



UNIVERSIDAD AUTONOMA DE SAN LUIS POTOSI
FACULTAD DE CIENCIAS



Digitalizador de Objetos en 3D

TESIS PROFESIONAL
para obtener el título de

Ingeniero Electrónico

PRESENTA:

Edgar Daniel Sauri de la Garza

SAN LUIS POTOSI, S. L. P. Marzo del 2004



UNIVERSIDAD AUTONOMA DE SAN LUIS POTOSÍ
FACULTAD DE CIENCIAS



Digitalizador de Objetos en 3D

TESIS PROFESIONAL
para obtener el título de

Ingeniero Electrónico

PRESENTA:

Edgar Daniel Sauri de la Garza

ASESORES DE TESIS:

M.C. Gilberto Marrufo Quirino
Ing. Roberto Bartali Marchetti
Fís. Héctor Eduardo Medellín Anaya

SAN LUIS POTOSÍ, S. L. P. Marzo del 2004

Índice General

<i>Tema</i>	<i>Página</i>
<i>Introducción</i>	5
<i>Capítulo 1: Esquema del sistema</i>	6
<i>1.1 Desarrollo del sistema</i>	8
<i>Capítulo 2: Módulo UBICACIÓN</i>	9
<i>2.1 Desarrollo del módulo</i>	10
<i>2.2 Pseudocódigo del módulo UBICACIÓN</i>	11
<i>Capítulo 3 : Módulo FOTOGRAFIAR</i>	12
<i>3.1 Comunicación del puerto paralelo hacia el hardware</i>	14
<i>3.1.1 El puerto paralelo</i>	14
<i>3.1.2 Bytes de "Data", "Status" y "Control"</i>	16
<i>3.1.3 PortTalk –Manejador para los Dispositivos de I/O en Windows NT/2000/XP</i>	17
<i>3.1.4 Acceso a los Puertos de I/O bajo Windows NT/2000/XP</i>	18
<i>3.1.5 Compatibilidad Usando aplicaciones bajo Windows NT/2000/XP</i>	18
<i>3.2 El paquete Lead Tools</i>	19
<i>3.2.1 Introducción a los Bits por Píxel</i>	20
<i>3.2.2 Lead Image y Lead Twain</i>	21
<i>3.3 El submódulo de Hardware</i>	22
<i>3.3.1 El laser-cámara</i>	23
<i>3.3.2 Selección de la Cámara para el Digitalizador</i>	24
<i>3.3.3 El sistema de motores</i>	26

3.4 Submódulo Software	29
3.4.1 Captura de imagen	29
3.4.2 Submódulo control de motores	30
3.4.3 Submódulo control de motores auxiliares	30
3.5 Proceso de captura de imágenes	30
3.6 Pseudocódigo del módulo FOTOGRAFIAR	32
Capítulo 4 : Módulo Filtro de Gabor	36
4.1 Submódulo Abrir Archivo	37
4.2 Submódulo Rotar la imagen	38
4.3 Submódulo Aplicando Filtro de Gabor	38
4.3.1 La forma de analizar las imágenes	39
4.4 Submódulo Guardar los resultados en archivo	39
4.5 Pseudocódigo del módulo Filtro de gabor	40
Capítulo 5 : Módulo CALIBRAR	42
5.1 Submódulo Hardware	43
5.2 Submódulo Software	45
5.2.1 Submódulo Eje rotación	46
5.2.2 Submódulo Abrir archivo	46
5.2.3 Submódulo Datos objeto	47
5.2.4 Submódulo Ver	48
5.2.5 Submódulo Dibujar	48
5.2.6 Submódulo Borrar	49
5.2.7 Submódulo Ajustar	50
5.2.8 Submódulo Crear Archivo	56
5.3 Pseudocódigo del módulo CALIBRAR	56
Capítulo 6 : Módulo Visualización	59
6.1 Open GL	60
6.2 Submódulo Crear nuevo	61
6.2.1 Crear una sola lista de imágenes	62
6.3 Submódulo Visualizar	63

<i>6.3.1 Transformación de puntos 3D a Píxeles</i>	64
<i>6.3.2 Abrir Archivo</i>	65
<i>6.4 Pseudocódigo del módulo VISUALIZAR</i>	67
<i>Capítulo 7 : Conclusiones y Trabajo a futuro</i>	69
<i>7.1 Conclusiones</i>	69
<i>7.2 Trabajo a futuro</i>	69
<i>Lista de los módulos del sistema</i>	70
<i>Referencias:</i>	71

INTRODUCCIÓN

En el ámbito de los sistemas de adquisición de datos, la visión en las computadoras, denominada visión artificial, se ha convertido en parte vital para entregar información útil. Estos avances han permitido que las computadoras perciban la realidad a través de sensores que funcionan normalmente con principios Ópticos. Estos sistemas denominados comúnmente como escáner pueden ser de 2D (2 dimensiones) o de 3D(3 dimensiones). Los más comunes son los de 2D, que perciben imágenes de objetos planos; Los de 3D, pueden digitalizar objetos en el espacio.

Debido a que estos sistemas no son aún tan comunes en nuestro medio y aún a su alto costo en la industria moderna, el uso de estas tecnologías, se ha convertido en una obligación para ser más competitivos. Es una ventaja que se pueda aumentar la eficiencia en cualquier proceso. En la actualidad, el escáner 3D es una herramienta de gran utilidad en el diseño en ingeniería. Estos equipos permiten obtener objetos virtuales óptimos para realizar todas las simulaciones requeridas en un proceso de diseño. Existen diversas clases de escáner 3D. Muchos de ellos emplean métodos láser, que les permite obtener una gran precisión en los datos adquiridos.

Esta tesis se ha basado en un problema de una empresa del calzado el cual quiere ser más eficiente en su área de producción. Quieren un sistema que sea capaz de escanear una horma de zapato el cual se pueda visualizar en una pantalla para luego trabajar sobre ella agregándole textura, color y otros detalles, que el sistema sea capaz de hacer una relación de medida entre el objeto real y el visualizado en la pantalla para que luego esa información de la imagen se pueda convertir en un molde plano que se pasara a una máquina cortadora de piel, así minimizando la pérdida de material en la realización del calzado.

Una de las posibles soluciones a este problema es realizar un Digitalizador de tipo escáner, proponiendo para ello el tratamiento de líneas generadas por el rayo láser. Se pretende utilizar métodos robustos que detecten las líneas proyectadas sobre el objeto.

El objetivo de este sistema es llegar a la parte de visualización del objeto y para lograrlo se desarrolló lo necesario para obtenerlo. Dado que la documentación total del desarrollo del software realizado es extenso se optó por agregarlo en un disco compacto (CD), adjunto a este trabajo, en este CD se encuentra el programa ejecutable y sus fuentes, así como algunas aplicaciones utilizadas, se agrega también documentación extra la cual sirvieron de referencia para el desarrollo de algunos módulos del sistema y también se agregó información obtenida durante el desarrollo de esta.

Capítulo 1 : Esquema del sistema.

El análisis del sistema es una parte que engloba todos los requerimientos necesarios para el desarrollo de esta, se identificó las necesidades, se realizó el análisis técnico y se asignaron funciones al software. Todo esto siguiendo unos lineamientos ya establecidos.

El sistema tiene que trabajar de la siguiente manera, se quiere que el láser y la cámara se consideren como uno solo, que estén puestos en una base que sea capaz de moverse en un eje, que suba o baje según lo requerido. Una base para colocar el objeto el cual se rota para poder fotografiarlo por completo y estas fotos tomadas serán guardadas en disco para luego de alguna manera analizarlas y con los datos obtenidos visualizar el objeto en pantalla.

El sistema debe de contar con un láser, con una cámara digital, una base móvil, una base giratoria para colocar el objeto, una computadora personal (PC) con sistema operativo Windows 2000 (2K) o Windows Xp ya que este sistema es más comercial actualmente y vienen ya instalada en casi toda PC comercial, por medio de esta PC se controlara el sistema del Digitalizador.

Una vez que se tengan los requerimientos para realizar el sistema se hace un diagrama modular, como en la figura 1.1.

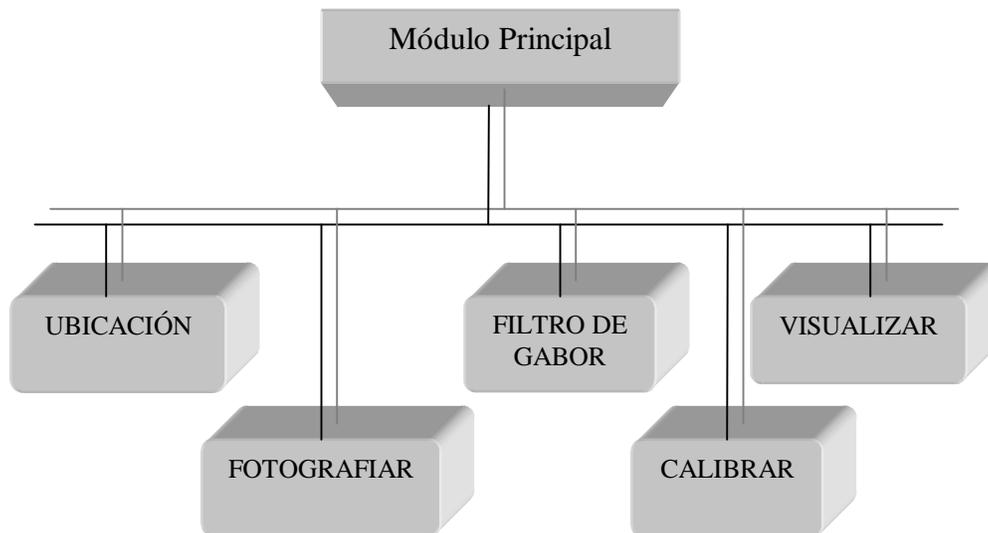


Figura 1.1. Diagrama modular del Digitalizador donde se ven los módulos principales que contiene.

Módulo PRINCIPAL: este módulo esta compuesto de software, de aquí se puede partir a los siguiente módulos o terminar una sesión

Módulo UBICACIÓN: este módulo esta compuesto de software, este es el encargado de dar la ubicación ya sea para salvar o para abrir un archivo o un conjunto de archivos.

Módulo FOTOGRAFIAR: este módulo esta compuesto por hardware y software, ya que este módulo será capaz de fotografiar un objeto y las imágenes guardarlas en disco, llevar el control de los motores para mover la base del objeto y para la base del láser-cámara que servirán para fotografiar el objeto.

Módulo FILTRO DE GABOR: este módulo esta compuesto por software, dado que las imágenes previamente tomadas y guardadas en disco, son imágenes de la línea reflejada sobre el objeto que tiene un grosor, pasa la imagen por el Filtro de Gabor que sirve para filtrar un conjunto de coordenadas de esa línea, quedando como resultado de ese conjunto una sola coordenada que al unirla con otras coordenadas de la misma imagen dan la misma imagen pero ya adelgazada.

Módulo CALIBRAR: este módulo esta compuesto por hardware y software, es el encargado de calibrar el sistema, de crear un archivo donde se guarden los datos necesarios para crear una relación de medida entre objeto real que servirá para desplegar en pantalla, también guardará el dato del punto de rotación del objeto el cual sirve para unir las imágenes tomadas del objeto ya que algunas de ellas se rotan, este módulo solo es necesario activarlo cuando el Digitalizador se prepara por primera vez o se hizo algún cambio en la posición del láser y de la cámara.

Módulo VISUALIZAR: este módulo esta compuesto por software, este utiliza los datos obtenidos por el Filtro de Gabor y también los datos del archivo creado por el módulo CALIBRAR, una vez leídos los datos servirán para desplegar en pantalla el objeto.

Estos módulos a su vez están compuestos por otros sub-módulos que ayudan a realizar el trabajo de los módulos principales.

El lenguaje que se uso para desarrollar el software fue el Builder C++, ya que este tiene librerías más adecuadas para el desarrollo de interfaces amigables y soporta librerías de tipo *Visual Component Libraries* (VCL)[18]. Se aplico Open Gl sobre Builder C++ para desarrollar la parte de la visualización.

1.1 Desarrollo del sistema.

En la figura 1.1, del módulo Principal puede ir al módulo UBICACIÓN y de ahí se puede tomar dos caminos, uno al módulo FOTOGRAFIAR si se va a fotografiar por primera vez un objeto, el segundo es al módulo FILTRO DE GABOR esto porque se puede tener las imágenes ya guardadas en disco de una sesión anterior, de este último podemos pasar al módulo CALIBRAR o pasar desde el inicio a este, cuando las imágenes ya están guardadas en disco y si la calibración del sistema no se ha realizado, de estos últimos módulos puede continuar hacia el módulo VISUALIZAR para desplegar el objeto en la pantalla.

Se requiere que el sistema en su parte del software, contenga ventanas amigables para el usuario, así que el módulo PRINCIPAL es la ventana principal menú del Digitalizador [7], figura 1.2, en esta se encuentran las opciones que puede seleccionar y que son los principales módulos del sistema.

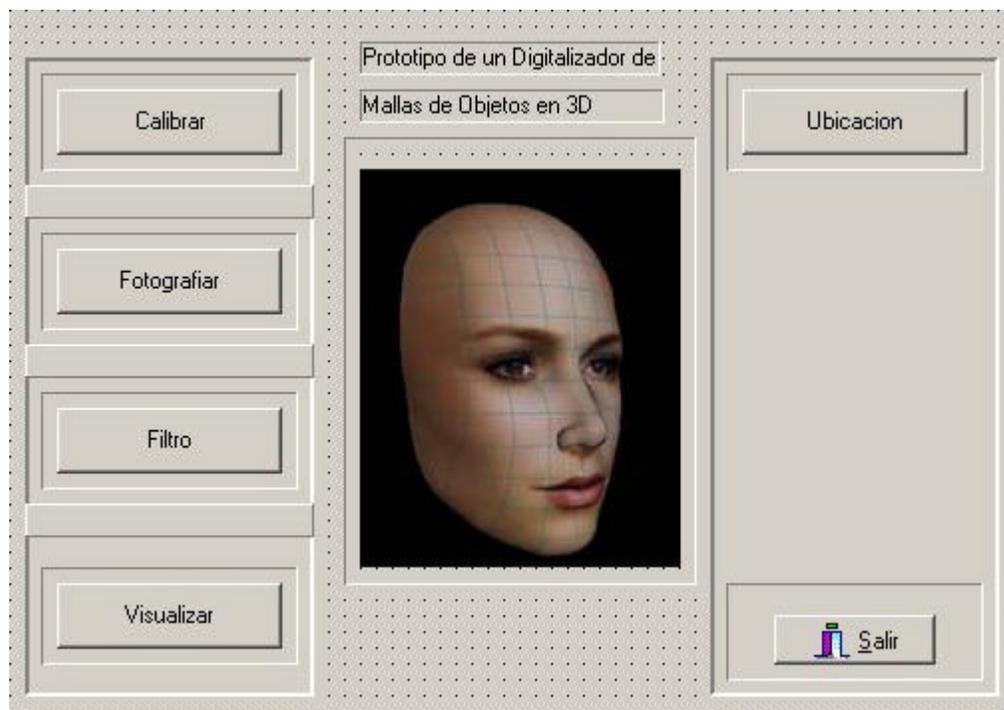


Figura 1.2. Ventana principal del programa del Digitalizador.

Al seleccionar cualquier opción este abrirá otra ventana, transfiriendo el control a los submódulos que posteriormente se denominarán subprogramas, al terminar de ejecutar sus funciones devuelven el control al módulo PRINCIPAL. Como algunos módulos son demasiado complejos se dividieron en módulos más pequeños.

Así que se empieza a desarrollar todos los módulos principales y los submódulos que contienen.

Capítulo 2 : Módulo UBICACIÓN

En la figura 1.2 del capítulo anterior, se mostró la ventana del menú principal y la opción Ubicación que corresponde al módulo UBICACIÓN, su función es tener disponible la ubicación en disco del archivo o conjunto de archivos que serán guardados o abiertos dependiendo del módulo que se seleccione después.

En la figura 2.1 se muestra el módulo UBICACIÓN con sus submódulos y los caminos que puede seguir.

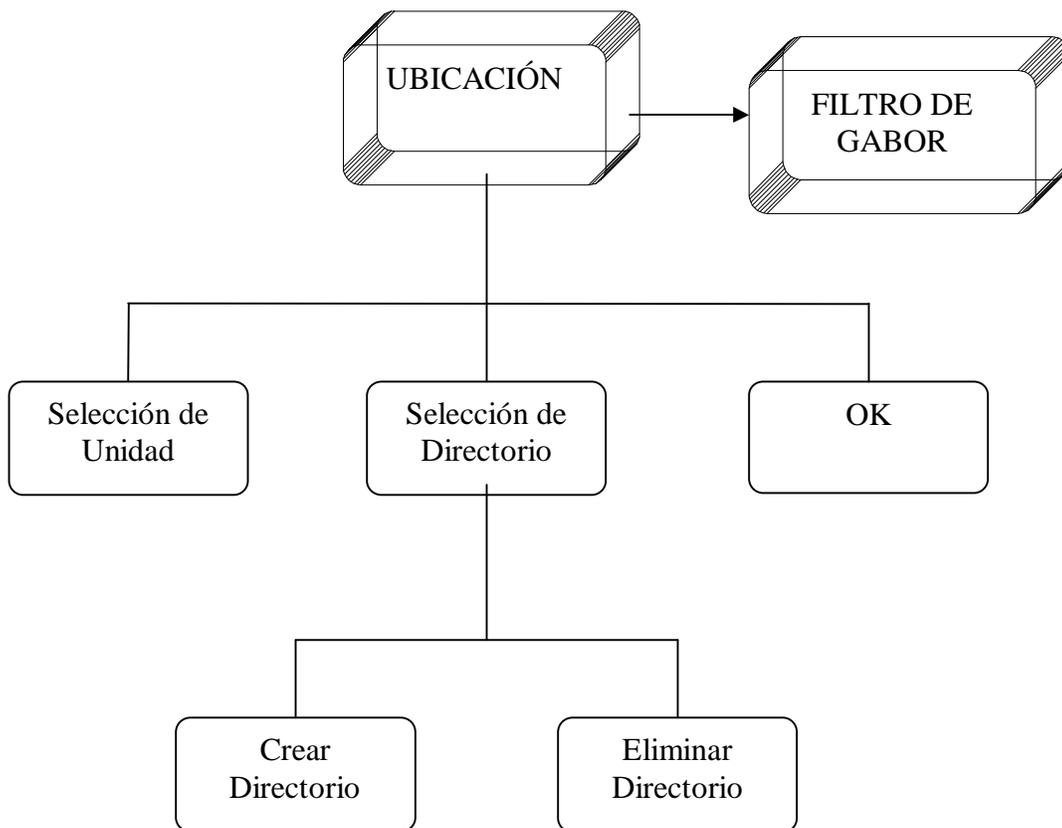


Figura 2.1 submódulos de UBICACIÓN

2.1 Desarrollo del módulo

Una vez activado este módulo se abre una ventana, figura 2.2, en la cual el usuario deberá seleccionar la unidad de disco, esto es porque se pueden tener diferentes unidades en una misma PC, después se puede seleccionar un directorio donde estarán los archivos con los cuales se va a trabajar o donde se desea guardar, o se puede crear un nuevo directorio. También se puede eliminar un directorio que ya no es necesario, en la

figura 2.2 se puede ver una marco donde dice Localización del Proyecto, ahí se visualiza la ruta que tendrá para trabajar, al salir del módulo ya se habrá tomado la ultima ruta, figura 2.3, que será donde se guardará o abrirán los archivos.

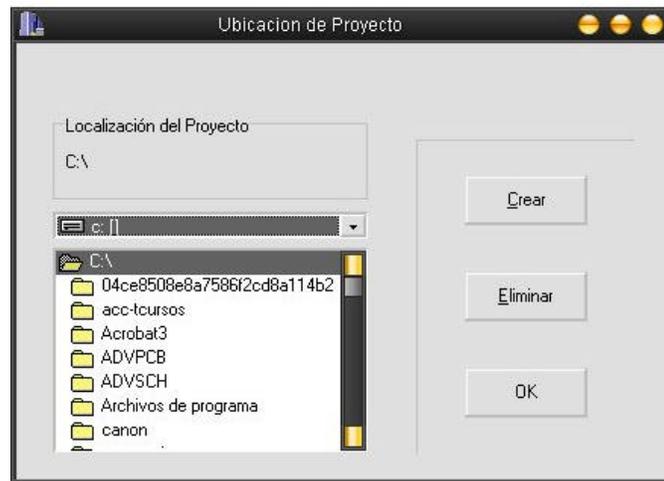


Figura 2.2 Ventana del módulo Ubicación

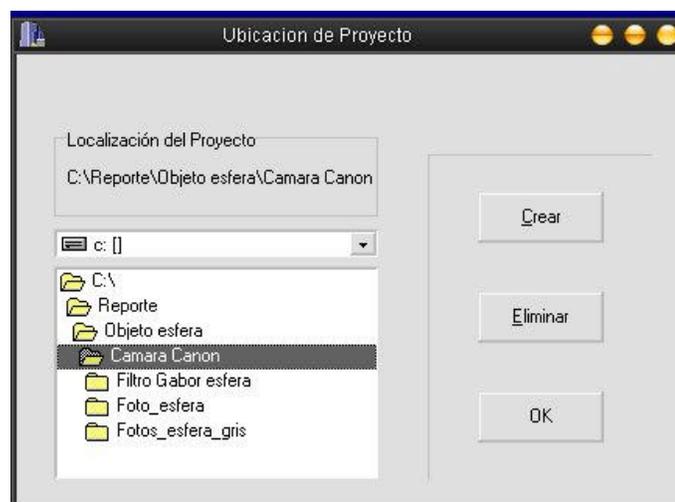


Figura 2.3 Selección de una ruta de trabajo.

Al terminar de ejecutar su tarea, esta ventana se cierra y regresa a la ventana principal donde el usuario selecciona el siguiente módulo a ejecutar.

