



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO



Facultad de Ingeniería

“APLICACIÓN DE LA VISIÓN
POR COMPUTADORA EN
TIEMPO REAL
EN UN
PROCESO DE MEDICIÓN”

T E S I S
PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
PRESENTA:
GABRIEL CHÁVEZ RUBIO

DIRECTOR DE TESIS: M. BENJAMÍN VALERA OROZCO

Septiembre, 2004





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A la Universidad Nacional Autónoma de México, la “Universidad de México”, por haberme brindado el apoyo que cualquier persona necesita para llegar a ser un buen profesionista.

Al Centro de Ciencias Aplicadas y Desarrollo Tecnológico, CCADET, por abrirme sus puertas para la realización el presente proyecto.

Al programa de Apoyo a Proyectos de Investigación Tecnológica, PAPIIT Num. IN111001 de la Dirección General de Asuntos del Personal Académico DGAPA, UNAM, que brindó los medios para que este pequeño sueño se lograra materializar.

Al departamento de metrología del CCADET, M. en I. Benjamín Valera, Ing. Sergio Padilla, Ing. Rigoberto Nava, Ing. G. Ruiz, Ing. José Sánchez a “Betí” y “El Sr. Rafael”, por sus consejos, sugerencias y apoyos en general durante el tiempo de realización de esta tesis.

Al Dr. Víctor García Garduño por haber confiado en la propuesta presentada y haberla apoyada hasta el final.

En especial al M. en I. BENJAMÍN VALERA OROZCO, por su paciencia y dedicación, por sus consejos y apoyos, por la confianza que me brindó en todo momento y me ayudó a salir adelante, por su sencillez, por ser un gran profesor, por guiarme hacia el camino que yo buscaba, por todo lo que significó para mí el tiempo que duró el desarrollo de este trabajo, a todos...

...GRACIAS

DEDICATORIAS

A mi madre:

Por que con su esfuerzo logró que hoy pudiera escribir estas palabras, por enseñarme que la persistencia logra buenos objetivos, que la fuerza del carácter crea oportunidades, por apoyarme en todo momento, bueno o malo, fácil o difícil, por hacerme sentir su apoyo, pero sobre todo, por ser ella, por ser mi madre...

A mi tío:

Que ha fungido como padre para mi durante todos estos años, que me ha brindado todo su amor y me ha enseñado a brindárselo a los demás, que me ha enseñado a tener la suficiente paciencia para alcanzar mis objetivos. A mi tío...que me a apoyado en todo momento y sé que lo seguirá haciendo. Por que ha sido una de las dos principales motivaciones para que hoy presente este trabajo.

A los CCH'ros:

Pero antes que eso, a mis amigos: Adrián Romero, Gerardo Enciso, Guillermo Garro, Héctor López, Irving Herrera, Juan Carlos, Ignacio y Noe por ser parte fundamental de lo que he llegado a hacer hasta ahora.

A los Ingenieros (o que pronto lo serán):

Ing. Haydee Bravo	Ing. Eugenia Leon
Ing. Vianey Martinez	Ing. Andres Olgin
Ing. José Luis Carrillo	Ing. Vianey López
Ing. Elizabeth Morales	Ing. Gilberto Dominguez
Ing. Miriam Porta	Ing. Claudia e Ing. Shair
Ing. Mónica Velasco	Ing. Mayra
Ing. Esther Barrios	Ing. Mariano Marco Antonio etc...

Ing. Francisco García, por ser un gran apoyo y motivación, por mostrarme su compañerismo, pero sobre todo por mostrarme su amistad en todo momento.

A la "Cosa Bonita" . . .

