar qué fotogramas de la secuencia se toman para el cálculo del mapa.

**Diseño externo** El formulario contendrá los siguientes controles:

- un botón para cargar el fichero AVI
- un botón para seleccionar el color de fondo
- un botón para iniciar el proceso
- un grupo de *radio-buttons* para seleccionar el tipo de correlación



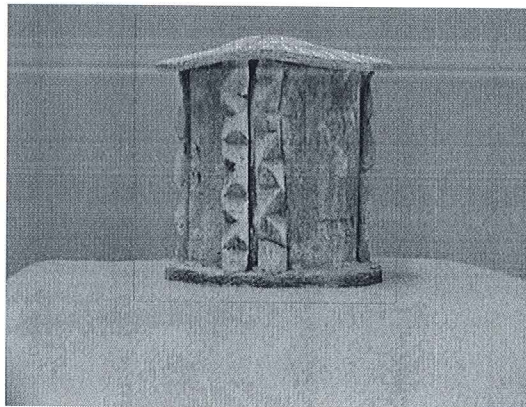


Figura 39: Seleccionando el área a procesar

- un grupo de *radio-buttons* para seleccionar el componente del color para calcular la correlación
- varios controles de tipo *Numeric Up Down* para seleccionar el tamaño de la ventana o *Window*, el área de búsqueda y la frecuencia de vistas reales
- un control de tipo *Drop Down Box* para indicar el sentido de giro del objeto
- un control ActiveX para mostrar la secuencia de imágenes que se va a procesar

**Diseño interno** Como en los anteriores formularios, al inicio todos los controles menos el botón para cargar el fichero AVI estarán deshabilitados y el usuario debe indicar el nombre y la ruta del fichero AVI. Una vez cargado este fichero, se muestra la primera imagen de la secuencia en el control ActiveX y se habilita el resto de controles.

A continuación el usuario debe indicar el color que no se tendrá en cuenta a la hora de procesar la imagen, normalmente el color de fondo, seleccionándolo directamente en la imagen con el puntero.

El resto de parámetros se controla mediante el resto de controles simples.

### Formulario para la interpolación de vistas

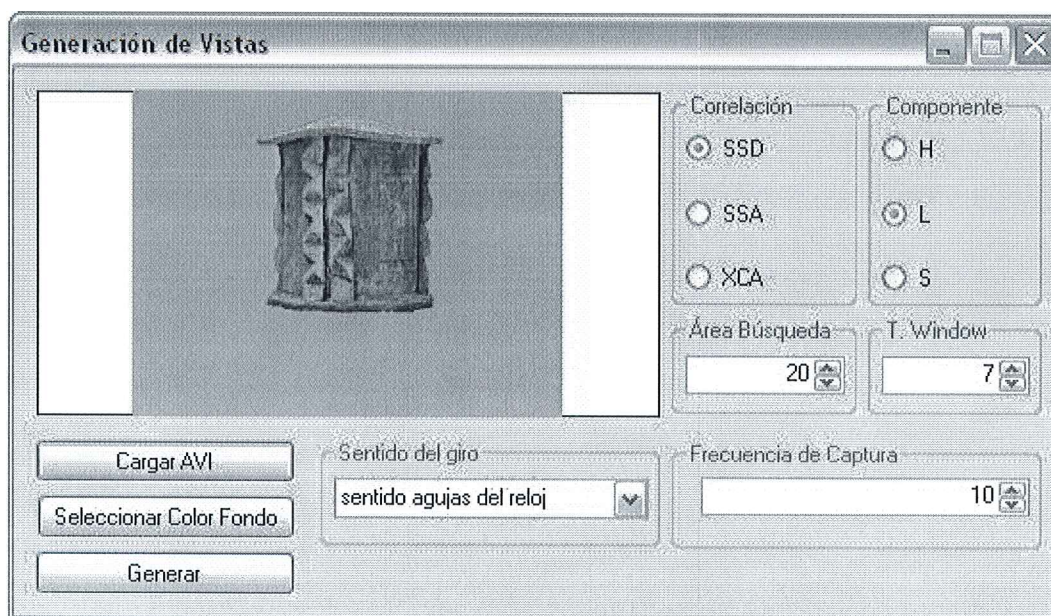


Figura 40: Formulario para el proceso de compresión

**Diseño externo** El formulario para la interpolación de vistas dispondrá de los siguientes controles:

- botones para la carga del fichero, para la selección del color de fondo y para iniciar el proceso de interpolación
- un control de tipo *Numeric Up Down* para indicar la cantidad de vistas *virtuales* que se debe generar
- un control de tipo *Drop Down Box* para indicar el sentido de giro del objeto

**Diseño interno** El diseño interno de la capa de presentación de esta aplicación es muy parecido al de las aplicaciones anteriores: al inicio todos los controles menos el botón para cargar el fichero AVI estarán deshabilitados y el usuario debe indicar el nombre y la ruta del fichero AVI.

Una vez cargado este fichero, se muestra la primera imagen de la secuencia en el control ActiveX y se habilita el resto de controles.

El usuario deberá indicar el color que no se debe procesar, sí lo hay, la cantidad de vistas virtuales que se generarán y el sentido de giro del objeto.