2.2 Pseudocódigo del módulo UBICACIÓN

En esta tesis se utilizará el pseudocódigo para describir los programas ya que es una herramienta de programación la cual permite describir con palabras los programas. Así que el pseudocódigo del módulo UBICACIÓN [19] es el siguiente:

Módulo Ubicación

Submódulo Unidad

Buscar las unidades disponibles
Tomar una unidad de default
Visualiza la unidad
Puede seleccionar la unidad

Submódulo Directorio

Buscan los directorios disponibles en esa unidad
Visualizan los directorios en una lista
Puede seleccionar el directorio
Si selecciona un directorio y se actualiza la lista
Se despliega el contenido del directorio si la hay

Submódulo Crear

Toma la ruta de la unidad seleccionada
Si hay ruta para crear dentro de otro directorio
Crea un nuevo directorio
Actualiza la lista de directorios

Submódulo Eliminar

Toma la ruta del directorio a eliminar
Confirma que desea borrar ese directorio
Elimina el directorio si se confirma
Actualiza la lista de directorios

Submódulo OK

Regresar una bandera en verdadero al modulo PRINCIPAL
En una variable se guarda la ruta que usara el módulo PRINCIPAL
Cerrar la ventana del módulo Ubicación

Termina el módulo UBICACIÓN
Regresa el control al módulo PRINCIPAL

Capítulo 3 : Módulo FOTOGRAFIAR

Este módulo está compuesto por software y hardware, en la parte de software se utilizaron dos herramientas externas, una de ellas facilita el manejo de las imágenes, la otra facilita la comunicación hacia el hardware.

La parte del hardware son todos los componentes físicos, mecánicos, electrónicos que componen la parte física del Digitalizador.

En la figura 3.1 se muestra el diagrama del módulo FOTOGRAFIAR y sus submódulos estos últimos tienen comunicación entre ellos ya que por medio del software se realizarán los movimientos de los motores y por medio de la cámara digital se capturará la imagen de la luz láser reflejada sobre el objeto, esta captura es manejada por software la cual será guardada en disco para su utilización en el módulo FILTRO DE GABOR.

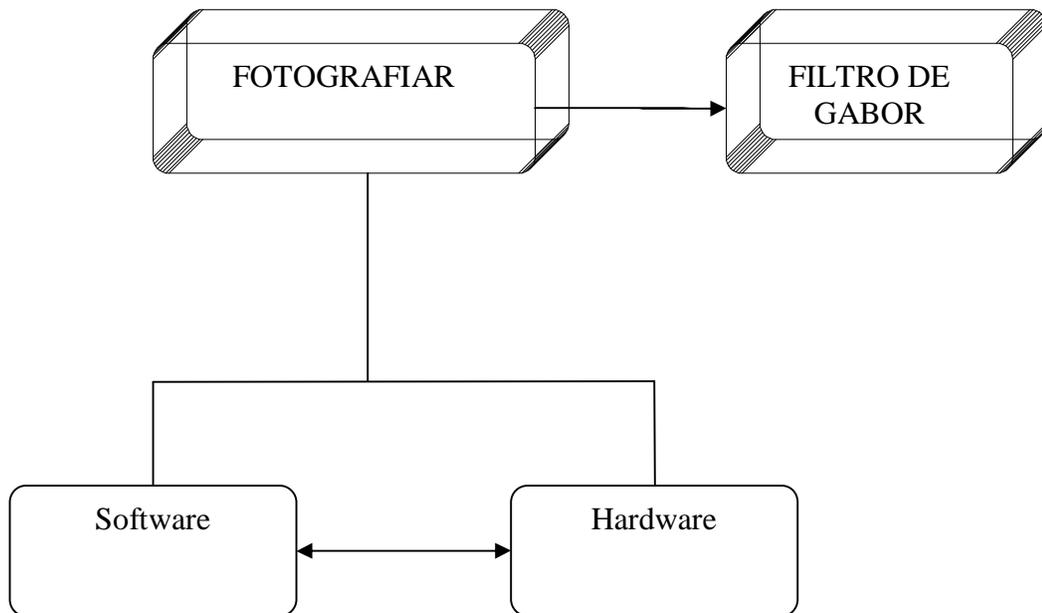


Figura 3.1 Submódulos que componen al módulo FOTOGRAFIAR

En la figura 3.2 se ve el diagrama del módulo de Hardware en la que contiene dos motores a pasos, uno de ellos se utiliza para mover la base donde se encuentran la cámara digital y el láser, el otro motor se utiliza para girar la base donde se monta el objeto.

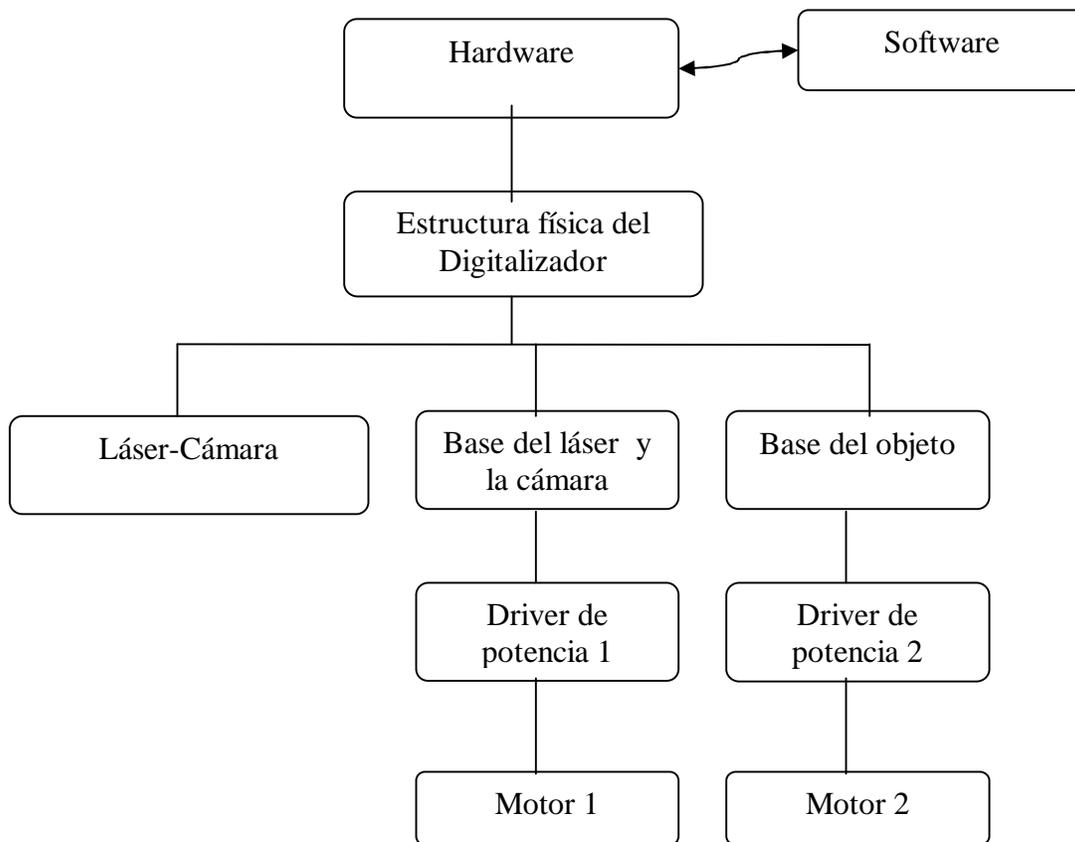


Figura 3.2 Diagrama de módulos del hardware.

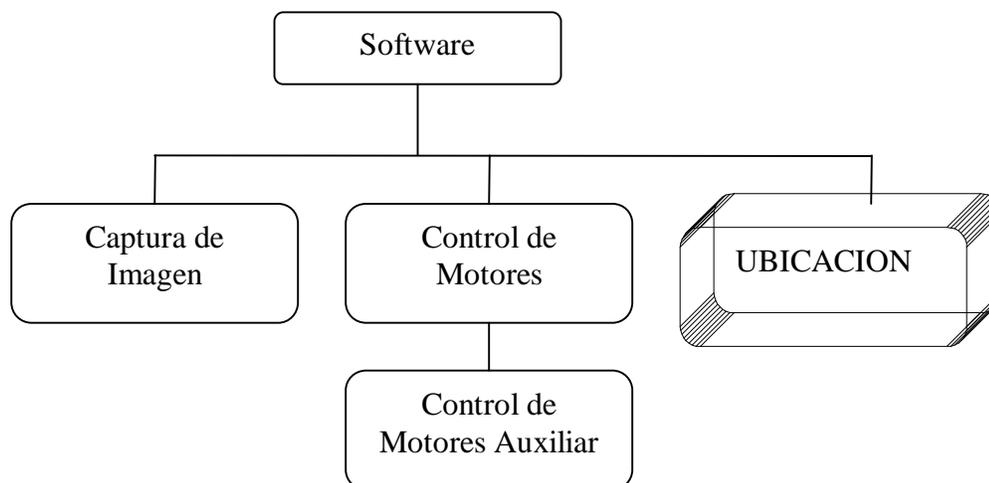


Figura 3.3 Diagrama de módulo del software

En la figura 3.3 se ve el diagrama de submódulos que componen al módulo de Software, este hace un llamado a un módulo PRINCIPAL el cual devuelve una dirección, como se vio para el módulo UBICACIÓN, en esa dirección se guarda la imagen obtenida por la cámara, después de la captura se activa el motor de la base del objeto, luego de completar la toma de fotos seleccionada por el usuario, se activa el motor que mueve la base donde se encuentra el láser-cámara y volviendo hacer el proceso de la toma de fotos, se repite esto dependiendo de los datos que se asignaron.

Para poder desarrollar este módulo se utilizaron dos herramientas extras y externas, una de ellas ayuda a la comunicación del puerto paralelo con el hardware, esto debido que el sistema operativo Windows Xp restringe el uso libre del puerto paralelo, eso debe por seguridad del sistema, la otra ayuda al fácil manejo de las imágenes.

3.1 Comunicación del puerto paralelo hacia el hardware en el sistema operativo Windows 2000 y Windows Xp

Para lograr la comunicación a través del puerto paralelo hacia el hardware del Digitalizador, cuyo objetivo es tener el control de los motores a pasos que se encargan del movimiento del láser-cámara y de la rotación del objeto, que interactúan con el sistema operativo.

En seguida se especifican los archivos necesarios para hacer posible esta comunicación, los cuales son de uso libre y se encuentran disponibles en internet [3]:

- ? Allowio.exe
- ? Porttalk.reg
- ? Porttalk.sys
- ? PortTalk_IOCTL.h
- ? Porttalk.c
- ? Porttalk.rc
- ? Ioexample.c
- ? Pt_ioctl.c
- ? Readme.txt
- ? Porttalk.pdf

3.1.1 El puerto paralelo

Este puerto se comunica con el mundo mediante un conector de 25 terminales disponible en toda computadora personal (PC), el fue diseñado específicamente para la conexión de una impresora [Van Gilluwe, 94]. Este tipo de conector es llamado "DB 25 hembra" (Las siglas "DB" significan "Data Bus").

Los puertos paralelos han permanecido iguales desde las primeras PC. Se han hecho algunas extensiones a su funcionamiento, pero siempre respetando la compatibilidad con los modelos anteriores. El modelo de puerto paralelo que se describe aquí es el modelo estándar, de modo que su funcionamiento es válido para cualquier tipo de PC.

Este modelo es unidireccional (significa que cada terminal es de entrada o salida, pero no ambos) y, puede transmitir 150 Kbytes por segundo; el cual es un estándar a partir de la computadoras y IBM modelo XT. Actualmente mediante rutinas adecuadas de software se puede alcanzar una comunicación de 2 Mbytes por segundo, inclusive se puede obtener la comunicación bidireccional a través de este puerto. Sin embargo, se pretende aquí definir globalmente algo que pueda funcionar por igual en todas las PCs, y no solamente en los modelos más nuevos.

La figura 3.4 muestra el diagrama de un puerto paralelo estándar. El dibujo muestra la salida del puerto tal y como se ve al estar viendo a la computadora desde atrás, y se encuentra en la mayoría de los manuales técnicos. Las señales que salen del puerto, indicadas con una flecha hacia fuera, representan los comandos que se envían a esta: salto de hoja (*auto feed*), inicialización (*initialize*), selección (*select*). Las señales que se reciben de la impresora indicadas con una flecha hacia al puerto son: error (*error*), impresora no seleccionada (*select*), impresora ocupada (*busy*), sin papel (*out of paper*) y reconocimiento (*acknowledgment*). Además de las líneas de control disponemos de ocho terminales para la transferencia de datos enumeradas desde DO 0 hasta DO 7. Las ocho líneas restantes son conexiones de tierra[3].

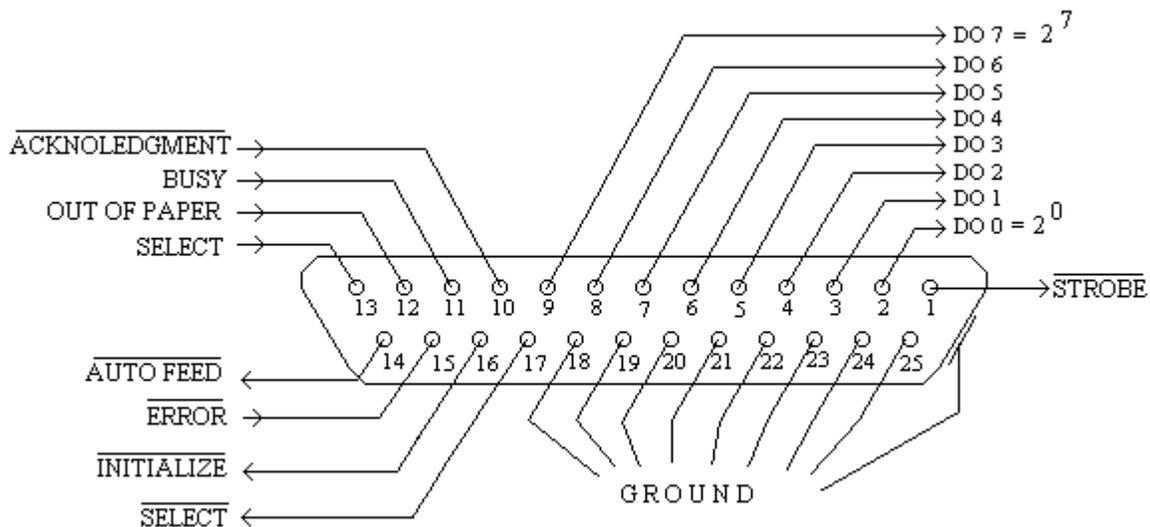


Figura 3.4 Vista del conector de un puerto paralelo

Como se puede observar tenemos un total de doce líneas de salida (bits) y cinco de entrada, lo cual es una desafortunada combinación desde el punto de vista de la comunicación paralela. Podemos mandar perfectamente un byte (8 bits) de datos al exterior, pero tenemos que recibirlo en partes.

3.1.2 Bytes de “Data”, “Status” y “Control”

Debido a la configuración de bits de entrada y salida que tiene el puerto paralelo, el intercambio de información con este es ligeramente complicado. Sin embargo, una vez entendidos los principios básicos de funcionamiento y con la lógica de programación adecuada, no debe presentar mayores problemas.

Estos 12 bits de salida y 5 bits de entrada están agrupados en 3 bytes, conocidos ampliamente en la literatura computacional como los bytes “Data”, “Status” y “Control”. Esto proviene del diseño original del puerto que, como ya se mencionó, fue hecho pensando en la impresora solamente.

El byte llamado “Data” contiene 8 bits de datos de salida que mandan el carácter a la impresora. El byte de “Control” es para darle órdenes y está formado por 4 bits solamente debido que los otros 4 no se utilizan, finalmente, el byte “Status” está formado por 5 bits de entrada que reciben información del estado de la impresora. Las siguientes figuras 3.5, 3.6, 3.7 muestran cómo están distribuidos estos bits en sus respectivos bytes. La configuración del puerto paralelo se caracteriza por estos tres bytes.

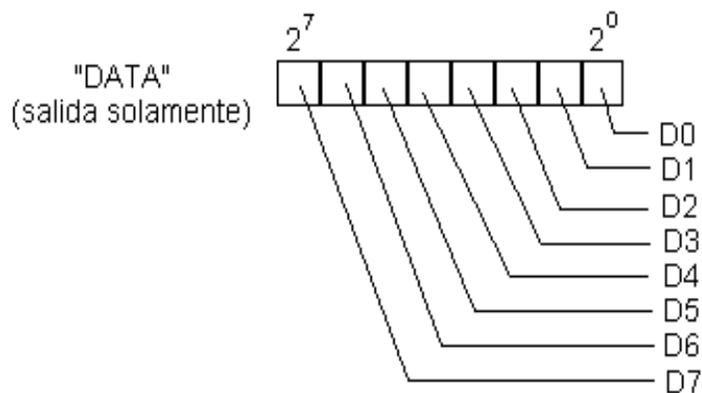


Figura 3.5 Distribución de bits del “Data Byte”

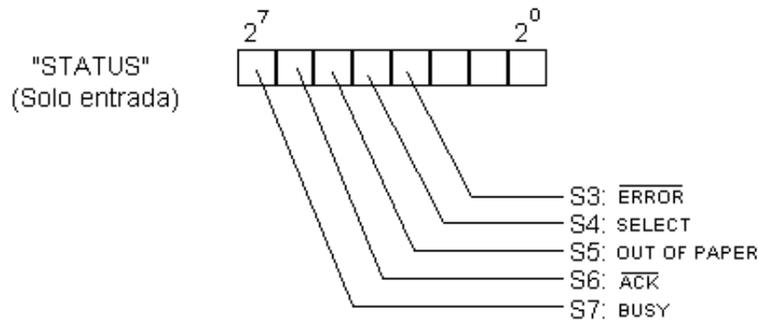


Figura 3.6 Distribución de bits del "Status Byte"

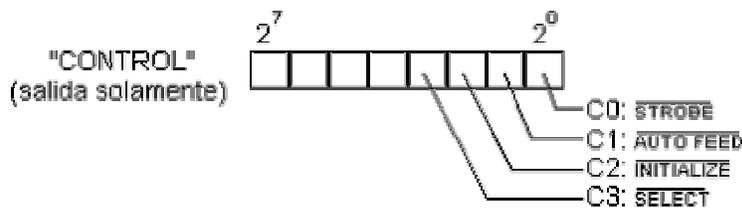


Figura 3.7 Distribución de bits del "Control Byte"

3.1.3 PortTalk –Manejador para los Dispositivos de I/O en Windows NT/2000/XP

Un problema que presentan los sistemas Windows NT/2000 y Windows XP es el no tener un estricto control sobre los puertos de entrada/salida (I/O), a diferencia de Windows 9x y ME. Windows NT/2000/XP causará una excepción (instrucción privilegiada) si detecta un acceso a los puertos de I/O generado por un programa hecho por el usuario, el cual no cuenta con privilegios para la comunicación, debido que cualquier procesador 80386 o superior ejecuta el programa en modo protegido.

El acceso a los puertos de I/O en modo protegido es posible realizarlo modificando el nivel de privilegios (I/O *privilege level*, IOPL), usando el registro de *EFLAGS*. Otro método es usando los permisos del mapa de bit del segmento de tareas (*Task State Segment*, TSS).

Bajo Windows NT, solo hay dos niveles de acceso para I/O, el nivel 0 y el nivel 3, en el modo de usuario, los programas corren en nivel 3, mientras que algunos manejadores de dispositivos y el kernel se ejecutan con privilegios de nivel 0 esto permite acceder a los

puertos, mientras que el usuario tiene menos privilegios para la comunicación con los I/O, ya que esto puede ocasionar conflictos en el sistema. Todo programa en el modo de usuario debe comunicarse por medio de un manejador de dispositivos.

Los permisos del mapa de bits pueden ser autorizados a los programas no privilegiados, para crear el acceso a los puertos I/O. Cuando el procesador ejecuta una instrucción primero checa si tiene privilegio para la comunicación, en caso afirmativo permitirá el acceso al puerto. Sin embargo, si la comunicación no es permitida por los privilegios, entonces revisara los permisos del mapa de bits.

La dirección de cada puerto esta representado en el mapa de bit, ahí se representa la dirección de cada puerto, si el bit correspondiente a un puerto esta activado, entonces la instrucción genera una excepción, sin embargo, si el bit no esta activado la operación procederá. Esto permitir que ciertos procesos puedan acceder a los puertos.

3.1.4 Acceso a los Puertos de I/O bajo Windows NT/2000/XP

Hay dos formas para el acceso de I/O bajo Windows NT/2000/XP. La primera es escribir un manejador para los dispositivos que corra en anillo 0 (privilegio de nivel 0). Así el manejador puede ejecutar la instrucción para el I/O, el problema con esto es que se asume que se tiene el código fuente para hacer dicho manejador.

La otra posible alternativa es modificar los permisos del mapa de bits para permitir una tarea en particular, acceda a cierto puerto. Esto concede que los programas corran en el nivel 3, seleccionado el puerto sin restricción. Este método realmente no se recomienda, pero proporciona un medio para que algunas aplicaciones puedan correr bajo Windows NT/2000/XP. Escribir un manejador de dispositivo es lo preferible para su hardware.

Sin embargo, usar el manejador *PortTalk* puede ser una solución eficiente. Cuando se hace una lectura o una escritura de un byte o una palabra del puerto, el proceso hace un cambio, del nivel 3 al nivel 0 y un cambio de regreso.

Un ejemplo semejante se puede encontrar en el USBLPTPD11[1].

El manejador de *PortTalk* viene con su código fuente. Proporcionando la facilidad para modificar el permiso de IO del mapa de bits y/o escribe y lee de los puertos.

3.1.5 Compatibilidad - Usando aplicaciones bajo Windows NT/2000/XP

PortTalk es usado en conjunción con *Allowio* para hacer programas que ejecuten el acceso a los puerto I/O bajo Windows NT/2000/XP. Se han desarrollado librerías de tipo DLL para Windows 9x y ME, sin embargo si corremos estos bajo Windows NT/2000/XP producirán una excepción. Para esto el *Allowio* y *PortTalk* ayudaran para lograr la comunicación hacia los puertos.