Índice

Introducción	1
Objetivo	1
Definición del problema	1
Antecedentes	1
Descripción del problema a resolver	2
Relevancia y justificación	3
Método	4
Resultados esperados	5
Alcance y limitaciones	5
Resumen de la tesis	5
1. Conceptos básicos	7
1.1. Metrología dimensional	7
1.1.1. La razón de medir	8
1.1.2. Calibraciones y aseguramiento de la calidad	9
1.2. Instrumentos básicos de medición dimensional	13
1.2.1. Reglas y escalas graduadas	13
1.2.2. Bloques patrón	15
1.3. Principios de visión por computadora	17
1.3.1. Fundamentos de la imagen digital	18
1.3.2. Geometría de la imagen	19
1.3.3. Reconocimiento de patrones	24

1.4. Control Digital	26
1.4.1. Fundamentos del control digital	27
1.4.2. Controlador PID utilizado	29
2. Descripción de la máquina para la calibración de reglas y escalas graduadas de alta exactitud	33
2.1. Introducción	33
2.2. Sistema mecánico	35
2.3. Sistema electrónico	38
2.3.1. Codificador lineal Acu-rite	39
2.3.2. Tarjeta de interfase PC7266	42
2.3.3. Impulsor de motores	46
2.4. Sistema de visualización	48
3. Descripción del sistema propuesto para la medición	51
3.1. Introducción	51
3.2. Calibración de cámara	53
3.2.1. Extracción de características	55
3.2.2. Estimación de la distorsión	58
3.2.3. Compensación de la distorsión	61
3.3. Extracción de la medición	63
3.3.1. Simplificación de la imagen	64
3.3.2. Análisis de la imagen	65
3.4. Control de posición	67
4. Descripción del software desarrollado	71
4.1. Introducción	71
4.2. Descripción en el ámbito del operador	73
4.2.1. Programa “Calibración de cámara 2”	73
4.2.2. Proceso de calibración	75

4.2.3.	Programa “MCReg Final 2”	78
4.2.4.	Proceso de medición	83
4.3.	Descripción en el ámbito del programador	84
4.3.1.	Programa “Calibración de cámara 2”	85
4.3.2.	Programa “MCReg final 2”	94
5.	Resultados y conclusiones	109
5.1.	Resultados	109
5.1.1.	Calibración de cámara	110
5.1.2.	Extracción de la medición	114
5.1.3.	Control de posición	117
5.1.4.	Operación en tiempo real	117
5.2.	Conclusiones	118
5.3.	Trabajo a futuro	120
	Bibliografía	123

Introducción

Objetivo

Desarrollar e implementar algoritmos de visión por computadora y control digital para asistir el proceso de medición de escalas graduadas sobre la base de un sistema con operación en tiempo real.

Definición del problema

Desde hace ya algunos años, la visión por computadora ha obtenido un gran auge debido al potencial que se puede alcanzar, el abaratamiento de la tecnología y la capacidad de procesar instrucciones cada vez más rápido han hecho posible que muchos de los algoritmos matemáticos para el tratamiento de imágenes se hayan podido implementar en sistemas cada vez más diversos y con ello obtener resultados precisos, rápidos, económicos y confiables.

Tal es el caso del presente trabajo, en el cual se desarrollaron e implementaron algoritmos de visión por computadora y control digital para automatizar el proceso de medición de escalas graduadas.

Antecedentes

En el Laboratorio de Metrología del CCADET¹, se ha desarrollado y construido un prototipo mecánico de una máquina para la calibración de reglas y escalas graduadas de alta exactitud. La atención a la calibración de reglas y escalas graduadas forma parte de un grupo de servicios estándar que le son solicitados constantemente por la industria como parte de sus sistemas de aseguramiento de la calidad. Este sistema de calibración fue pensado como una “máquina de elaborar informes”, en donde la materia prima son reglas o escalas graduadas y el producto final son informes de calibración.

¹Centro de Ciencias Aplicadas y Desarrollo Tecnológico, UNAM.

En fechas recientes, el laboratorio logró semi-automatizar el movimiento de la máquina [1, 22]. Sin embargo, su operación sigue requiriendo de la intervención humana, ya que el registro de la medición demanda la observación directa por parte del operador y un posicionamiento semiautomático, figura 1.



Figura 1. Máquina para la calibración de escalas graduadas.