Los programas de 16Bits de Windows y los de DOS corren en máquinas virtuales. Existen muchas aplicaciones que son transparentes sobre Windows NT/2000/XP. No obstante hay otras que son rechazadas al momento de ejecutarlas. Las máquinas virtuales cuentan con el soporte para la comunicación de los puertos, video, ratón (*mouse*) y teclados. Cualquier programa que use los puertos comunes de I/O deberá correr, sin embargo eso puede ocasionar un problema con el reloj. En MS-DOS sus programas necesitan acceder al hardware específicamente por medio de manejadores de dispositivos virtuales (VDDs, *Virtual Devices Drivers*) escritos para ser usados con Windows NT[2][3].

La figura 3.8 muestra el esquema visual de la forma de trabajo del *PortTalk*.

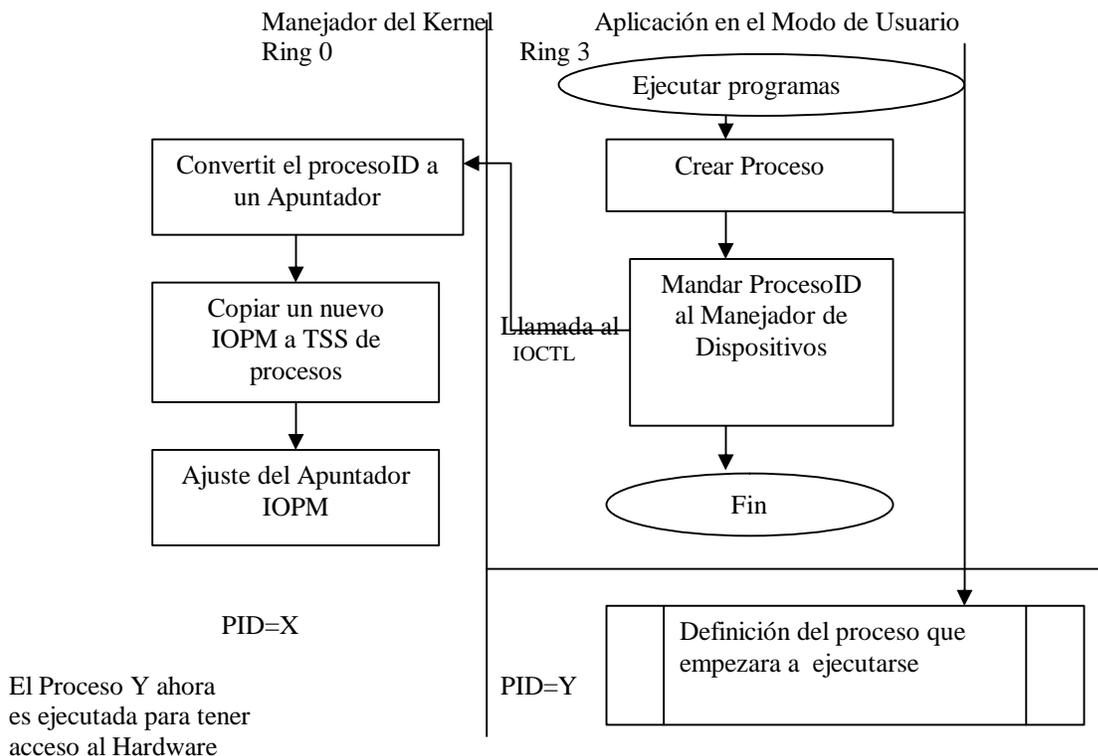


Figura 3.8 Esquema del trabajo del *PortTalk*.

3.2 El paquete *Lead Tools*

La otra herramienta externa que se utiliza es el paquete *Lead Tools*, que sirve para el manejo de imágenes, este contiene librerías VCL que consisten en varios paquetes. Cada paquete contiene uno o más comandos desarrollados por Lead Technologies, Inc., que son:

Paquete principal, LEADTOOLS VCL: Contiene el módulo del mando principal, *Lead Main* y *Lead Imaging* que es un módulo de diálogo común. El módulo del mando principal trabaja en 32-bit, proporcionando un control para la compresión de imágenes,

con formatos de archivos para estas, haciendo más fácil el trabajo de los procesamientos de imágenes.

El paquete adicional LEADTOOLS VCL: Contiene el modulo de control llamado *Lead ImageList*, *Lead ThumbnailBrowser* y *Lead Screen*. *Lead ImageList* (32-bit solamente) proporciona un control para desplegar y manipular listas de la imágenes. *Lead ThumbnailBrowser* (32-bit) proporciona un control para desplegar imágenes desde un archivo. *Lead Screen Capture* proporciona controles para realizar capturas como: la pantalla entera, ventana activa, el menú seleccionado, menú bajo el cursor, el objeto de la ventana seleccionado (botón, barra de la herramienta, icono, o área del cliente), el área seleccionada (rectángulo, elipse, triángulo, polígono o a pulso), indicador del ratón. Las opciones incluyen multi-captura, *hot keys* , e intervalo de tiempo [5].

El paquete LEADTOOLS VCL Scanning: contiene *Lead TWAIN* y *Lead ISIS*. *Lead TWAIN* modulo de control de 32-bit, proporciona el controlador para escáner TWAIN, para las aplicaciones. *Lead ISIS* (32-bit) proporciona controles ISIS para el mando del escáner en sus aplicaciones.

Se puede usar el LEADTOOLS VCL en los compiladores de Borland:

Delphi 4.0 (32-bit)

Delphi 5.0 (32-bit)

C++ Builder 4.0 (32-bit)

C++ Builder 5.0 (32-bit)

3.2.1 Introducción a los Bits por Píxel

La terminología de los formatos de las imágenes puede ser confusa porque, a menudo, hay diversas formas para describir el mismo formato.

Si una imagen contiene 24 bits por píxel, viene llamada imagen de 24 bits, imagen de color verdadero, o una imagen de dieciséis millones de colores porque es el número que se puede representar con 24 bits, donde ocho bits son de rojo, otros ocho para verde y otros ocho para azul (RGB, *red, green, blue*).

Una imagen de 32 bits por píxel, es una imagen especializada en color real, donde el byte extra contiene información que puede ser convertida o ignorada cuando la imagen es cargada. El byte extra es usado para la transformación de un archivo RGB a un archivo CMYK (*Cyan, Magenta, Yellow, Black*), en color verdadero para las impresoras de color. LEADTOOLS, por defecto convierte un archivo de 32bits a uno 24 bits .

Si una imagen es de 16 bits por píxel es llamada imagen de 16 bits, imagen de color alto o color de imagen de treinta dos mil colores, porque se pueden representar treinta y dos mil colores con 16 bits, para mantener la uniformidad de los colores se le asignan 5 bits para cada uno de los colores, rojo, verde, azul y el último bit viene ignorado.

Si la imagen es de 8 bits por píxel, es llamada imagen de 8 bits o imagen de 256 colores.

Si la imagen es de 4 bits por píxel, es llamada imagen de 4 bits o de 16 colores.

Si la imagen es de 1 bit por píxel, es llamada imagen de 1bit o binaria y solo tiene dos posibles colores que son el negro o el blanco.

En una imagen en escala de grises, los valores correspondientes a los valores rojo, verde y azul vienen unificados y pueden tomar cualquiera valor entre 0 y 255 produciendo una imagen con 256 posibles tonos de gris (el rojo, verde y azul adquieren el mismo valor), de esta forma cada píxel viene representado por un byte. LEADTOOL soporta las siguientes escalas de grises: 4, 8,12 y 16 bits por píxel logrando un máximo de sesenta y cinco mil tonos de gris con 16 bits.

3.2.2 *Lead Image y Lead Twain*

En el sistema se utilizaron dos herramientas de LEADTOOLS, una es *LeadImage* , que sirve para desplegar y manipular las imágenes y la otra es *Lead Twain*, la cual ayuda para la captura de imágenes por medio de una cámara digital.

La ventana de dialogo de *LeadImagen* para el manejo de imágenes viene suministrada con librería que combina la ventana de diálogo de Windows con funciones como: abrir archivo, salvar archivo, resolución de la imagen, filtros y efectos.

Como la ventana de diálogo contiene varios métodos, se enumera solo los más usados por el sistema que son:

? Método *Load* : carga una o más imágenes desde un archivo que puede ser de cualquier formato y puede contener cualquier cantidad de bits por píxel, comprime o expande las imágenes. Cuando carga una imagen de un archivo, LEADTOOLS normalmente detecta el formato principal. Si el formato no es reconocido, regresa un error a la apertura.

? Método *Save*: salva el mapa de bits a un archivo en cualquier formato de compresión soportado.

? Método *Rotate*: este método rota la imagen según el ángulo especificado.

LEADTOOLS *Twain* soporta imágenes de 16 y 32 bits traídas desde un archivo o almacenadas desde un buffer de la memoria RAM. El programador tiene la opción de utilizar la fuente predefinida cargada por los controladores de *Twain* o también se puede seleccionar el controlador *Twain* con el que se desee trabajar cuando se dispone de varios. El control de LEADTOOLS *Twain* ofrece las opciones de controlar el tamaño de la imagen, la posición, el brillo, el contraste, la resolución y la orientación.

El Twain de LEADTOOLS soporta dispositivos de escáner, cámaras digitales como Kodak, Hewlett-Packard, Microtek y Fuji entre otras.

3.3 El submódulo de Hardware

El Digitalizador esta compuesto por una estructura de metal en la cual se encuentran las bases, la del objeto y el láser-cámara, cada base es controlada por un motor a pasos, estos motores a su vez son controlados por el submódulo de software el cual se encargara de activarlos y dar la dirección de giro, también por ese submódulo se controlara la captura del la imagen emitida por la cámara. La base del láser-cámara esta sostenida en un brazo el cual esta en un barra con cuerda (un tornillo), al girar este sube o baja al brazo dependiendo de la dirección de giro del motor. En la figura 3.9 se observa la parte física del Digitalizador.

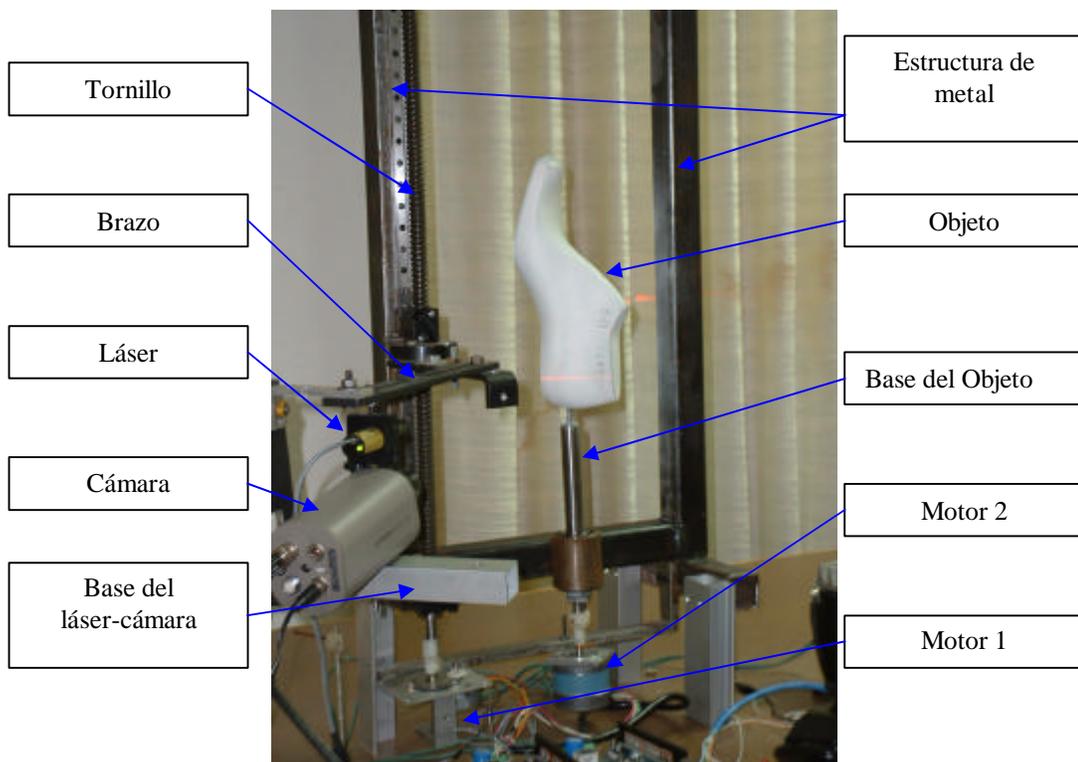


Figura 3.9 Partes físicas que componen al Digitalizador

3.3.1 El láser-cámara

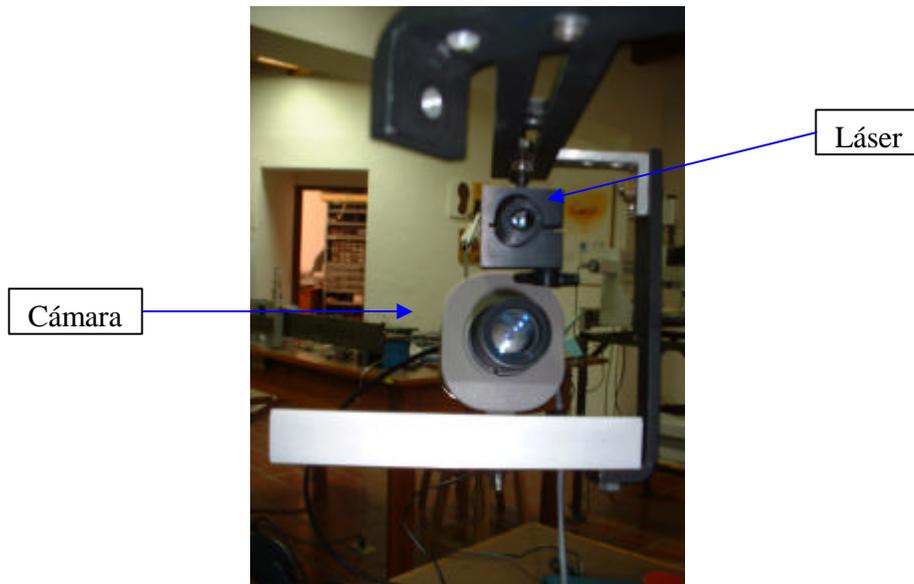


Figura 3.10 El láser-cámara

En la figura 3.10 se muestra el láser-cámara. El láser utilizado para llevar a cabo el desarrollo del sistema es un láser de *Lasiris Inc* de tipo Snf laser, figura 3.10a el cual cuenta con las siguientes características principales[20]:

- ? Potencia del Diodo: de 1 a 150 mW
- ? Longitud de onda : 635nm
- ? Ángulos posibles : 1 a 120°
- ? Voltaje de entrada : 5 a 6 Vcd
- ? Impedancia : >1 K? ?
- ? Frecuencia : 10KHz



Figura 3.10a Imagen del láser(cortesía de Stockeryale[20])

3.3.2 Selección de la Cámara para el Digitalizador

Para el Digitalizador, se experimentó con tres diferentes cámaras, para buscar cual ofrecía las mejores prestaciones.

Una de las cámaras fue una cámara web (webcam) de marca Prochip, figura 3.11, con la cual se podía capturar una imagen con resolución de 320 x 240 pixeles, monocromática, se conecta a la computadora por medio de un puerto USB y el tamaño del archivo resultante es aproximadamente 4.5 Kbytes.



Figura 3.11 Cámara Prochip

Otra utilizada fue una cámara digital de Cannon, serie G2 Power Shot [6], figura 3.12, cuyas características son las siguientes:

- ? 4-megapixel CCD con filtro de color primario RGB.
- ? Tamaño de la Imagen de 2272x1704, 1600x1200, 1024x768 y 640x480
- ? Compresión de imagen de tipo JPEG con 3 modalidades(Superfino, Fino y Normal) en cualquier resolución
- ? Canon 3x optical f/2.0-2.5 zoom lente de 34-102mm en formato de 35mm
- ? LCD de1.8-inch, ángulo variable, de color
- ? ISO 50, 100, 200 ó 400
- ? Velocidad de disparo de 15 seg. a 1/1000 seg.
- ? Tamaño del archivo aproximado por Imagen : 385k
- ? La toma de las Fotos utiliza en este caso su propio software
- ? Descarga de las Imágenes por puerto USB



Figura 3.12 Cámara Canon (cortesía de Steve´s Digicams[6])

Por último se hicieron pruebas con la cámara COHU, figura 3.9, la cual resultó ser la más adecuada para las necesidades del sistema por esa razón se muestran sus características técnicas:

ELECTRICAL	
Imager	Single CCD using frame transfer method
Image Area	8.8 by 6.6 mm (2/3 inch format)
Active Picture Elements	754 (H) by 488 (V)
Number of Picture Cells	780 (H) by 244 (V)
Cell Size	11.5 μm (H) by 27 μm (V)
Resolution	<ul style="list-style-type: none"> • Horizontal: 565 lines • Vertical: 350 lines
Sensitivity	See table 1a
Contrast Variation, 25 °C	<5% overall
Scanning System	RS-170 2:1 Interlaced
Video Output	1.0 Vp-p, 75 ohms unbalanced
Gamma	0.5 or 1.0, jumper selected
Gray Scale	Renders all shades of gray on EIA tv resolution chart, 1956
Agc	6 dB variable gain (peak-average adjustable); jumper selectable on/off
Auto Lens	Peak/average adjustable
Signal to Noise Ratio, 25 °C	<ul style="list-style-type: none"> • 50 dB (gamma 1, gain 0 dB), unweighted 8 MHz bandwidth • 55 dB (gamma 1, gain 0 dB), weighted
Automatic Black Level	Maintains setup level at 7.5 \pm 5 IRE units if picture contains at least 10% black
Synchronization	<ul style="list-style-type: none"> • EIA RS-170 crystal, 14.31818 MHz clock output standard • Genlock, with crystal or line lock backup (jumper selectable) • External H and V drive
Power Options	<ul style="list-style-type: none"> • 12 V dc/ac \pm10%, 60 Hz ac • 24 V dc/ac \pm10%, 60 Hz ac • 115 Vac 60 Hz, \pm10% (with optional wall transformer)
Power Consumption	4.2 Watts

Debido a que la cámara tiene una salida de video no compatible con la computadora, figura 3.14, se necesita utilizar una interfase para la captura de la imagen y su almacenamiento en la computadora. Esta cámara también ofrece imágenes monocromáticas con un tamaño aproximado del archivo de 30.6K

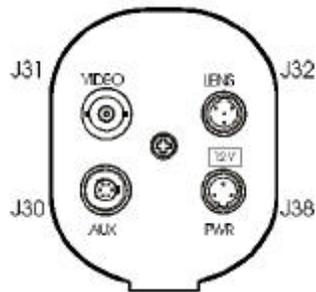


Figura 3.14 Esquema de los conectores en la parte posterior de la cámara COHU

Esta interfase es una Pinnacle, figura 3.15, que recibe la señal de video de la cámara y la transforma a un formato compatible a la computadora transfiriéndola a través de un puerto USB, por medio de un controlador *Twain* el tamaño de la imagen se convierte a 1600 x 1200 píxeles y es almacenada en un archivo cuyo tamaño es de 44.5 Kbytes



Figura 3.15 Interfase Pinnacle utilizada entre la cámara y la computadora.

3.3.3 El sistema de motores

Para el Digitalizador de Objetos se usaron dos motores a pasos para hacer el desplazamiento del láser-cámara y la rotación del objeto.

Para el funcionamiento de los motores a pasos se utilizó la comunicación del puerto paralelo, encargada de dar los pulsos necesarios para el movimiento. Se utilizaron para cada motor unos drivers de potencia.

El motor encargado de rotar el objeto tiene las siguientes características [21]:

- ? Alimentación 5 Vcd

- ? Corriente 3 Amp
- ? Resistencia 1.5?
- ? Inductancia 17 mH
- ? Unipolar de 6 líneas
- ? 200 pasos para que el objeto complete 360°

El motor encargado de mover el láser-cámara tiene las siguientes características[21]:

- ? Alimentación 5 Vcd
- ? Corriente 2 Amp
- ? Resistencia 1.7? ?
- ? Inductancia 11 mH
- ? De 6 líneas
- ? 350 pasos para completar un desplazamiento lineal de 0.5cm

Algunas de las características de los drivers de potencia son[22]:

- ? Alimentación de 24 a 40 Vcd
- ? Salida de Corriente ajustable de 0.5 a 3.5 Amp
- ? Impedancia de entrada de 1K?
- ? Señal de entrada 5Vcd
- ? Corriente de señal de entrada 1mA
- ? Nivel lógico bajo de 0 a 2 Vcd
- ? Nivel lógico alto de 3 a 5 Vcd

Terminales de señal de entradas usadas:

- ? *Dir* : Selecciona la dirección en que tiene que girar el motor
- ? *Run* : Cuando esta señal se activa, mueve el motor en la dirección seleccionada
- ? *AWO*: Esta señal desactiva, la corriente del motor

circuito [22].

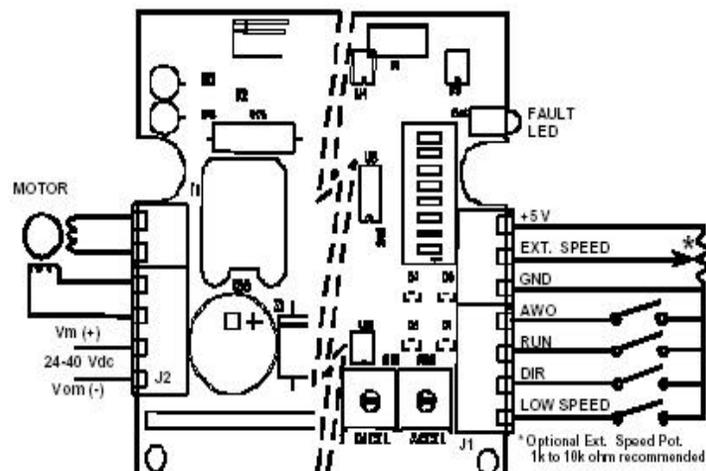


Figura 3.16a Esquema del Driver



Figura 3.16b Vista físicamente

En la figura 3.16a se muestra el diagrama de bloques del driver, en la figura 3.16b se presenta una imagen real del driver.