Descripción de problema a resolver

La calibración de escalas graduadas consiste en un proceso minucioso de observación y comparación entre la escala a medir y un patrón de longitud de mayor precisión. La información final se forma con la obtención de las diferencias entre la medida estándar y la escala a lo largo de toda su longitud. En un proceso típico, se pueden obtener mediciones en intervalos de, por ejemplo, 1 cm en una regla de 1 m, con lo cual la medición comprende tres series de 100 lecturas cada una. Al final las series se promedian obteniendo así la curva de error que caracteriza a la escala bajo inspección. El proceso descrito se puede apreciar gráficamente en la figura 2.

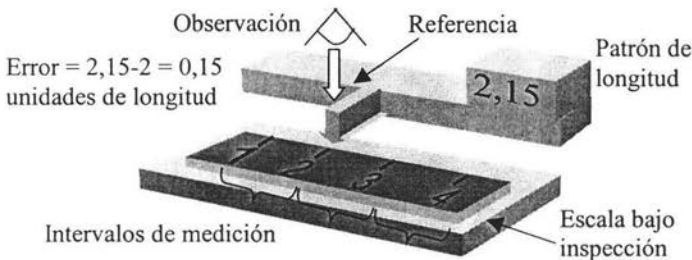


Figura 2. Proceso de calibración de escalas graduadas.

En la calibración, deben considerarse y controlarse minuciosamente las variables que influyen en la formación de incertidumbre. Algunas de estas variables son las siguientes:

- 1) Variaciones de temperatura durante la calibración.
- 2) Alineamientos mecánicos al comparar contra patrones
- 3) Deficiencias en los trazos grabados en las reglas
- 4) Distorsión en la imagen formada por los dispositivos de observación.

Los puntos 1 y 2 son resueltos como parte del diseño mecánico utilizado actualmente para la calibración. Dicho sistema garantiza la confiabilidad de la calibración de reglas o escalas graduadas de alta exactitud conforme a estándares nacionales e internacionales [12].

Los puntos 3 y 4 constituyen un obstáculo para que el operador que visualiza el proceso de medición realice mediciones exactas. La deficiencia en los trazos observados influye fuertemente para que el operador cometa errores involuntarios. El presente trabajo abordó la problemática descrita al desarrollar e implementar un sistema de visión por computadora que incrementa el desempeño en el proceso de observación. En otras palabras, el operador de la máquina para la calibración de escalas será reemplazado por visión por computadora con el propósito de incrementar la exactitud de la medición. Asimismo, el desarrollo del algoritmo de visión contempla el posicionamiento automático del dispositivo de observación sobre un intervalo de medición previamente establecido.

Relevancia y justificación

El instrumento planteado en la presente tesis ofrece un enorme ahorro de tiempo y costo de operación, ya que una vez calibrada la máquina, esta puede realizar mediciones consecutivas en línea de reglas y escalas graduadas durante todo el periodo de vigencia de su calibración que es típicamente de un año. Las calibraciones entonces pueden realizarse con un alto grado de automatización y exactitud.

El sistema de visión que se desarrolló ofrece la enorme ventaja de prescindir de personal para su operación; en lugar de ello, el proceso de medición se delegará a un sistema de visión por computadora, reduciendo los errores del tipo humano

Se planea desarrollar un instrumento de medición dimensional con alta exactitud y calidad de producto terminado. De esta forma, dotamos a la máquina con una enorme posibilidad de que el instrumento o la tecnología usada sean transferidos a la industria.

Las labores necesarias para la culminación exitosa del proyecto contemplan el desarrollo de tecnología innovadora e investigación.

Método

La figura 3 ejemplifica el esquema planteado para la medición añadiendo visión por computadora y control de posición a la máquina para la calibración de escalas. En el esquema, una computadora captura lecturas longitudinales provenientes de un transductor óptico de posición unido a la bancada de la máquina. Los errores de la escala graduada se obtienen por medio de la diferencia entre la longitud de la escala graduada y la lectura registrada en el codificador. Inicialmente, el codificador es calibrado con un sistema de mayor exactitud y resolución (sistema de interferometría láser). El programa para PC específico procesa longitudes nominales contra verdaderas. La computadora gobierna un sistema electrónico para el posicionamiento del visor sobre las marcas previamente establecidas en la escala graduada y obtiene la lectura del codificador óptico. Las lecturas de la escala graduada son obtenidas automáticamente por el sistema de visión que consta de una cámara multimedia conectada a una PC y un microscopio de aumento.