Las señales de entrada, *Dir*, *Run* y *AWO* son activadas por medio del módulo software utilizando el puerto paralelo para la comunicación. Antes de que los bits escritos por el puerto paralelo lleguen a los motores se utiliza una pequeña interfase de hardware entre ellos, esta interfase está compuesta por un TTL(74LS04), los bits llegan primero al TTL, este tiene la función de separar el puerto paralelo de los drives de potencia, para que ninguna corriente generada por estos, regrese al puerto ocasionando algún daño al equipo. Las salidas del TTL están conectadas a las señales de entrada de los drivers de potencia, cada uno de ellos utiliza solo tres bits, un bit es para *AWO*, otro es para *Dir* y uno más para *Run*, en la figura 3.17 se muestra el esquema en que fue conectado el puerto paralelo a los drivers de potencia y estos a los motores, en la figura 3.17a,b se presentan el conjunto de los drivers y la interface.

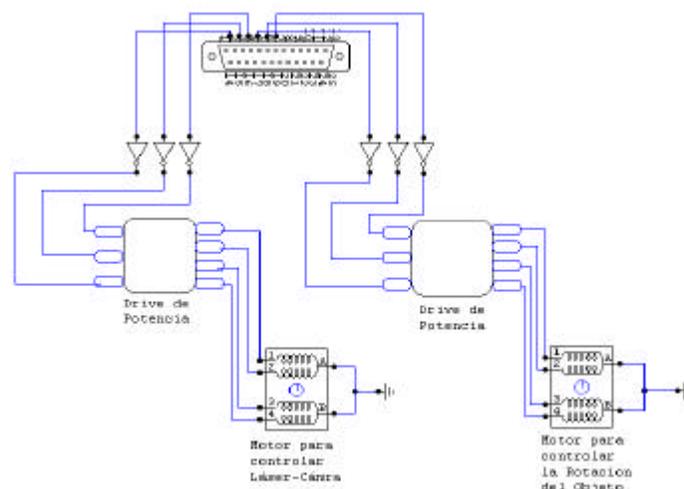


Figura 3.17a Comunicación del puerto paralelo hacia los motores

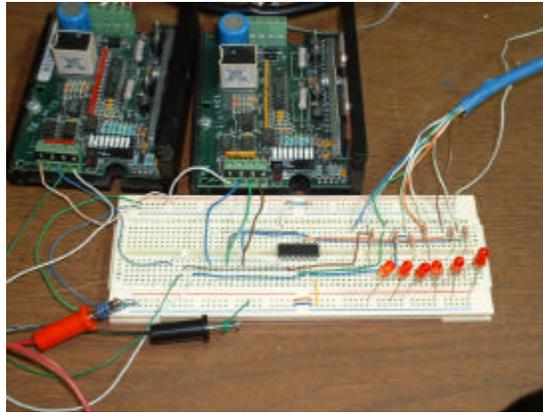


Figura 3.17b Imagen del dirver de potencia y los conectores hacia el puerto paralelo.

3.4 Submódulo Software

Después de la descripción de las partes electrónicas, mecánicas y físicas que compone al Digitalizador, se hablara del desarrollo del Software.

Para tomar las fotos, el láser-cámara estará en una posición inicial, iniciando un ciclo de fotografías, este ciclo consiste en tomar una seria de imágenes en esa posición y se completa cuando el objeto haya girado los 360° o ninguno dependiendo de los datos dados por el usuario. Una vez que el ciclo se termine la posición del láser-cámara cambia desplazándose una distancia de 0.5 o 1.0cm de pendiendo de la selección del usuario y el ciclo vuelve a comenzar, esto se repite hasta cubrir toda la altura del objeto.

3.4.1 Captura de imagen

Se desarrollo el software con el cual la imagen capturada se guarde en disco, de esta imagen se necesita saber ciertos datos los cuales están escritos en el nombre de esta, el nombre se compone de la siguiente información; la palabra *imagen*, seguido de guión bajo “_”, después tiene un número el cual indica la posición del láser-cámara (la posición inicial se toma como 0), eso ya que el objeto se puede girar y se necesita saber la posición de esas imágenes tomadas, luego de ese número se escribe otro guión bajo “_”, despuse se escribe el grado de rotación que tiene el objeto en el instante que se tomo la imagen, se escribe otro guión bajo “_” y por último se escribe el número de la imagen tomada, para el total de imágenes se suma 1 esto se debe que la serie de imágenes comienza desde 0, y, la extensión es JPEG, ya que se pide como requisito del sistema que las imágenes guardadas en disco no ocupen mucho espacio. Por ejemplo un nombre de una imagen quedaría de la siguiente forma:

Imagen_3_90_26.jpg

? *Imagen*, es el nombre con que comienza.

- ? 3, se encuentra en la posición que indica el número de veces que se ha desplazado el láser-cámara, todas las imágenes tomadas en esta posición se llamarán *rebanada del objeto*, también dice cuantos ciclos de fotografías lleva, en este caso indica que ya lleva cuatro ciclos de fotografía, dado que se suma 1 al total.
- ? 90, se encuentra en la posición que indica el grado donde estaba el objeto cuando se tomó esa imagen, en este caso su posición actual es de 90°, es decir que el objeto ya giró 90°.
- ? 26, se encuentra en la posición que indica el número de fotos tomadas hasta el momento, en este caso es la 26.
- ? .jpg, es la extensión en que se guarda la imagen.

3.4.2 Submódulo control de motores

Una vez guardada la imagen en disco se escribe por el puerto paralelo un dato el cual llega al driver de potencia y activa el motor de pasos que se encarga de hacer girar al objeto, esto se repite el número de veces necesario para hacerlo girar el ángulo deseado, puede girar de 45° en 45° o de 90° en 90° o 180° en 180°, hasta completar los 360° o bien puede seleccionar que no gire, una vez tomada la imagen completa se considera que llego a los 360°, al completar un giro del objeto el ciclo de la toma de imágenes se termina y ahora se manda un dato por el puerto paralelo el cual llega al otro driver de potencia y este activa el motor encargado de mover el láser-cámara, esto se repite hasta completar el desplazamiento vertical deseado que puede ser 0.5 o 1.0 cm, cada vez este proceso de toma de las imágenes se repite hasta completar la altura del objeto teniendo así rebanadas de este.

3.4.3 Submódulo control de motores auxiliar

Antes de iniciar el proceso de toma de fotografías, quizá se necesite colocar en cierta posición el láser-cámara o el objeto, así que el submódulo control de motores auxiliares da la libertad de mover los motores paso a paso, se tiene por inicio 3 pasos el cual se puede cambiar y el usuario dará el número de pasos, la dirección hacia donde gire. El control de cada motor es independiente uno del otro, así que se requiere que se den los datos para uno y para otro. Una vez que se haya dejado en la posición deseada tanto el láser-cámara como al objeto se podrá comenzar el proceso de la toma de fotografías.

3.5 Proceso de captura de imágenes

Al seleccionar la opción Fotografiar en el menú principal, figura 1.2, se abrirá una ventana, figura 3.18, en la cual el usuario dará los siguientes datos:

- ? Ubicación, se hace un llamado al módulo UBICACIÓN ya descrito.
- ? La altura del objeto, se da la altura del objeto en múltiplos de 0.5cm esto se debe que la base del láser-cámara se desplaza con esa distancia, este dato se utiliza para determinar los ciclos de captura de imagen.

- ? Distancia entre fotografía, el usuario selecciona la distancia entre 0.5 y 1.0cm que debe tener la rebanada del objeto.
- ? Grados, se selecciona los grados que rotara el objeto para crear la rebanada.
- ? Fotografiar, empieza con el proceso de captura de imagen del objeto.
- ? Auxiliar Motores, abre otra ventana, figura 3.19, permite ajustar la posición de inicio del láser-cámara y del objeto.
 - o Pasos del Motor 1, el usuario da el número de pasos que quiere girar el obejeto.
 - o Giro_der, gira el objeto hacia la derecha
 - o Giro_izq, gira el objeto hacia la izquierda.
 - o Pasos del Motor 2, el usuario da el número de pasos que quiere que se mueva el láser-cámara.
 - o Arriba, desplaza el láser-cámara hacia arriba.
 - o Abajo, desplaza el láser-cámara hacia abajo.

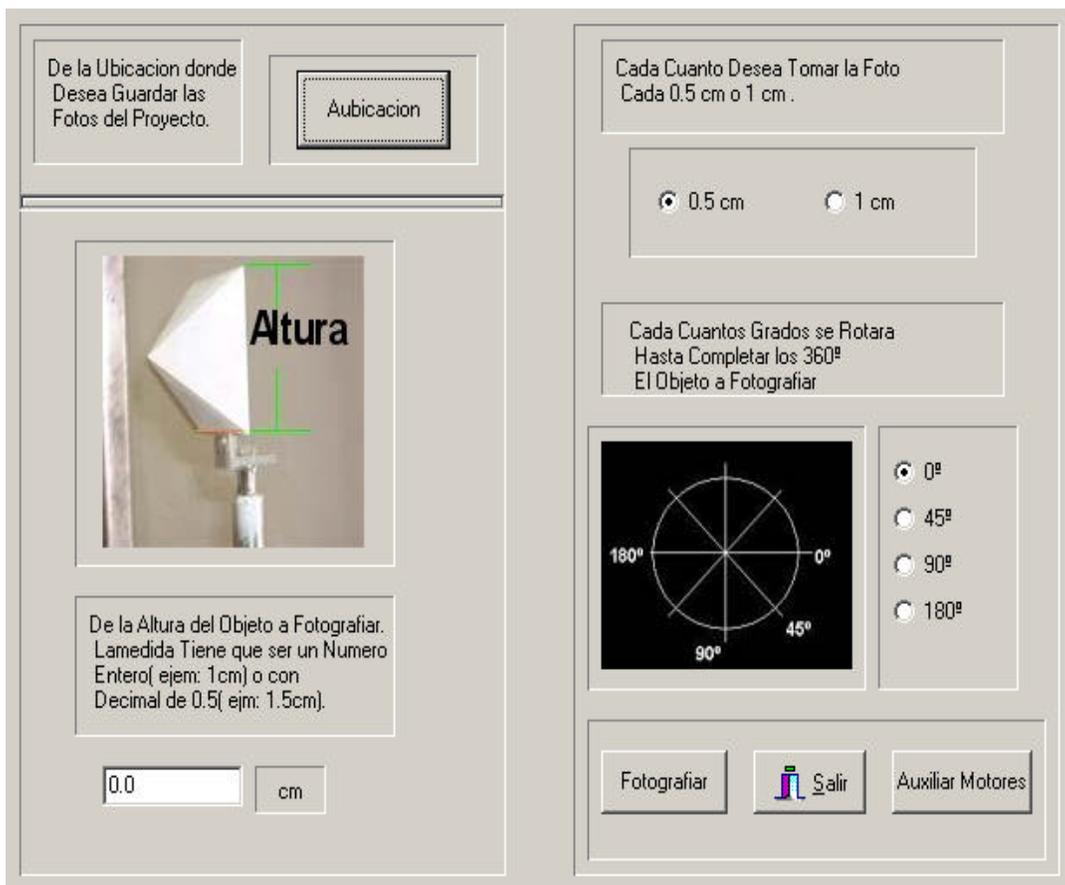


Figura 3.18 Ventana del módulo de Fotografiar.

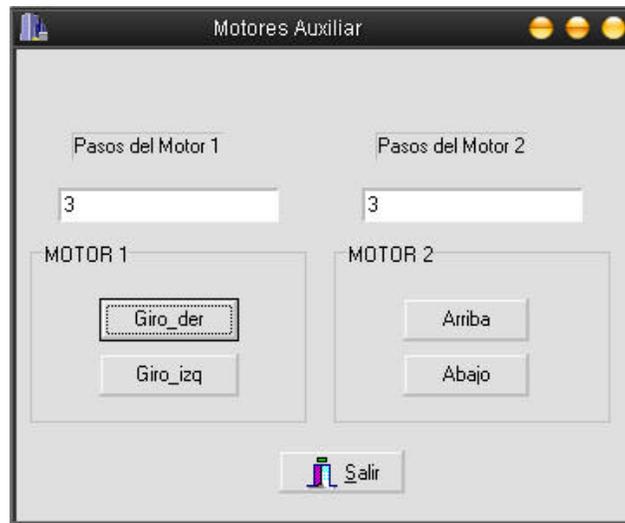


Figura 3.19 Ventana de la opción de Auxiliar Motores

Una muestra de una imagen a capturar se ve en la figura 3.20, en el cual se ve el objeto que refleja la luz láser sobre el, en la figura 3.21 se ve la imagen guardada en disco que después pasara el Filtro de Gabor.

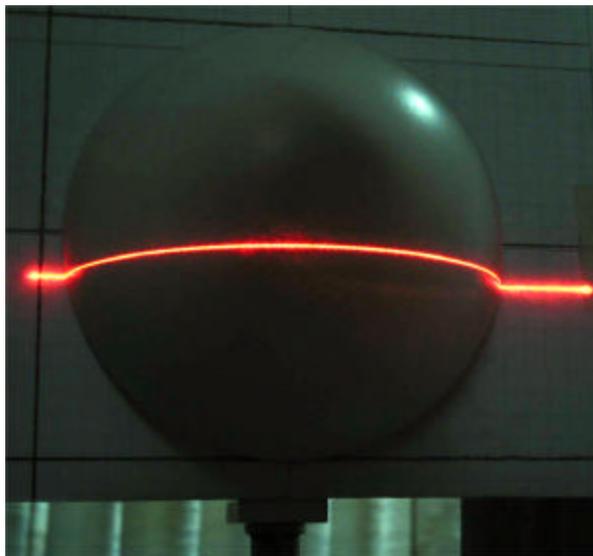


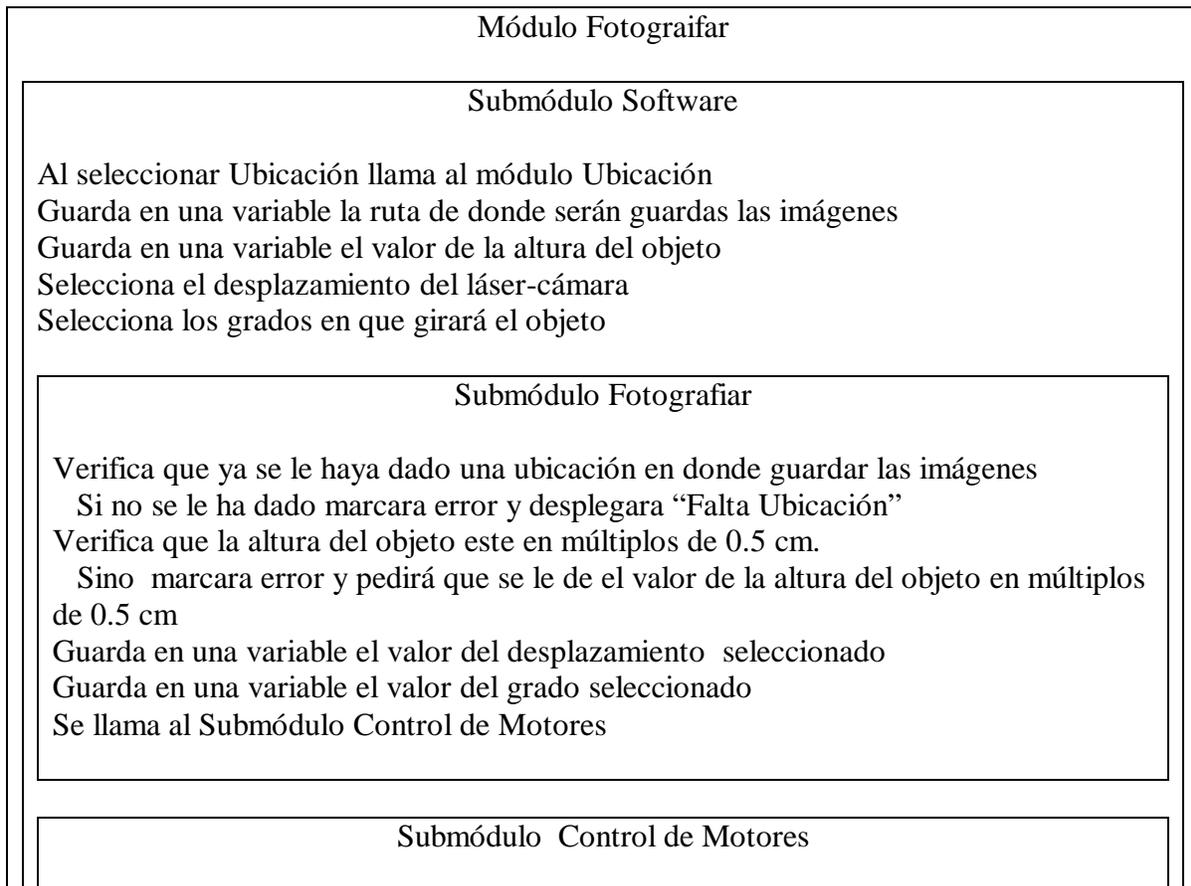
Figura 3.20. Imagen de una media esfera



Figura 3.21. Imagen capturada de una media esfera con nombre imagen_3_0_3.jpg

3.6 Pseudocódigo del módulo FOTOGRAFIAR

Dado que el módulo FOTOGRAFIAR [8][9], es muy extenso se presenta un fragmento de este.



Se copia en una variable la dirección en donde se guardarán las imágenes
Se pone una variable que cuente el número de imágenes, igual a 0 al inicio
De la variable que guardó los grados se busca sus nuevos datos
Si tiene el valor 0 entonces no envía el dato al motor y hace una captura de imagen
Si tiene el valor 45 entonces tiene que enviar el dato al motor 25 veces y hacer 8 capturas de imagen
Si tiene el valor 90 entonces tiene que enviar el dato al motor 50 veces y hacer 4 capturas de imagen
Si tiene el valor de 180 entonces tiene que enviar el dato al motor 100 veces y hacer 2 capturas de imagen
Se pone una segunda variable de altura igual a 0
Se pone una variable de repetición de ciclo igual a 0
Ciclo1: se repite tantas veces tenga el valor de la variable altura
Ciclo2: se repite tantas veces tenga el valor de capturas de imagen
Verifica si la interfase de la cámara esta activada, si no desplegar mensaje de que la interfase no se encontró
Se captura la imagen
Se crea el nombre del archivo
Se guarda en disco la imagen con el nombre creado
Se llama el módulo mover motor1
Se incrementa la variable de repetición
Se incrementa la variable que lleva la cuanta de las imágenes
Se termina el ciclo2
Se llama al módulo mover motor2
Se termina ciclo1

Submódulo mover motor1

Ciclo1: se repite tantas veces tenga que enviar el dato al motor
Escribes en el puerto paralelo el número en hexadecimal 6f
Espera 20 milisegundos
Escribe en el puerto paralelo el número en hexadecimal 2f
Espera 20 milisegundos
Se termina el ciclo1

submódulo mover motor2

Ciclo1: se repite tantas veces tenga que enviar el dato al motor
Escribes en el puerto paralelo el número en hexadecimal f2
Espera 20 milisegundos
Escribe en el puerto paralelo el número en hexadecimal f6
Espera 20 milisegundos
Se termina el ciclo1

Termina el submódulo de Control de motores

Submódulo Motores Auxiliares

Se guarda en una variable el valor que se da para el motor1

Se guarda en una variable el valor que se da para el motor2

Submódulo Giro_der

Ciclo1: se repite tantas veces tenga el valor que se da para el motor1

 Escribes en el puerto paralelo el número en hexadecimal 0f

 Espera 20 milisegundos

 Escribe en el puerto paralelo el número en hexadecimal 4f

 Espera 20 milisegundos

Se termina el ciclo1

Submódulo Giro_izq

Ciclo1: se repite tantas veces tenga el valor que se da para el motor1

 Escribes en el puerto paralelo el número en hexadecimal 6f

 Espera 20 milisegundos

 Escribe en el puerto paralelo el número en hexadecimal 2f

 Espera 20 milisegundos

Se termina el ciclo1

Submódulo Arriba

Ciclo1: se repite tantas veces tenga el valor que se da para el motor2

 Escribes en el puerto paralelo el número en hexadecimal f2

 Espera 20 milisegundos

 Escribe en el puerto paralelo el número en hexadecimal f6

 Espera 20 milisegundos

Se termina el ciclo1

Submódulo Abajo

Ciclo1: se repite tantas veces tenga el valor que se da para el motor2

 Escribes en el puerto paralelo el número en hexadecimal f4

 Espera 20 milisegundos

 Escribe en el puerto paralelo el número en hexadecimal f0

 Espera 20 milisegundos

Se termina el ciclo1

Termina el submódulo Motores Auxiliar

Termina módulo Fotografiar
Regresa el control al módulo PRINCIPAL

Capítulo 4 : Módulo Filtro de Gabor

El Digitalizador está basado en el procesamiento de líneas proyectadas por el láser y para llegar a hacer una calibración más exacta entre el objeto real y la imagen desplegada en pantalla se necesita que la imagen capturada se pase por un filtro, esto debido a que la línea gruesa describe la superficie del objeto, la ubicación de la coordenadas es importante ya que luego se utilizarán para reconstruir el objeto. Esta línea gruesa que esta formada por un conjunto de píxeles, se tiene que adelgazar, así que es necesario identificar dentro de ese conjunto un punto, las coordenadas del píxel que represente esa línea, la imagen debe estar en una escala de grises que es de 8 bits (0 a 255 niveles de gris).