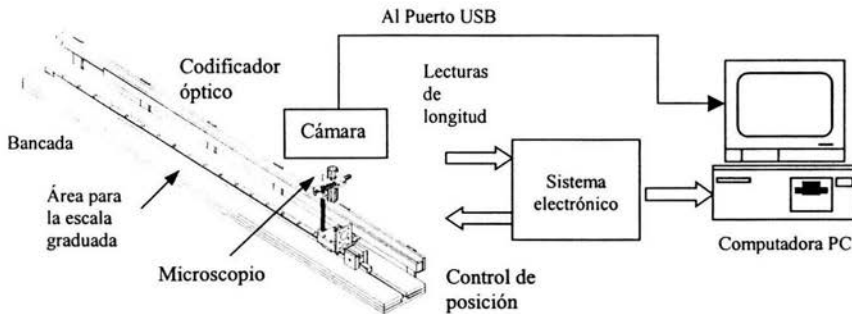


Figura 3. Visión por computadora y control digital aplicados a la máquina para la calibración de escalas graduadas.

El método planteado involucra el uso de técnicas de visión por computadora para mejorar la calidad en la imagen y para el reconocimiento de patrones. Se incluyen también técnicas de control digital para el posicionamiento del dispositivo visor sobre posiciones previamente establecidas. La calidad de la imagen es resuelta al calibrar el sistema de adquisición de imágenes compuesto por la cámara y el microscopio. Una vez compensadas las distorsiones del sistema de visión, se aplica el reconocimiento de características para obtener la medición sobre la escala graduada. Entonces la medición obtenida por el

sistema propuesto elimina los errores de carácter humano y automatiza el proceso.

Resultados esperados

Actualmente existen resultados previos al comparar el proceso de calibración de reglas graduadas utilizando la máquina en su estado original, es decir sin incluir el sistema de visión que desarrollamos. Tales resultados, basados en los procedimientos internos del Laboratorio, han producido una incertidumbre estimada en $\pm(1.0+10L)$ μm , en donde L es la longitud de la regla en m. Con la inclusión del sistema de visión, esperamos conservar la misma incertidumbre y adicionalmente planteamos las siguientes características que mejoran el desempeño del sistema original.

- Eliminación de la intervención humana.
- Eliminación de errores de apreciación.
- Automatización completa del proceso de medición.

Finalmente, deseamos construir un instrumento de medición completamente funcional con calidad de producto terminado.

Alcance y limitaciones

Una de las principales limitaciones es impuesta por la calidad de los objetos a medir; estos no deben presentar daños locales que ocasionen la aparición de artefactos nocivos al procesamiento de las imágenes como parte del sistema de visión propuesto. Asumimos que la calidad de las escalas a medir permite la extracción de características en los algoritmos de visión por computadora. Por otra parte, algunas restricciones en el algoritmo de extracción de características nos limitan a medir escalas con incertidumbre menor a $\frac{1}{2}$ de su resolución, como se verá en 3.3.

Resumen de la tesis

En el capítulo 1, se tratarán los conceptos que fueron necesarios consultar y de los que se hace uso para la elaboración del sistema de reconocimiento, de calibración, de control y de medición. Se incluyen temas matemáticos así como metrológicos, necesarios para seguir una línea en el desarrollo de la investigación e implementación.

En el capítulo 2, se describe el prototipo de la máquina de calibración de reglas y escalas graduadas de alta exactitud, como originalmente se encontraba antes de implementar el sistema propuesto. Se incluye

información tanto del sistema mecánico como eléctrico y visual. Si requiere profundizar más sobre ello se puede consultar la referencia [1].