Una forma para adelgazar la línea es encontrando el valor de intensidad máximo del tono de gris en la imagen y tomar éste como la coordenada donde pasa la línea, pero este método incurre en errores ya que si la región abarca algunos píxeles no se sabría que píxel elegir, si se calculara el promedio de la coordenada de estos píxeles se incurriría en un error porque dentro de la misma región pueden existir pequeñas variaciones en la intensidad de gris lo que tampoco sería muy exacto. Así que se utilizó un método más robusto, los *Filtros de Cuadratura*, estos son una alternativa para obtener un par de señales que convolucionan en una región de la imagen, se llaman en cuadratura debido a que las señales del filtro difieren solamente por un corrimiento de $\pi/2$, la salida de este filtro es semejante a una función analítica, sin embargo la magnitud de la función de transferencia no es unitaria y puede tomar cualquier valor real. El filtro más conocido es el filtro pasa banda de Gabor, este filtro determina el valor máximo, dando la posición en donde se encuentre el cruce por cero de la fase obteniendo con ello una precisión de subpíxel [10].

Los componentes del módulo Filtro de Gabor se ven en al figura 4.1

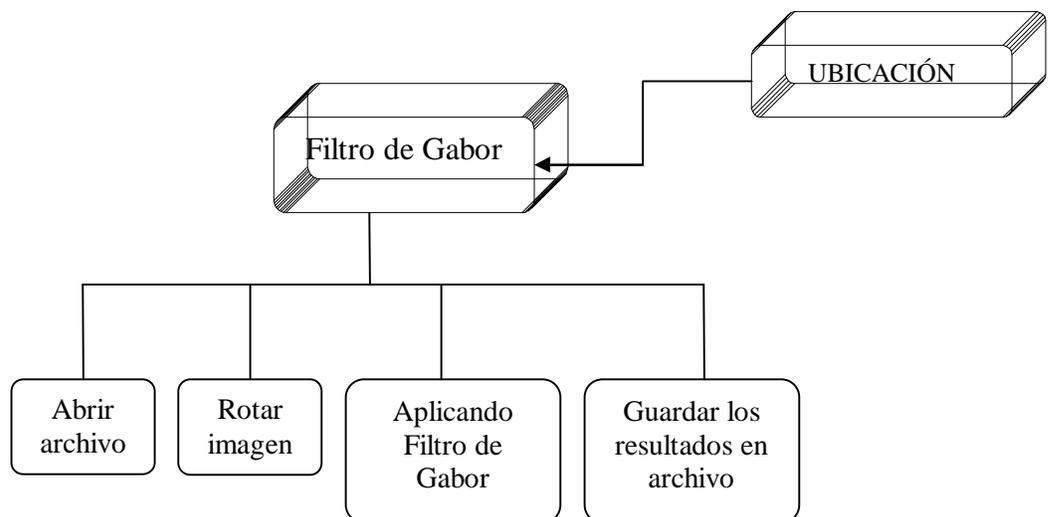


Figura 4.1 Esquema del Módulo Filtro de Gabor

Dado que el sistema hasta el momento es incapaz de convertir por si solo una imagen a escala de grises, se necesitó de un programa externo para que hiciera este trabajo

como puede ser el Adobe Photoshop o el Paint Shop, esto debido que, el paquete con el cual se manejan las imágenes, *Lead Tools*, no cuenta con esta característica.

Antes de seleccionar la opción Filtro de Gabor , figura 1.2, hay que seleccionar la opción Ubicación esto es porque el módulo Filtro de Gabor necesita saber la ubicación de las imágenes a analizar lo hará para todas las imágenes que se encuentren en esa dirección.

Una vez realizado el módulo de UBICACIÓN se podrá seleccionar la opción Filtro de Gabor el cual abrirá una ventana nueva [11], figura 4.2, en la que desplegará la imagen a la que se le aplicara el filtro.



Figura 4.2 Ventana de la opción Filtro de Gabor

4.1 Submódulo Abrir Archivo

Este submódulo buscará de la dirección obtenida del módulo UBICACIÓN todos los archivos de extensión *jpeg* , se utiliza la herramienta de *Lead Tools* , carga esta imagen en un mapa de bits desplegándola en pantalla[11], ver figura 4.3.

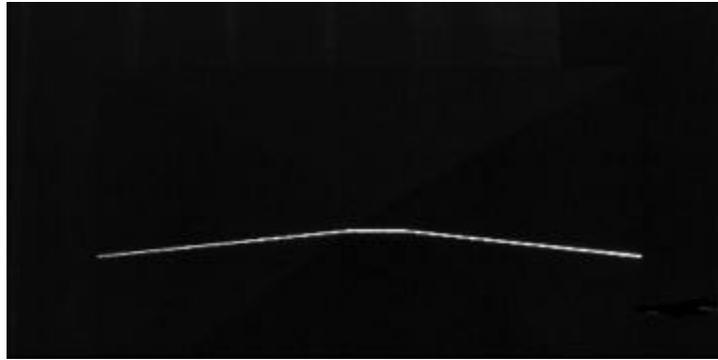


Figura 4.3 Imagen abierta de la proyección de una línea de una pirámide

4.2 Submódulo Rotar la imagen

Una vez que la imagen ya fue cargada y desplegada en pantalla se procede a rotar la imagen 90° hacia la derecha utilizando una de las herramientas de *Lead Tools*, para luego poder aplicar el *Filtro de Gabor* [11], ver imagen 4.4.

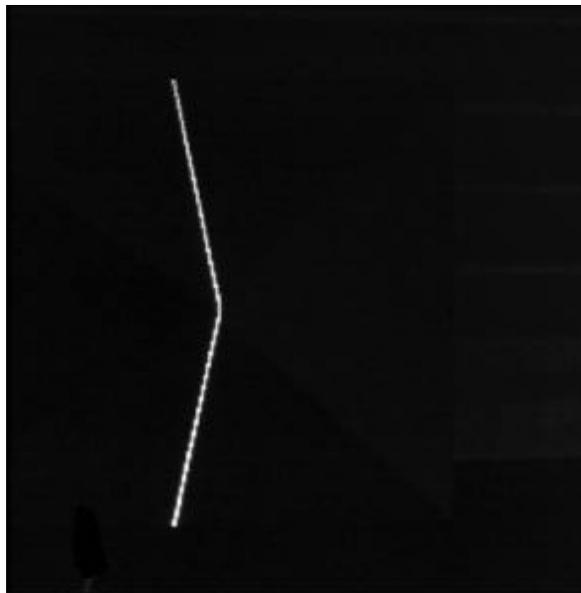


Figura 4.4 Imagen girada 90°

4.3 Submódulo Aplicando Filtro de Gabor

Esta parte del Digitalizador se basó en un programa ya desarrollado para el Filtro de Gabor[23], el cual tenía que ejecutarse cada vez que se analizaba una imagen, se tomó la parte donde se aplicaba el Filtro de Gabor y de este solo se le hicieron los cambios necesarios; uno de los cambios mas importantes que se le hizo fue que trabajará continuamente ya que en al anterior al abrir y cerrar el programa utilizaba bloques de memoria RAM y al terminar estos bloques eran marcados como ya utilizados, aunque

fueran liberados y no se podían reutilizar llegando así a un punto donde el sistema se paraba por falta de memoria, esto sucedía cuando era un conjunto grande de imágenes a analizar. Una vez arreglado este punto importante para el sistema y hecho las modificaciones correspondientes se puede aplicar el Filtro de Gabor a la imagen.

4.3.1 La forma de analizar las imágenes

Ya que la imagen se encuentra girada a 90° , se le aplica el Filtro de Gabor, este analiza cada renglón de píxeles de la imagen, ver figura 4.5, sacando la coordenada del valor máximo de ese renglón, en ocasiones esa coordenada esta dada en subpixel.

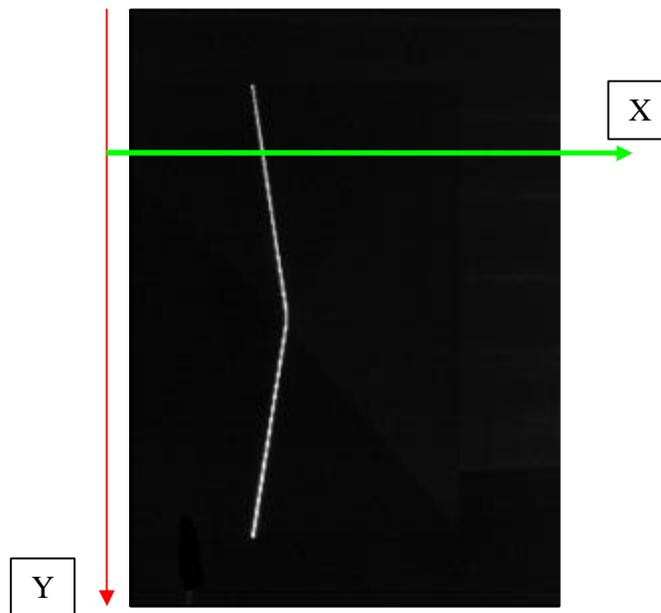


Figura 4.5 Barrido de la imagen

El filtro empieza a realizar un barrido en la coordenada (0,0) luego analiza el primer renglón de X de 0 hasta N donde N es el número máximo de píxeles en X, guardando en una lista el resultado del filtro, esto se repite para cada Y de la imagen.

4.4 Submódulo Guardar los resultados en archivo

Una vez analizada la imagen y que los resultados están almacenados en una lista, lo siguiente que hay que hacer es guardar esos resultados en un archivo.

El archivo tendrá el mismo nombre de la imagen, esto es para identificar el resultado de cada imagen, con extensión TXT y serán guardados en la misma dirección

donde se encuentren las imágenes, el contenido de este archivo tendrá el siguiente formato: se escribirá en el archivo el numero real de la coordenada con cuatro cifras decimal, es luego para separar una cifra de otra se escribirá una coma “,” y así se escribirán todos los datos contenidos en la lista al archivo, ver figura 4.5.

```

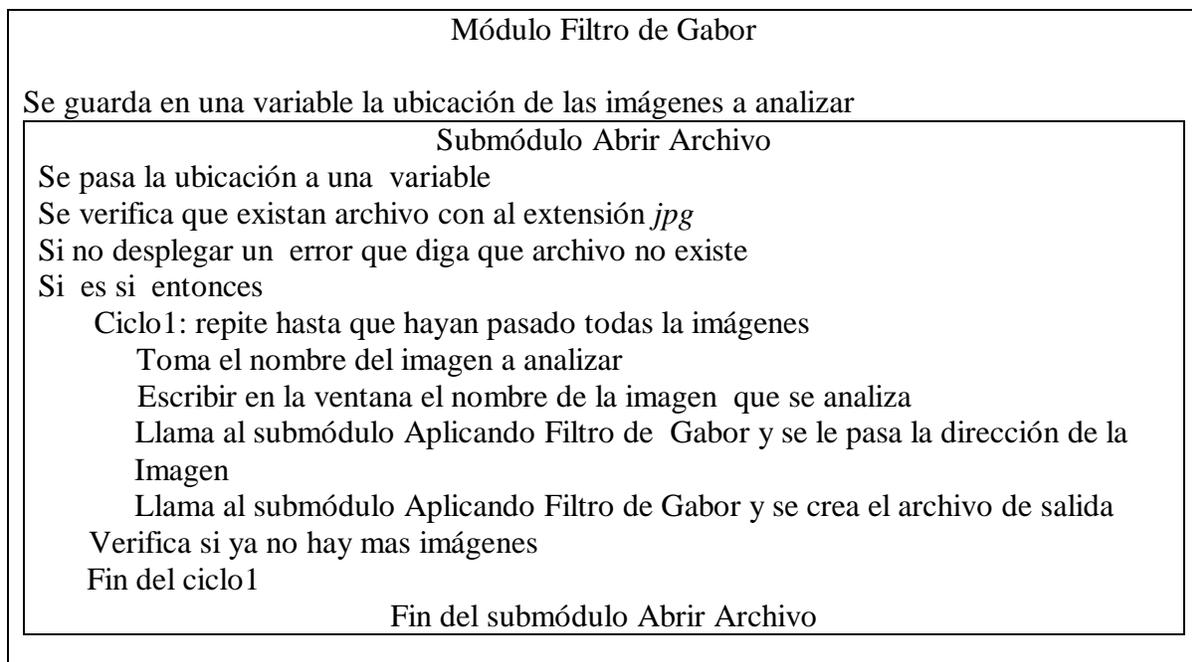
0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.
0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.
0.0000,0.0000,665.4199,665.3806,665.4079,665.3455,665.40
2.9973,663.0436,662.9907,663.0001,663.0060,662.5176,662.
8.0731,657.9810,657.9871,658.0300,658.0109,657.9854,657.
6.4816,656.4293,655.6559,655.4984,655.6477,654.6899,654.
3.0157,653.0135,653.0135,653.0135,653.0135,653.0135,653.
3.0135,653.0135,653.0332,652.9875,652.9737,652.9811,652.
1.4963,651.5620,651.0320,650.9970,650.8030,650.4265,650.
8.4588,648.4121,648.1246,648.2103,648.1746,648.0549,647.
7.0073,647.0073,647.0073,647.0073,647.0073,647.0073,647.

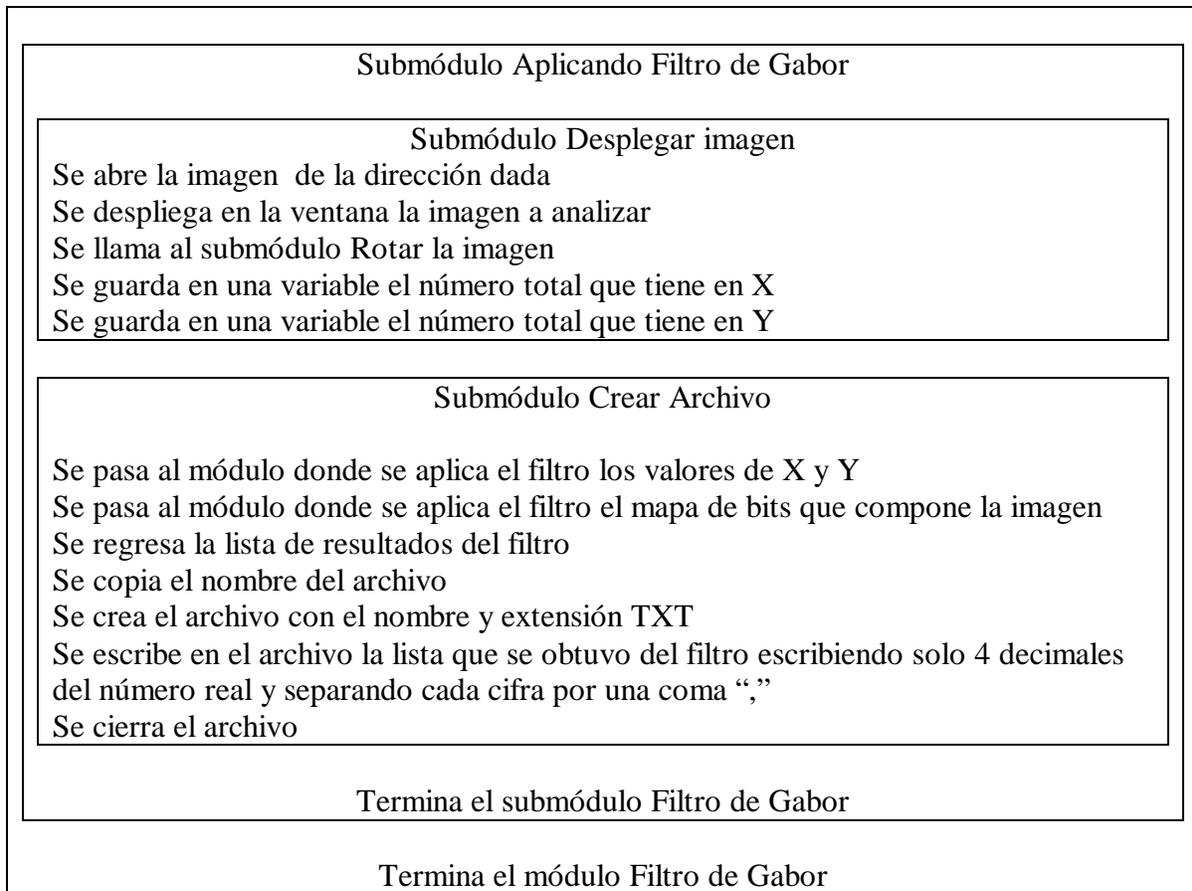
```

Figura 4.6 Fragmento de una archivo de una imagen analizada

4.5 Pseudocódigo del módulo Filtro de Gabor

Dado que el módulo es muy extenso en la parte de la aplicación del filtro solo se escribe un fragmento del módulo [7] [11][12]. Recordar que primero se llama al modulo UBICACIÓN y luego al módulo Filtro de Gabor.





Una vez Filtrada la imagen se puede ver como en la figura 4.7

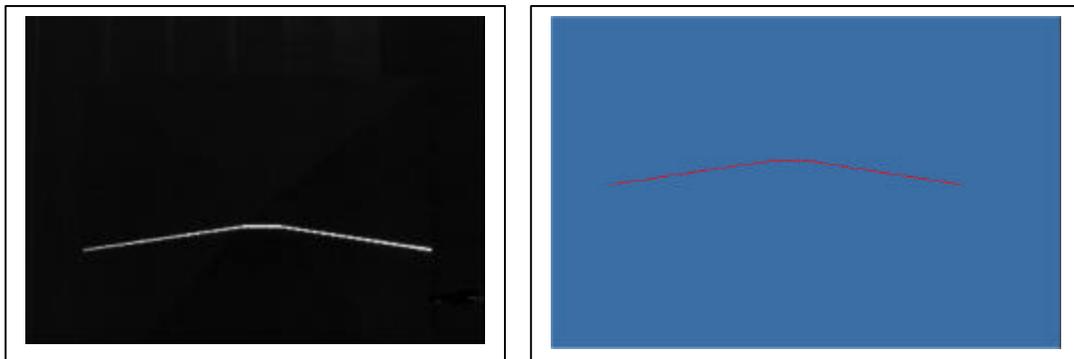


Figura 4.7 Izq. Imagen capturada a la Der.. Imagen analizada

Una vez que se haya aplicado este módulo se puede proceder ya sea con el módulo CALIBRAR o con el módulo VISUALIZAR.

Capítulo 5 : Módulo CALIBRAR

Para la calibración se necesita un objeto del cual se pueda conocer sus medidas, fotografiarlo primero, luego aplicar el Filtro de Gabor y entonces se procederá a calibrar el sistema.

Una vez que las imágenes se pasaron por el Filtro de Gabor y se adelgazó la línea del la luz láser proyectada sobre el objeto en la imagen, ahora se buscará encontrar una relación entre esos puntos y su posición en le objeto real.

Este modulo esta compuesto por dos submódulos, ver figura 5.1, uno de software para el cual se desarrolló lo necesario para hacer una relación entre el objeto real y la imagen desplegada crear un archivo donde se guardarán los datos que servirán más adelante al módulo VIZUALIAR, el otro submódulo es de hardware, este módulo se refiere a la parte física, es decir el objeto en el espacio y la forma de tomar los ejes X, Y, Z.

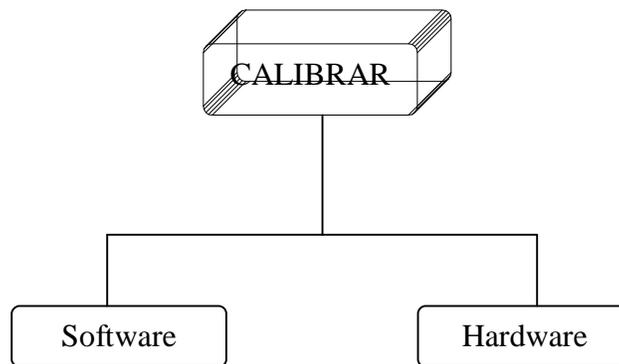


Figura 5.1 Módulo Calibrar y sus submódulos

Una vez seleccionada la opción calibrar, figura 1.2, se abrirá una ventana nueva en la cual se trabajará para calibrar el sistema, ver figura 5.2, esta tiene una barra de menú que ayudara a hacer la calibración del sistema.

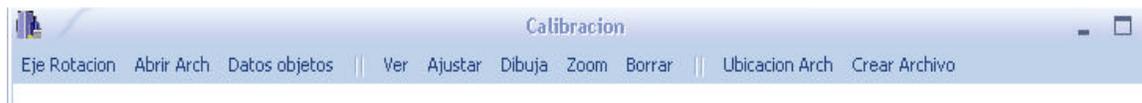


Figura 5.2 Menú de la ventana Calibrar

5.1 Submódulo Hardware

Las coordenadas espaciales X, Y, Z referidas al objeto se calculan en centímetros por el Digitalizador utilizando movimientos verticales del láser de 0.5 cm. o de 1.0 cm. Las coordenadas X, Z son creadas por la proyección de la línea de luz láser sobre el objeto, la coordenada Y viene dada por la posición que tiene el láser-cámara, ya que esta se mueve en un eje vertical paralelo al eje Y del objeto, ver figura 5.3.

La luz láser proyectada puede considerarse como un plano, que al reflejarse por el objeto forma una línea sobre su superficie, misma que es captada por la cámara. Como se puede ver en la figura 3.10, el movimiento del láser-cámara que es quien hace un barrido sobre la superficie del objeto controlado por un motor a pasos, es en el eje Y, dando así un dato conocido en esa coordenada, debido a que se conoce la distancia que se mueve en ese eje en múltiplos de 0.5 cm. o de 1.0 cm.

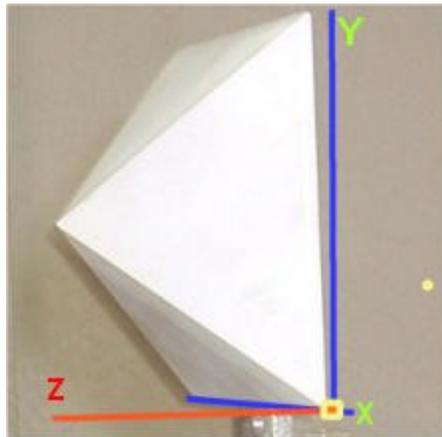


Figura 5.3 Ejes coordenados en el espacio del objeto

La luz láser proyecta una línea y crea un conjunto de coordenadas, estas son parte del objeto que está montado sobre una base giratoria, con lo cual se pueden obtener los tres ejes coordenados X, Y, Z, ver figura 5.4.



Figura 5.4 Luz láser proyectando una línea sobre un objeto

Cuando la imagen sea capturada y se le haya aplicado el Filtro de Gabor, el archivo creado ayudará a determinar las dimensiones de los ejes X y Z, por lo tanto la relación entre la imagen capturada y el objeto.

Para lograr una calibración precisa y poder unir las rebanadas del objeto tomado, es importante determinar un punto de rotación el cual será el origen coordenado y se hace que los objetos roten alrededor de este origen. En la figura 5.5 se muestra la base donde se colocan los objetos. De una manera rudimentaria se saca el punto de rotación, esta manera consiste en colocar una aguja sobre la base y hacerla girar, con un puntero que indica la proximidad del centro, la aguja tiene que girar de manera que se aleja lo menos posible de este puntero, una vez logrado esto se proyecta la luz láser sobre la aguja y se captura la imagen que este proyecta, se le aplica el Filtro de Gabor y, dado que la aguja es muy delgada, da un solo punto en el archivo creado por el filtro, este punto se toma como el origen de las coordenadas.



Figura 5.5 Base donde se busca el origen de las coordenadas

Para poder calibrar y sacar una relación de medidas de cada píxel, se tomará un objeto del cual se pueda conocer sus medidas, para este sistema se utilizó una pirámide y se tomaron sus medidas, como se ven en las siguientes figura 5.6a ,5.6b,5.6c, las cuales fueron:

- ? Base:8.975 cm,
- ? Base al centro de la pirámide: 4.29 cm
- ? Altura (tomada desde la base al pico de la pirámide): 6.425 cm



Figura 5.6a Medida de la base



Figura 5.6b Medida base al centro



Figura 5.5c Medida de la altura

Un vez que se encontró la coordenada de rotación para el objeto y su origen, y conociendo las medidas de un objeto se puede aplicar el software que hará la calibración del Digitalizador

5.2 Submódulo Software

Este submódulo es el encargado de hacer los cálculos necesarios para realizar la relación del objeto real con el desplegado en pantalla, para lograrlo se dividió en otros submódulos, ver figura 5.6.

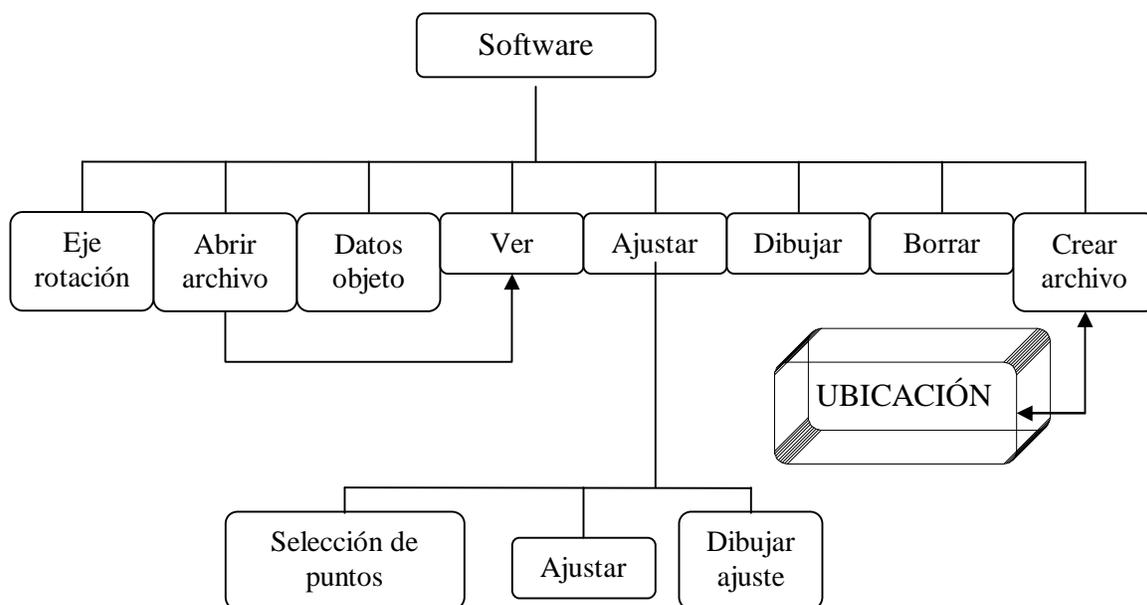


Figura 5.6 Submódulos que componen al Submódulo Software

5.2.1 Submódulo Eje rotación

Este submódulo se encarga de abrir el archivo donde se encuentra los datos del eje de rotación el cual servirá para el origen de los ejes coordenados, para probar el sistema se capturo la imagen de la aguja, la imagen es de 1600 x 1200 píxeles una vez que pasada por el Filtro de Gabor dio las siguientes coordenadas; en $X = 952$, $Y = 669.7452$, estas coordenadas se tomaran como el origen, así que de la toda X menor que 952 se considera negativo, mayor que 952 es positivo. Las coordenadas menores fue 669.7452 en Y se consideran positivas, las mayores son negativas de Y , en ese punto Z es igual a 0. Estos datos ayudaran a hacer la relación de media entre el objeto y los píxeles de la imagen desplegado en pantalla, estos datos se utilizaran de aquí en adelante hasta la terminación de este trabajo y, para ello se explicarán los siguientes módulos.

5.2.2 Submódulo Abrir archivo

Este submódulo abre los archivos que fueron creados por el Filtro de Gabor, o sea los archivos de extensión TXT. Para la calibración en este trabajo el objeto que ayudara a hacer esto es una pirámide, así que no es necesario abrir todas las imágenes hasta la mitad de la pirámide es suficiente para sacar la relación de medida ya que la pirámide es un objeto de medida constante y no es necesario ajustar todas las imágenes ya que se obtendrían datos repetitivos.

Se crea una lista en la cual se guardan el nombre del archivo y un número (num) de archivo que es de cuantos van leídos, donde se guarda los datos se llama nodo de la lista y en cada nodo de esa lista se crea una lista con las coordenadas leídas del archivo, ver figura 5.7, solo se guardan las coordenadas diferentes de 0, así se evita guardar en memoria más datos que no se utilizan

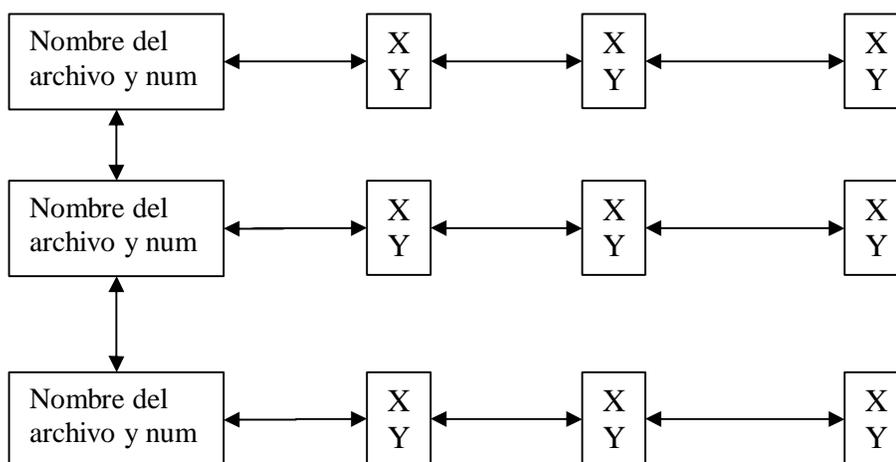


Figura 5.7 Una lista de listas

Hay que recordar que el archivo escrito por el módulo Filtro de Gabor, esta con los ejes girados por 90°, así que al leer el archivo se rotan las coordenadas, ya que tenemos el valor de X en el archivo y a cada valor le corresponde un valor de Y así que al rotar los ejes los valores de X quedan en Y y los valores de Y pasan a X y de esta manera se guardan en la lista los valores correspondiente de la imagen.

5.2.3 Submódulo Datos objeto

Al activarse ese submódulo abre una ventana nueva, ver figura 5.8, en el cual se pide que el usuario de los datos del objeto en centímetros, con el cual se realizará la calibración, en este caso se introducirán las medidas de la pirámide ya vistas en el punto 5.1

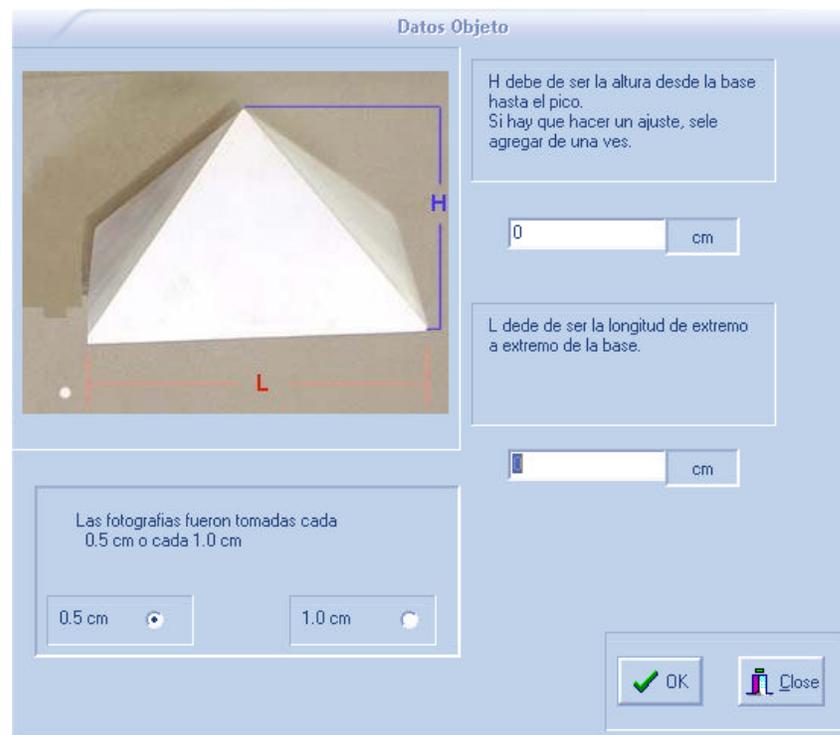


Figura 5.8 Ventana donde se dan los datos del objeto

Los datos serán guardados en unas variables las cuales se usarán más adelante para sacar la relación de medida, los datos que pide el sistema son la altura del objeto, a esta altura si hay que hacerle algún ajuste hay que agregarlo, es decir si el objeto utiliza algo más para estar montado en la base y aleje este del centro de rotación; la longitud del objeto y pregunta si las fotos fueron tomadas cada 0.5 cm o 1.0 cm, todo esto son datos esenciales para el sistema.

5.2.4 Submódulo Ver

Este submódulo está relacionado con el submódulo Abrir archivo, ya que conforme se vayan abriendo los archivos va creando una lista, en la cual el número que se le agrega al nodo de la lista junto con el nombre de archivo, ver figura 5.7, le corresponde ese mismo número en la lista que se despliega en la opción *Ver*, ver figura 5.9, y que va creciendo conforme aumenta la lista de archivos leídos. Esto sirve para que se pueda seleccionar una imagen individual para hacer el ajuste necesario para la calibración y al seleccionar una imagen esta se marcará con una palomita indicando que esa imagen ya ha sido seleccionada. Y así dar paso a la opción *Ajustar*.

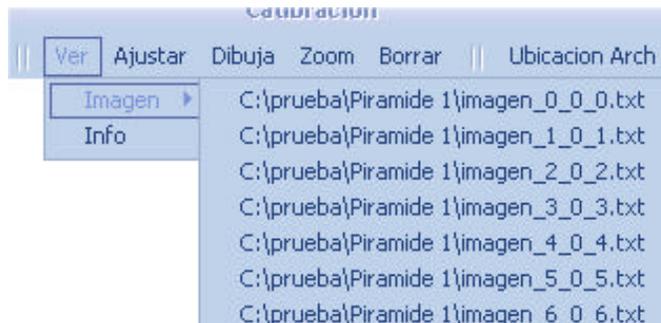


Figura 5.9 Opción *Ver* desplegando la lista de imágenes abiertas

5.2.5 Submódulo Dibujar

Este submódulo se encarga de desplegar los píxeles en la pantalla, de la opción *ver* se selecciona el archivo que se quiere visualizar y luego se selecciona *dibujar* en cual hay dos opciones: dibujar todas o solo una imagen, ver figura 5.10.

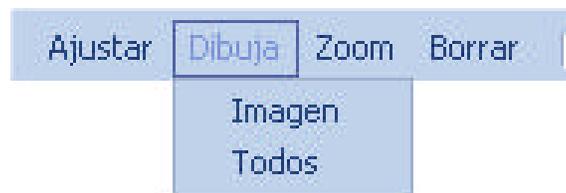


Figura 5.10 Menú de la opción *Dibujar*

Al seleccionar *Dibujar* luego *Imagen*, desplegará en pantalla la imagen seleccionada de la lista de la opción *Ver*, ya que el número que le corresponde en la lista de la opción *Ver* corresponde también en la lista creada donde está el nombre del archivo y su lista de coordenadas, lo que hace es buscar en la lista el número que le corresponde, una vez encontrada se pasa a la lista donde están las coordenadas, se lee la lista y se va graficando

en la pantalla punto por punto hasta terminar la lista, una vez esto en la pantalla se vera la imagen que corresponde al archivo, ver figura 5.11.



Figura 5.11 Despliegue en pantalla de las coordenadas de un archivo seleccionado

Si selecciona Todos, este desplegara en pantalla todos los archivos abiertos hasta ese momento, ver figura 5.12.

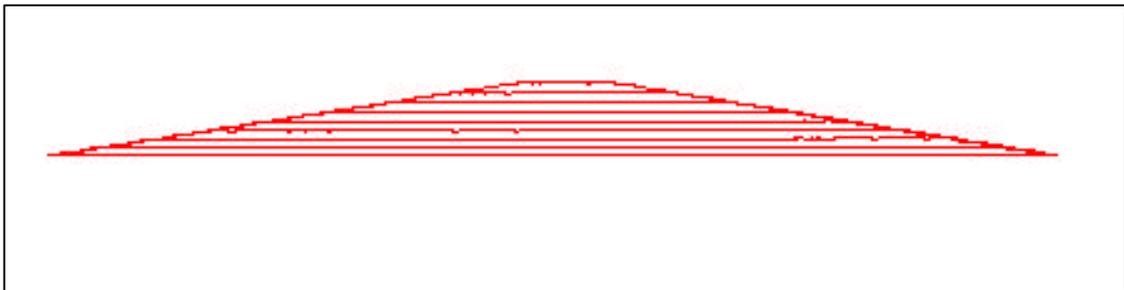


Figura 5.12 Despliegue de todos los archivos abiertos hasta ese momento

5.2.6 Submódulo Borra

Al seleccionar la opción Borra de la barra de menú, figura 5.2, lo que hace es limpiar la pantalla, borrando todos los puntos graficados y esta listo para ver otra imagen, esto se hace para cada imagen que se tiene que analizar, al terminar de analizar la imagen se borra la actual pantalla y se selecciona otra de la opción Ver luego se Dibuja, se analiza, se Borra y este ciclo se repite con todos los archivos.

5.2.7 Submódulo Ajustar

Una vez que se abrieron los archivos y se crearon las listas, lo que se va hacer ahora es analizar cada imagen para obtener los datos para hacer una relación de medida. Se requiere saber cuantos píxeles equivale a 1cm.

De la opción Ver, figura 5.9, se va seleccionar la imagen que será analizada, se grafican los datos del archivo y de las opciones que tiene Ajustar, figura 5.13, se selecciona Selección Puntos el cual abrirá una ventana de mensaje desplegando un mensaje que dice que se tienen que seleccionar los puntos de izquierda a derecha.



Figura 5.13 Opciones que tiene Ajustar

De la imagen desplegada en pantalla se define el punto A como el principio de la imagen y B como el punto final, la imagen se analiza de izquierda a derecha.



Figura 5.14 Se define el punto de inicio y final de la imagen

Se obtienen las coordenadas correspondientes al punto A y al punto B, con el cursor del ratón se seleccionan dos puntos sobre la recta \overline{AC} , figura 5.15, luego se seleccionan otros dos puntos sobre la recta \overline{CD} y se termina seleccionando otros dos puntos sobre la recta \overline{BD} , estos puntos seleccionados son guardados en una lista, como puede ser que el cursor del ratón no estuviera exactamente sobre la recta así que se ignorara el valor de la coordenada Y y el valor de la X es la que ayudará a encontrar el punto que se relaciona sobre la imagen ya que para cada X existe solo una Y, una vez que se localiza el primer punto en la lista se coloca un apuntador en ese punto y se continua hasta encontrar el segundo punto y se coloca otro apuntador a ese punto, así se hace con los demás, entonces la imagen esta dividida en tres segmentos de recta y a cada segmento se le aplica el sistema de ajuste de mínimos cuadrados[13], esto debido que cada segmento puede tener puntos no alineados y al aplicar el ajuste se obtienen puntos ya alineados, figura 5.16, al terminar con este ajuste se grafican los puntos ajustados sobre los puntos originales, de estas nuevas

rectas se encuentran sus puntos, el primer segmento de recta \overline{AC} con el segmento de recta \overline{CD} y la intersección de estas es la coordenada el punto C, luego se busca la intersección entre los segmentos \overline{CD} y \overline{BD} dando así las coordenadas del punto D, este proceso se repita para el resto de las imágenes.

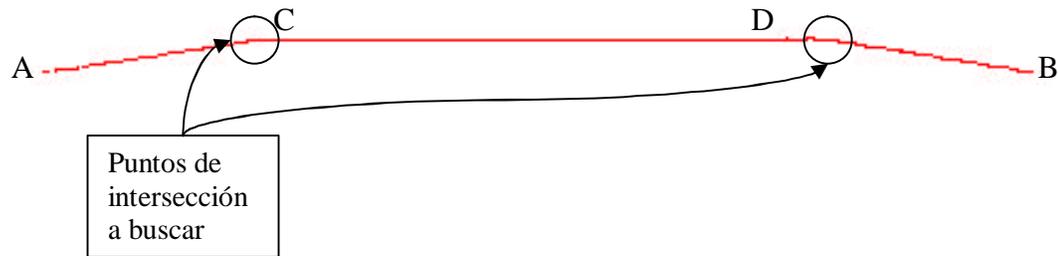


Figura 5.15 En busca de los puntos de intersección

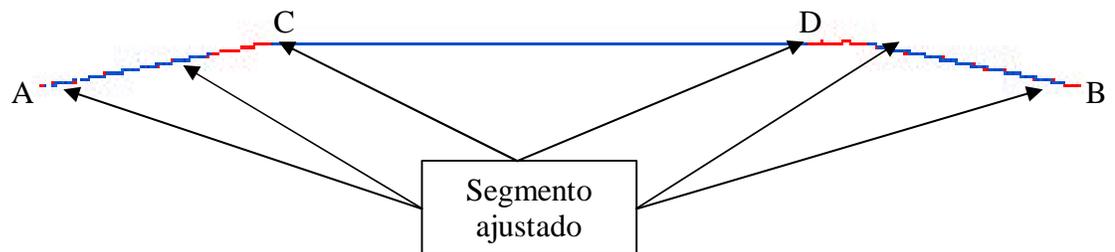


Figura 5.16 Se muestra los segmentos que se ajustaron utilizando el sistema de mínimos cuadrados

Una vez que se realizó ese proceso en todas las imágenes, lo que hace el programa es tomar las coordenadas iniciales y finales de cada imagen y saca la diferencia, eso para luego sacar un promedio del número de píxeles que tienen del punto A hacia el punto B, que servirá para sacar la relación de medida de la base del objeto, en te caso la base de la pirámide mide 8.975 cm. y la el número de píxeles que hay del punto A al B es de 1204 píxeles esto quiere decir que cada 1204 píxeles equivalen a 8.975 cm., una vez explicado esto se puede hacer lo siguiente.

De la figura 5.16 tenemos los siguientes datos de los siguientes puntos:

A(190,853.6497)	B(1396,854.6066)
C(316.0385,834.0887)	D(1273.721,835.4599)

Ahora se busca la distancia entre los puntos A y B, también entre los puntos C y D

$$d_1 = B_x \cdot A_x$$

$$d_2 = D_x \cdot C_x$$

donde

$$d_1 = 1204 \text{ píxeles}$$

$$d_2 = 957.6836 \text{ píxeles}$$

La distancia entre los puntos C y D se conocerá como l , ahora se procederá a buscar al valor de l en cm. , ya que se conocen las medidas del objeto y como es una pirámide se toma una de sus caras, ver figura 5.17 y, como es un triángulo, se facilita el trabajo ya que se hace una relación de triángulos, el ángulo α es igual al ángulo β . B es la altura de todo el triángulo, b es la altura del segundo triángulo formado por el segmento de recta \overline{CD} hacia el pico del triángulo, así que se obtiene lo siguiente?

$$B = 4.4875 \text{ cm}$$

$$A = 8.975 \text{ cm}$$

$$\frac{A}{2} = 4.4875 \text{ cm}$$

$$\text{Posicion } Y = 1 \text{ cm}$$

$$b = B - (\text{Posicion } Y) = 3.4875 \text{ cm}$$

$$\text{tg} \alpha = \text{tg} \beta$$

$$\frac{B}{A} = \frac{b}{a}$$

$$a = b \cdot \frac{A}{B}$$

$$a = 3.4875 \cdot \frac{4.4875}{4.4875}$$

$$2a = 6.975 \text{ cm} = l$$

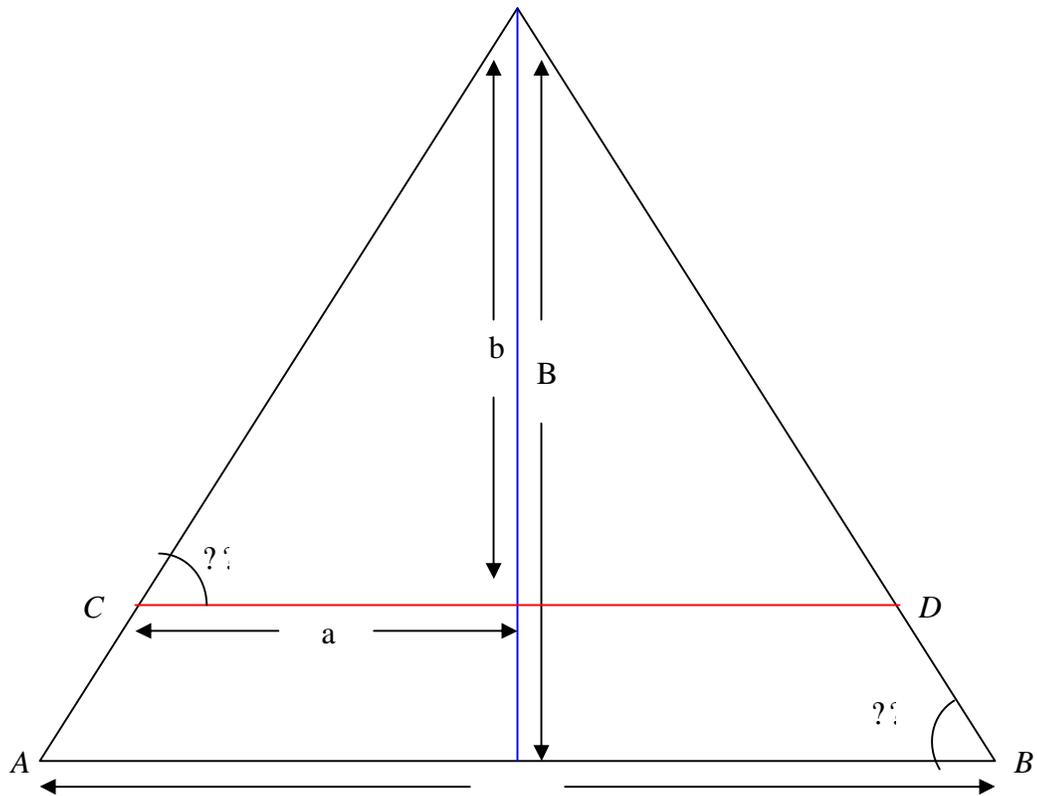


Figura 5.17 Triangulo que representa la cara de la pirámide

La posición Y es la altura en que se encuentra el láser-cámara en este caso es de 1cm de la base del triangulo es por eso que se tiene 3.4875, en la figura 5.17 es el segmento de recta \overline{CD} , el valor de a es la mitad del triangulo así que para obtener el triangulo completo se multiplica por 2 quedando así $2a$, que da el valor de l en cm de 6.975 cm.

Ahora se busca la altura $?h$ del punto sobre el objeto, ver figura 5.18, ese punto específica el volumen sobre el objeto, obteniendo así una relación de píxeles-volumen, para eso se buscaron las coordenadas de los puntos C y D para ayudar encontrar el volumen, estos pasos se realizan con todas la imágenes para calibrar. Obteniendo así una tabla de relación entre píxel y medidas reales.

A ? 8.975

H ? 6.42 la altura de la base al pico

a ? 1cm es la posición de Y

»h ? ?

tg? ? tg?

$$\frac{?h}{a} ? \frac{H}{A/2}$$

$$?h ? a \frac{H}{A/2}$$

$$?h ? 1 \frac{6.42}{4.4875}$$

$$?h ? 1.43064$$

$$?h ? 1.43064$$

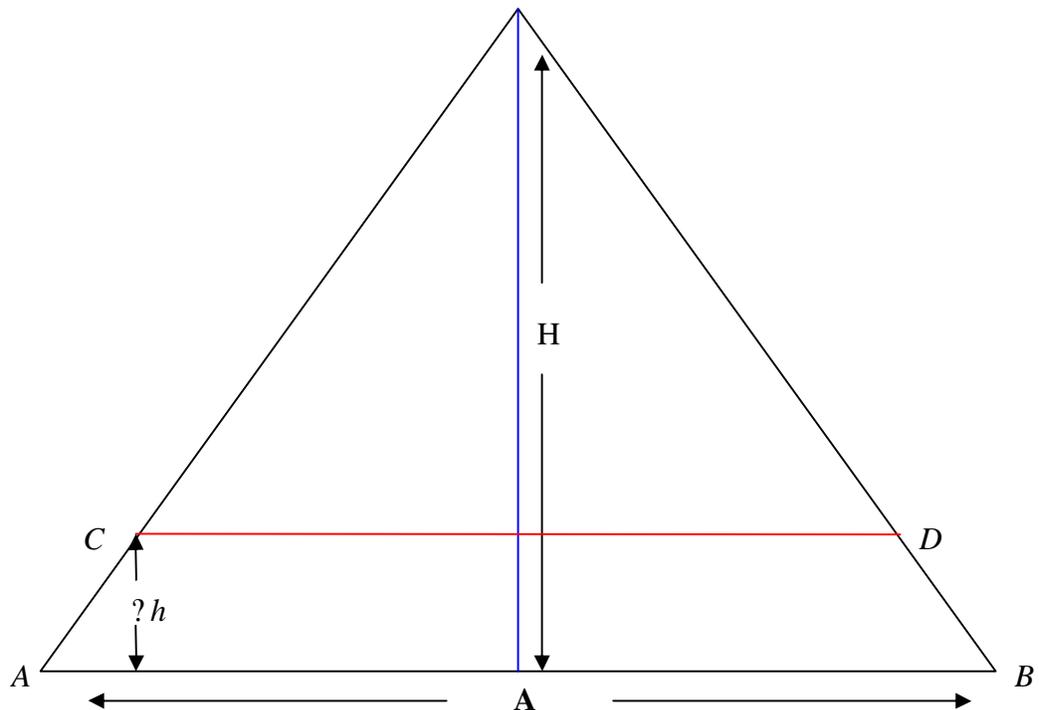


Figura 5.18 Buscando la ?h para encontrar el volumen

Estos pasos se repiten con todas las imágenes utilizadas para calibrar. Obteniendo así una tabla de relación entre píxel y medidas reales.

La siguiente tabla es la relación de longitud, esto es sobre el eje X, se tiene una relación del objeto real y la imagen, cada 120.81 píxeles son 0.975 cm

Altura en Y en cm	Medida en centímetros con el objeto real	El número de píxeles que tiene la medida del punto C al D
0	8.975	1204
0.5	7.975	1084.275
1	6.975	957.6836
1.5	5.975	828.1976
2	4.975	694.8358
2.5	3.975	562.7112
3	2.975	416.0953
3.5	1.975	269.9115
4	0.975	120.8145

La siguiente tabla es la relación que se tiene en el eje Z, que es el volumen del objeto.

Altura en Y en cm	Posición en eje Z sobre el objeto real	La altura que tiene el punto C con respecto al punto A
0	0	0
0.5	0.71532	10.54392
1	1.430641	21.08784
1.5	2.145961	31.63176
2	2.861281	42.17568
2.5	3.576602	52.7196
3	4.291922	63.26352
3.5	5.007242	73.80744
4	5.722563	84.35136

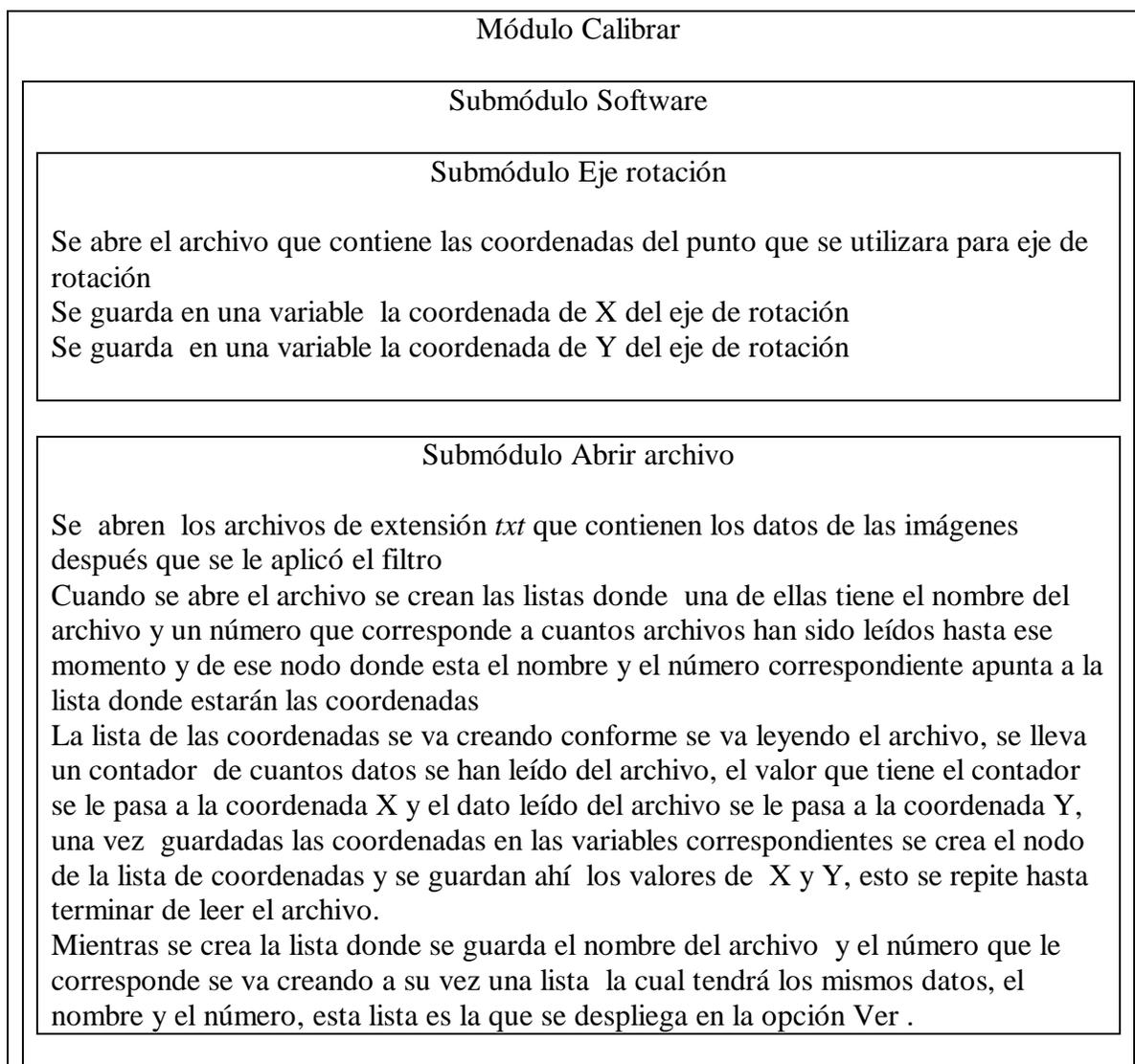
Con estos datos de cada tabla se hace la regresión lineal, para obtener los datos a y b de la fórmula $X = a + (b * x)$, que al seleccionar cualquier punto de la imagen se puede localizar un punto sobre la superficie del objeto exacta sobre el objeto [14].

5.2.8 Submódulo Crear Archivo

En el archivo que se crea se guardan los datos del centro de rotación que sirve de origen, se guardan los valores que sirven para hacer la regresión lineal para el eje X y la del eje Y ya que estos ayudarán para encontrar cualquier punto en la superficie de un objeto, el archivo es guardado en "C:\\Dig_Calibracion\\calibracion.txt". Con esta parte se termina de calibrar el sistema [7][15].

5.3 Pseudocódigo del módulo CALIBRAR

Dado que el módulo es muy extenso en la parte de la aplicación de la calibración solo se escribe un fragmento del módulo [14][15][7.]



Submódulo Datos objeto

Se guarda en una variable al dato de altura de objeto
Se guarda en una variable al dato de longitud que tiene el objeto
Se guarda en una variable el valor del desplazamiento que hizo el láser-cámara en la toma de las imágenes

Submódulo Ver

Despliega los archivos leídos hasta el momento
Se busca en la lista el archivo seleccionado el cual se va analizar
Se guarda en una variable el número de la imagen seleccionada

Submódulo Dibujar

Grafica la lista de las coordenadas que corresponden al archivo que se selecciona
O Se grafica toda la lista de archivos con sus correspondientes coordenadas

Submódulo Borrar

Borra todos los puntos desplegados en pantalla

Submódulo Ajustar

Submódulo Selección de puntos

Checa que ya se haya seleccionado una imagen de la opción Ver
Si no desplegar mensaje de error que diga que no hay imagen seleccionada
Checa si la imagen no ha sido ya ajustada para no repetir el mismo proceso
Si es así se despliega un mensaje de error diciendo que la imagen ya ha sido seleccionada anteriormente
Despliega en pantalla el mensaje que diga seleccionar los puntos de izquierda a derecha

Se crea una lista con los puntos seleccionados por el usuario por medio del ratón

Submódulo Ajustar

La de la variable que tiene el número de la imagen seleccionada se busca el igual en las lista que contiene el los nombres del los archivos y sus coordenadas correspondientes
Al encontrar el número que corresponde al archivo se pasa a la lista correspondiente de las coordenadas X y Y
De la lista que se creó de los puntos seleccionados, se toma el primer valor de X y se busca el igual en la lista de coordenadas que tiene el archivo

Al encontrar el valor igual de X se pone un apuntador a ese nodo
Se avanza en la lista de puntos obtenidos por el ratón y se realiza el mismo proceso anterior hasta terminar la lista de puntos seleccionado por el ratón
Se hace el ajuste de mínimos cuadrados
Se crea una nueva lista con los nuevos puntos del ajuste
Se hace la regresión lineal para el eje X y los valores obtenidos para localizar cualquier punto sobre ese eje son guardados en unas variables.
Se hace la regresión lineal para el eje Y y los valores obtenidos para localizar cualquier punto sobre ese eje son guardados en unas variables

Submódulo Crear Archivo

Se crea el archivo en la con la dirección C:\\Dig_Calibracion\\calibracion.txt
Se escriben las variables que contienen las coordenadas del centro de rotación
Se escriben las variables que tienen los valores de la regresión lineal del eje X
Se escriben las variables que tienen los valores de la regresión lineal del eje Y

Fin del submódulo Software

Fin del Módulo Calibrar

Capítulo 6 Módulo VISUALIZAR

Con este modulo se logra visualizar el objeto, para llegar a eso se crean dos Submódulos uno llamado Crear nuevo y el otro Visualizar, utiliza otro módulo ya visto el de Ubicación, figura 6.1. Al seleccionar la opción visualizar, figura 1.2, se abre una ventana, figura 6.2, nueva en el cual se tienen dos opciones, crear nuevo ó visualizar.

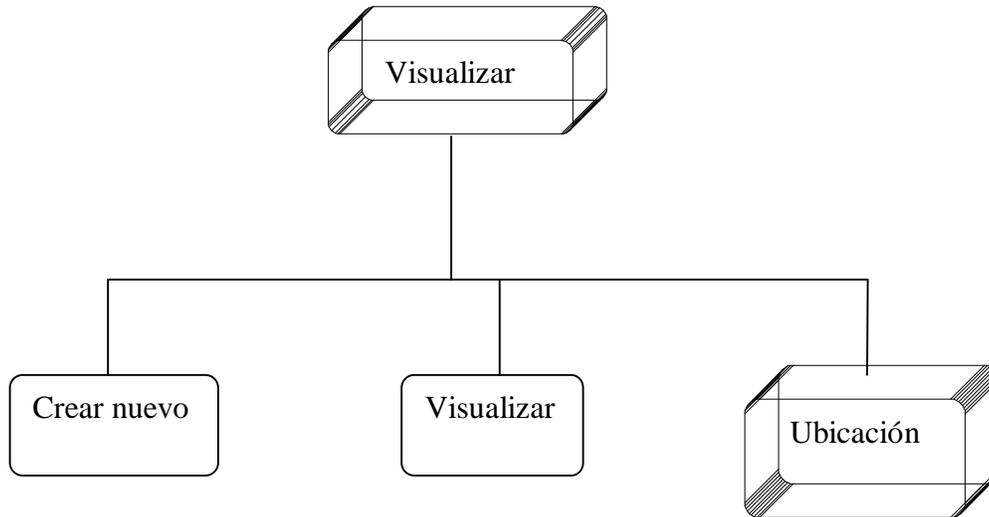


Figura 6.1 Diagrama de módulos que componen al módulo Visualizar



Figura 6.2 Ventana que se abre al seleccionar la opción visualizar

Para lograr la visualización se utilizó el OpenGL, que es una librería no un lenguaje de programación. Un lenguaje es generalmente C o C++ en el que se escribe el programa que realiza llamadas a funciones de las librerías, utilizando para ello la sintaxis de C.

6.1 Open GL

OpenGL es independiente del sistema de ventanas utilizado y del sistema operativo, es decir, no incorpora rutinas para el manejo de ventanas. Este manejo debe realizarse a través del API del entorno de ventanas elegido en este caso es el lenguaje utilizado para el desarrollo del sistema, Builder C.

OpenGL incorpora la funcionalidad básica para realizar la visualización (*rendering*), existe, no obstante, un conjunto de aspectos que no son estrictamente de visualización y que se encuentran en una librería estándar adicional llamada GLU (*OpenGL Utility Library*).

Existen algunas herramientas o *toolkits* que facilitan la labor de “pegar” OpenGL con un entorno de ventanas. Un ejemplo es GLUT (*OpenGL Utility Toolkit*), un completo conjunto de herramientas que facilitan el trabajo con OpenGL y MS-Windows o X Windows.

Para trabajar con OpenGL necesitamos al menos las siguientes componentes:

- ? Un lenguaje de programación con su compilador, que pueda realizar llamadas a funciones en formato C: C, C++, VisualBasic, Java...
- ? Librería OpenGL, contiene las funciones propias de OpenGL, dedicadas a las tareas de *rendering*. Las funciones incorporan el prefijo gl-.
- ? Librería de utilidades GLU, contiene algunas funciones, con el prefijo glu- que realizan las siguientes tareas:
 - o Transformación de coordenadas.
 - o Poligonalización de objetos.
 - o Manipulación de imágenes para aplicarlas como texturas.
 - o *Rendering* de figuras canónicas: esferas, cilindros y discos.
 - o Curvas y superficies NURBS.
 - o Informe de errores.
 - o Librería para “pegar” OpenGL y el sistema de ventanas. Esta librería depende del sistema elegido. Entre los más habituales encontramos a WGL, es la librería para la familia de sistemas MS-Windows. El prefijo utilizado en sus funciones es wgl-.

- GLX: Es la librería para X Windows. Utiliza el prefijo glx- en sus funciones.

Es posible encontrar un amplio abanico de librerías que encapsulan el trabajo con el API del entorno de ventanas y facilitan la creación de ventanas, entre ellas destaca GLUT, para versiones MS.

En el desarrollo del sistema se va a trabajar OpenGL bajo Builder C++ [16], haciendo uso de las librerías OpenGL y GLU.

6.2 Submódulo Crear nuevo

Al seleccionar la opción crear nuevo, figura 6.2, se abre una ventana nueva, figura 6.3, en la cual hay que dar el dato de cómo fueron capturadas las imágenes, si cada 0.5 cm. o cada 1.0 cm., dado que puede haber objetos que al fotografiarse creen un conjunto grande de imágenes se pueden unir todas las imágenes de una rebanada en un solo archivo con la opción crear una lista de imágenes.



Figura 6.3 Ventana de la opción Crear nuevo

Los datos obtenidos por el Filtro de Gabor, utilizando el archivo creado por el módulo CALIBRAR se utilizan para crear nuevos archivos, convirtiendo esos datos que son coordenadas en la pantalla en puntos sobre la superficie del objeto, los cuales serán guardados en un nuevo archivo, con extensión *dat* con el mismo nombre de cada archivo cuando se convierte individualmente para cada imagen, con extensión *rot* y el nombre de la primera imagen cuando se crea un solo archivo con todas las rebanadas que tienen que ver con el objeto. Estos archivos serán utilizados para desplegar en la pantalla el objeto

Los datos contenidos en el archivo calibra, que son los valores de las coordenada del centro de rotación que es el origen, los valores de para localizar un punto en el eje X y los del eje Z ya que lo valores de Y viene dado en el nombre del archivo. Para poder seleccionar la opción crear nuevo se tuvo que dar la ubicación de los archivos de las imágenes ya pasadas por el filtro, esto es para abrir todas la imágenes contenidas creando una lista de estas, figura 5.7, una vez creada la lista lo que se hace es que para cada imagen, sus coordenadas se trasforman en puntos de la superficie del objeto, esto se logra mediante la aplicación la fórmula que se obtuvo de la regresión, $X = a + (b * x)$, donde X es la coordenada que se esta buscando, a y b son los valores que se obtuvieron en la regresión lineal del eje y x es la coordenada de la imagen, aplicando esta formula a cada eje, se obtiene los ejes X y Z ya que Y es obtenido del nombre de la imagen ya que este esta guardado ahí. Conforme se van obteniendo los puntos en la superficie del objeto se va creando una lista con los puntos que ahora contendran tres coordenadas X, Y, y Z, los cuales se guardaran en un archivo con el mismo nombre que le corresponde pero con extensión *dat*, la forma de escribir el archivo es escribiendo cada punto X, Y, Z en el mismo renglón, luego hace salto de renglón y escribe el otro punto, ver figura 6.4 , así hasta terminar la lista de las coordenadas de la superficie del objeto .

```

1.211,0.500,-1.234
1.199,0.500,-1.235
1.198,0.500,-1.225
1.144,0.500,-1.268
1.139,0.500,-1.263
1.125,0.500,-1.266
1.116,0.500,-1.264
1.109,0.500,-1.260
1.105,0.500,-1.254
1.099,0.500,-1.249

```

Figura 6.4 Fragmento de un archivo de salida de los puntos en la superficie

6.2.1 Crear una sola lista de imágenes

Para crear una sola lista es realiza el mismo procedimiento explicado con anterioridad pero antes de escribir el archivo se hace un proceso más.

El archivo de extensión *.rot* es la rebanadas del objeto, si se crearan como archivos de extensión *.dat* serian muchos y contendrían puntos que se repiten en cada imagen, es necesario eliminar los puntos repetidos y guardar todos esos puntos de una rebanada en un solo archivo.

Las imágenes de una rebanada son tomadas a una misma altura pero en diferente ángulo de rotación, así que primero tenemos que rotar las imágenes para que los puntos que se repiten sean eliminados, solo dejando un punto de todos ellos.

Se utiliza la matriz de rotación para rotar las imágenes alrededor del punto del centro de rotación, el origen, obtenido por la calibración.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Una vez ya rotadas se utilizo el método de ordenación *quicksort* [24], con el criterio de todos lo puntos del lado positivo de Z y los del lado negativo de Z, así se divide la rebanada del objeto en dos partes haciendo más fácil la localización de los puntos repetidos en el eje X y Y pero el punto de Z no puede estar repetido, obteniendo así la rebanadas con puntos no repetidos, una vez obtenidas esos puntos se crea el archivo con todos los puntos no repetidos de la rebanada del objeto, con el nombre de la primera imagen de esa rebanada y la extensión *rot*.

6.3 Submódulo Visualizar

Ya creados los archivos de las imágenes con extensión *dat* o *rot* se pueden visualizar en cualquier momento, seleccionando la opción visualizar, figura 6.2, se abrirá una nueva ventana, figura 6.5, en la cual se tiene unos paneles de control por medio de los cuales se puede Girar, Rotar o Escalar el objeto desplegado en pantalla, también se tiene un panel en el cual puede cambiar el punto de de vista ya sea en el eje X, Y, o Z.



Figura 6.5 Ventana de la opción Visualizar

6.3.1 Transformación de puntos 3D a Píxeles

Los objetos a visualizar se encuentran en el espacio cartesiano 3D, este espacio se mide en unidades del mundo real como metros, milímetros, otros, estas son las que llamamos coordenadas del mundo. Para que los objetos definidos en coordenadas del mundo puedan ser visualizados en una pantalla que solo puede definir dos dimensiones, deben realizarse diferentes transformaciones para dar la sensación visual de 3D.

Los objetos del mundo real son vistos desde un punto del espacio, denominado punto de vista, este punto de vista es el punto del espacio en el que está situado el observador. Este punto define un nuevo sistema de coordenadas, denominadas coordenadas del ojo, pues es donde se encuentra el ojo que mira al mundo real. La primera transformación que deben sufrir los puntos del objeto consiste en ponerlos en función del sistema de coordenadas del ojo. Esta transformación se llama transformación de vista (*Viewing Transformation*).

A los objetos situados en el mundo real se le pueden aplicar tres operaciones básicas; traslación, rotación y escalado. Estas operaciones permiten situar el objeto en la posición elegida (traslación), darle la orientación adecuada (rotación) y alterar sus dimensiones (escalado). La aplicación de un conjunto de estas operaciones constituye la segunda transformación; transformación de modelo (*Modeling Transformation*). Las transformaciones de vista y de modelo son duales, es decir, aplicar las operaciones elementales vistas es equivalente a alterar la posición del punto de vista y viceversa, por esta razón utilizamos una única transformación para representar este proceso dual que es la transformación de modelo y vista (*Modelview Transformation*).

La pantalla de la PC es un plano finito, por lo que no puede representar todo el espacio del mundo real, se define qué porción del espacio se va a visualizar, esta región se denomina volumen de recorte y está limitado tanto en alto y en ancho como en profundidad. Además, los objetos que hemos de visualizar se encuentran definidos en las coordenadas tridimensionales (x,y,z), para poder visualizarlos en una pantalla (espacio en dos dimensiones) es necesario transformar estos puntos tridimensionales en puntos de un espacio 2D (x,y), utilizando algún tipo de proyección (ortográfica o perspectiva). El proceso de recorte y el paso de los puntos en coordenadas del ojo a puntos en coordenadas del plano (*Clip Coordinates*) configuran la segunda de las transformaciones necesarias, denominada transformación de proyección (*Projection Transformation*). La utilización de un volumen de recorte permite eliminar de antemano los objetos que se encuentran fuera del mismo y acelerar, de esta manera, el proceso de visualización, ver figura 6.7.

6.3.2 Abrir Archivo

Para visualizar el objeto en pantalla se va a la opción Abrir Archivo, y se seleccionan los archivos ya sea de extensión *dat* o *rot*, conforme se van leyendo los datos

de los archivos se va creando una lista con todos los puntos con los cuales el OpenGL graficará en pantalla [17][25], ver figura 6.6.

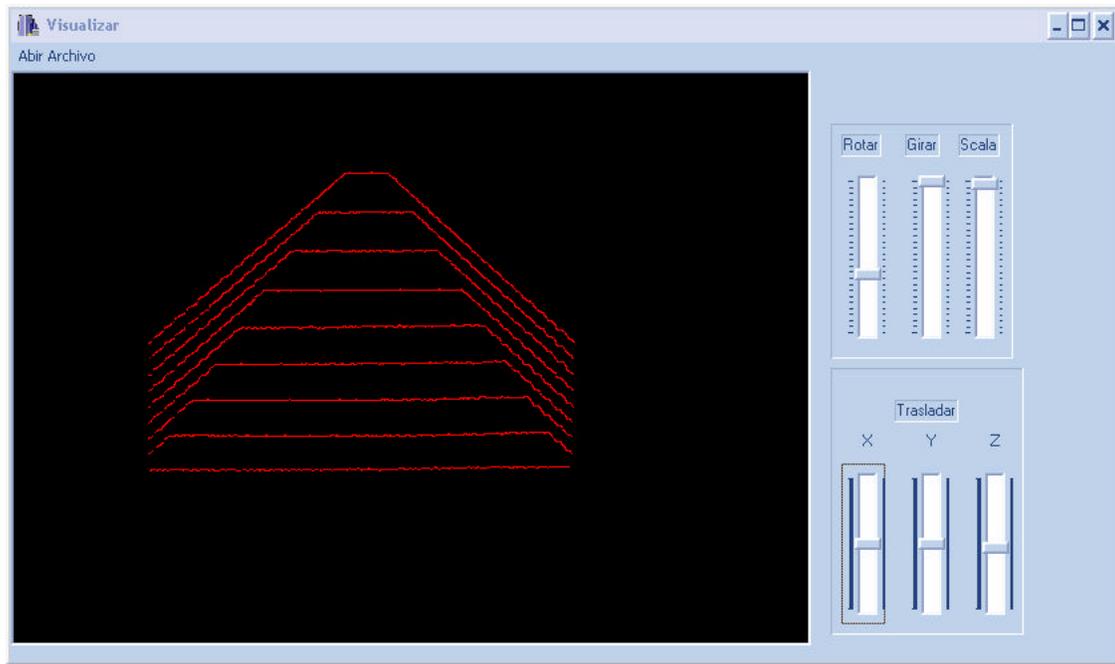


Figura 6.6 Visualización de media pirámide

Un objeto que se digitalizó fue la horma de un zapato, se pasó por todos los módulos anteriores hasta llegar a la visualización de este, en las siguiente figuras se muestra como se visualiza en diferentes puntos de vista y escalas, figura 6.7,6.8, 6.9.

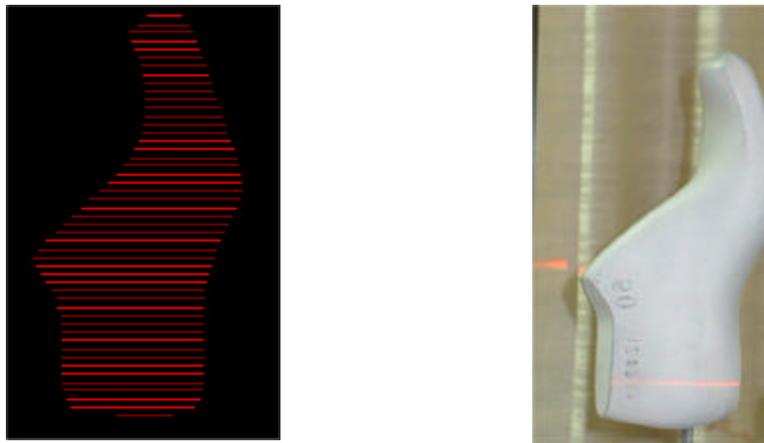


Figura 6.7 Del lado izquierdo esta el objeto visualizado, del lado derecho esta el objeto

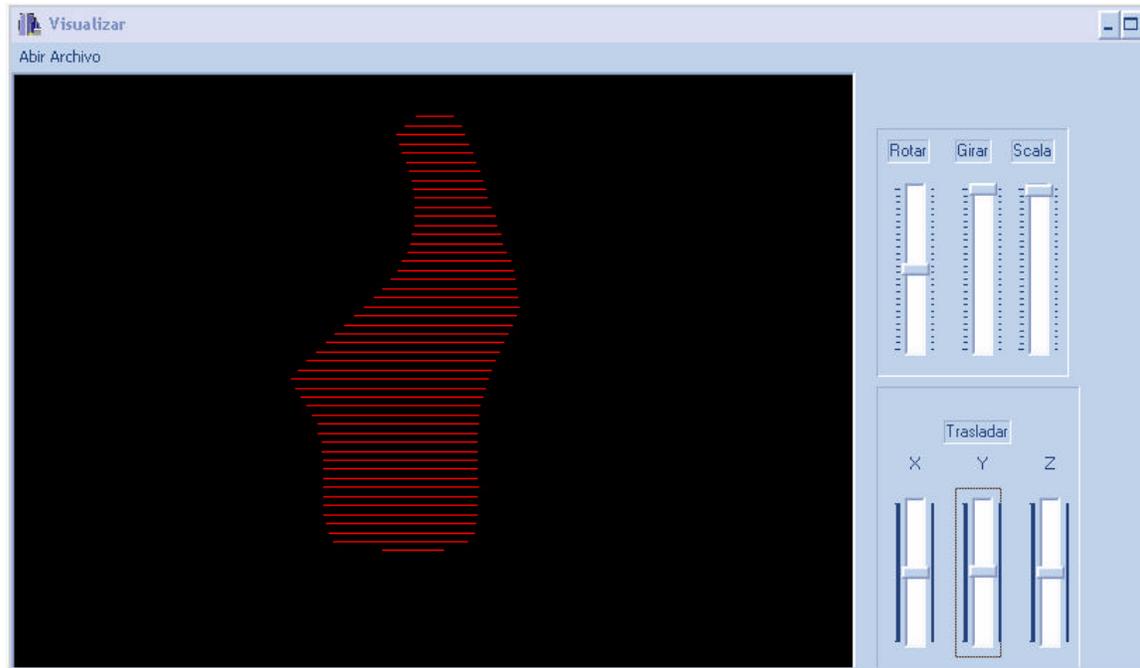


Figura 6.8 Visualización de la horma de un zapato

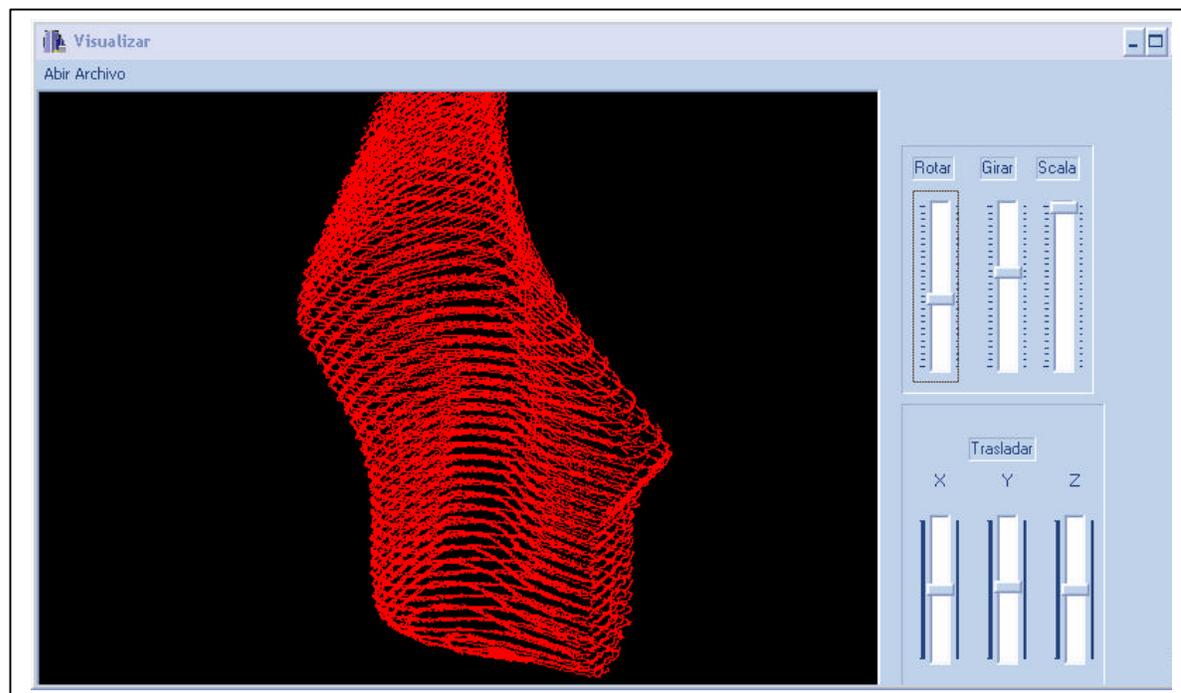


Figura 6.9 Se visualiza la horma del zapato a una escala y con una rotación

6.4 Pseudocódigo del módulo VISUALIZAR

Dado que el módulo es muy extenso en la parte de la aplicación de la visualización solo se escribe un fragmento del módulo [17][25][7].

Submódulo VISUALIZAR

Antes del entrar al submódulo Crear nuevo se llama al módulo UBICACIÓN

Submódulo Crear nuevo

Se checa que en la ubicación dada existan archivos de extensión *txt*
Si no se despliega un mensaje de error diciendo que no existen archivos de tipo *txt*
Si existen, se abre el archivo creado por el módulo CALIBRAR y se guardan en unas variables los datos contenidos
Se llama al submódulo crear lista

Submódulo Crear lista

Se abren los archivos de extensión *txt* que contienen los datos de las imágenes después que se le aplicó el filtro
Cuando se abre el archivo se crean las listas donde una de ellas tiene el nombre del archivo y un número que corresponde a cuantos archivos han sido leídos hasta ese momento y de ese nodo donde está el nombre y el número correspondiente, apunta a la lista donde estarán las coordenadas
La lista de las coordenadas se va creando conforme se va leyendo el archivo, se lleva un contador de cuantos datos se han leído del archivo, el valor que tiene el contador se le pasa a la coordenada X y el dato leído del archivo se le pasa a la coordenada Y.
Se obtienen las coordenadas de la superficie del objeto con las coordenadas leídas del archivo *txt*
Se guardan en la lista de las coordenadas de la imagen los nuevos puntos obtenidos
Este proceso se repite con todos los archivos existentes en la ubicación dada

Submódulo Archivo simple

Se crea el archivo con el nombre de la imagen que está guardada en lista y con extensión *dat*
Se escribe la coordenada de cada punto de la forma X, Y, Z con salto de renglón por cada punto, esto se repite hasta terminar la lista de coordenadas.

Submódulo Crear una sola lista de las imágenes

Se crea una lista con todos los puntos de una rebanada

Se aplica el método de ordenamiento *quicksort*

Se crean dos listas, una tiene todas las coordenadas donde Z es positivo la otra donde Z es negativo

Se buscan puntos repetidos de X y Y y se eliminan solo dejando solo un punto de los repetidos

Se aplica la matriz de rotación en cada punto

Los puntos nuevos se guardan en una lista

Se crea el archivo el cual tiene el nombre de la primera imagen de la rebanada

Se escribe la coordenada de cada punto de la forma X, Y, Z con salto de renglón por cada punto, este proceso se repite hasta terminar la lista de coordenadas.

Este proceso se repite hasta terminar con las lista de las rebanadas

Fin del submódulo Crear archivo

Submódulo Visualizar

Se seleccionan los archivos que se van a visualizar del objeto

Se crea la lista de las coordenadas de cada archivo

Se grafican los puntos de cada lista

Se utilizan métodos de Open GL para rotar la imagen, girarla, escalarla y mover el punto de visión hacia el objeto

Fin del submódulo Visualizar

Fin del módulo VISUALIZAR

Capítulo 7 : Conclusiones y Trabajo a futuro

7.1 Conclusiones

Se obtuvo la visualización del objeto el cual era el objetivo principal de esta tesis, ver figura 6.6,6.7,6.8,6.9,también se obtuvo una forma de hacer una relación de medidas de píxeles a centímetros, en los archivos hechos para la localización de los puntos sobre la superficie del objeto se obtuvo un margen de error de más-menos 0.3mm a 0.5mm, esto en parte se debe que el formato *jpeg* hace una compresión de la imagen degradando la información real de esta, otro factor es la forma de trabajar con los sub-píxeles ya que estos no son exactos.

Hasta este momento se llegó a una parte de la solución del problema de la empresa zapatera.

7.2 Trabajo a futuro

Los trabajos a futuro son:

- ? Minimizar el error en la localización del punto sobre la superficie
- ? Incrementar velocidad para captura de imágenes
- ? Hacer el sistema autosuficiente para que no dependa de ningún otro programa para su trabajo
- ? Mejorar la parte mecánica, implica tener todo dentro de un gabinete
- ? Hacer programas para diferentes aplicaciones como mandar superficie digitalizada para maquinado con control numérico.
- ? Uso posterior en paquetes comerciales de Cad como puede ser en animación, texturación.
- ? Otros tipos de generación de mallas como las Nurbs (Nonuniform Rational B-Splines), Polígonos.

Lista de los módulos del sistema

Aux_Motores.cpp
Calib_datos.cpp
Calib_trabajo.cpp
Callib_trabajo1.cpp
Calibracion1.cpp
Cap_fotos.cpp
Conversión.cpp
Datos_objetos.cpp
Digitalizador.cpp
Eje_rotacion.cpp
FGabor1.cpp
Filtro.cpp
Info_fotografias.cpp
Mallar_y_Visualiza.cpp
Rotacion.cpp
TArchivo.cpp
TConvercion.cpp
TConvercion21.cpp
TFiltro_Gab1.cpp
TFiltroG2.cpp
TFotografia.cpp
TGabor1.cpp
TListas.cpp
TMotores.cpp
TOpera1.cpp
TOpera_rto
Trabajo_Gab1.cpp
Unidades.cpp
Visual_puntos.cpp
Visualizar1.cpp
Visualiza_malla_puntos.cpp

El programa ejecutable es Digitalizador_Objetos.exe

Referencias:

- [1] www.beyondlogic.org/usb/usblptpd11.htm.
- [2] ProtTalk.pdf
- [3] www.beyondlogic.org
- [4] Ver programa llamado DMotor.cpp.
- [5] La documentación para Lead Screen Capture puede encontrarse en la LEADTOOLS Screen Capture VCL, Ltscrvc1.hlp en el directorio de LEAD\Help.
- [6] Características de Cámaras cortesía de www.steves-digicams.com
- [7] Menu_Principal.exe y su código en el Menu_Principal.cpp, agregado en el cd.
- [8] Módulo TFotografia.cpp
- [9] Módulo TMotores.cpp
- [10] capítulo 3 de la tesis Métodos de Calibración de un Digitalizador Óptico de Edgar Román Arce Santana.
- [11] Módulo FGabor1.cpp
- [12] Módulo Trabajo_Gab1.cpp
- [13] Análisis Numérico. Burden, Faires
- [14] Datos de la Tabla de relación en archivos de excel, agregado en el cd de la tesis
- [15] Módulo Calibracion1.cpp
- [16] Archivo llamado OpenGL.pdf., agregado en el cd
- [17] Módulo Mallar_y_Visualiza.cpp
- [18] Builder_vs_Visual.pdf agregado en el cd.
- [19] Módulo Unidades.cpp
- [20] Laser Snf.pdf agregado en el cd.
- [21] cd_setp_motor_cat.pdf
- [22] Driver MD4M.pdf
- [23] Cortesía de José Antonio Muñoz Gómez.
- [24] Estructura de Datos
Cairo / Guardati.
- [25] Módulo Vizualir1.cpp

Otras referencias:

Estructura de Datos
Cairo / Guardati.

Manual de Borland C++
Chips H.Pappas /William H. Murray

Programación con C++ Builder 4
Francisco Charte

Open GL programming Guide
Jackie Neider / Tom Davis / Mason Woo

Prgramacion en Open GL
Richard S. Wright Jr.

Handbook of Grid Generation
Jeo F. Thmpson / Bharat K. Soni / Nigel P. Weatherill

Computer Graphic: Principles and Practices
Foley/ van Dam/ Feiner/ Hudhes

Apuntes Opengl. Pdf
OpenGL Tutorial for 308-557.htm
OpenGL, glut, and Win XP.htm
Mínimos cuadrados.pdf

<http://www.mysteries-megasite.com/linux/C-tutorial.html>

<http://www.codeguru.com/opengl/>

<http://www.buddyydog.org/C++Builder/c++builder.html>

<http://m3d.uib.es/grafica2/Ejemplos.html>

http://www.3dlinks.com/hardware_scanners.cfm

<http://www.alice.org/>

<http://www.xmission.com/~nate/opengl.html>

<http://www.ap.dyndns.dk/software.htm>

<http://pyopengl.sourceforge.net/documentation/ref/glu.html>

<http://bdn.borland.com/article/0,1410,10528,00.html>

<http://www.beyondlogic.org/parlcd/parlcd.htm>

<http://www.stockeryale.com/i/lasers/products/snf.htm>