En el capítulo 3 introducimos al lector al sistema que proponemos e implementamos, el cuál consiste en tres etapas:

- Calibración de la cámara: Donde explico con detalle el algoritmo utilizado para la corrección de distorsiones.
- Extracción de la medición: Se tratan los métodos utilizados para el reconocimiento de los patrones (líneas) que serán utilizados para la medición.
- Control de posición: Una vez corregida la imagen y reconocidas las líneas, el siguiente paso es obtener la medición y hacer el proceso automático, en este subtema se explica con detalle la forma de posicionamiento en el sistema de visualización.

El capítulo 4 trata del software que se desarrolló para lograr que el sistema funcionara en tiempo real. Se divide en dos partes: descripción en el ámbito del operador y descripción en el ámbito del programador. El primero tiene como propósito servir como manual de consulta rápida para realizar la calibración de una regla. El segundo está dirigido a toda persona interesada en conocer el código de programación necesario para la realización del sistema.

Finalmente en el capítulo 5, damos a conocer los resultados que obtuvimos al utilizar el sistema. Se muestran los resultados de calibraciones realizadas a reglas de alta exactitud utilizadas para propósitos industriales y se hace una pequeña conclusión de los mismos.

Capítulo 1

Conceptos básicos

En el presente capítulo abordamos los conceptos básicos necesarios para la elaboración de ésta tesis. Por ejemplo, el alcance del área conocida como *metrología dimensional* proporciona la pauta para el aseguramiento de la calidad de las mediciones geométricas, aspecto de gran importancia en los ámbitos industrial y científico y que no puede estar fuera del objetivo de éste proyecto. Por otra parte, el uso de instrumentos básicos de medición no sólo implica el mantenerlos en buen estado y usarlos adecuadamente, también existe un compromiso del propietario que ofrece productos o servicios por mantenerlos calibrados con trazabilidad a patrones de mayor exactitud. El proyecto de tesis implica el dominio de técnicas de visión por computadora con el objetivo de implementar los algoritmos de visión necesarios en el método para la calibración de escalas. Por otra parte, las técnicas de control digital empleadas en el posicionamiento automático de la máquina, tienen su fundamento en los principios de control digital expuestos al final del presente capítulo.

1.1. Metrología dimensional

La metrología dimensional comprende los aspectos técnicos y legales relacionados a las mediciones de longitud y ángulo. Un estudio detallado de esta área de trabajo nos permite comprender y abordar el desarrollo de los algoritmos de medición y control del presente proyecto de tesis [9, 14].

1.1.1. La razón de medir

El ser humano todo lo mide, lo ha hecho desde sus inicios. Puede decirse que ello a contribuido de manera sustancial a su desarrollo como ser social. Medir el tiempo, medir distancias, medir terrenos, son algunos ejemplos donde el hombre tuvo que concebir y dar significado a palabras como cantidad, peso, volumen, magnitud o dimensión. Puede decirse que los intereses metroológicos del hombre primitivo estaban relacionados con sus necesidades fundamentales, la necesidad de delimitar un lugar, la necesidad de intercambiar productos, la necesidad de predecir el clima para una buena cosecha por ejemplo, obligó a constituir marcos de referencia y normas de carácter universal, de tal forma que fueran adoptadas por las comunidades con las que se tenía un intercambio comercial y cultural.

Sin embargo todo ello fue de forma empírica, no fue sino hasta la época de Galileo donde las mediciones fueron realizadas de forma objetiva y científica.

Pero el inicio de la metrología como la conocemos en la actualidad, surge después de la revolución francesa con el decreto de la Asamblea Nacional Francesa, en donde se instituye su sistema nacional de pesas y medidas con la nomenclatura de sus unidades, los múltiplos y submúltiplos, entonces se establece el Sistema Métrico. En 1790 el gobierno francés ordenó a la directiva de la Academia Francesa de ciencias estudiar y proponer un sistema único de pesas y medidas para reemplazar todos los sistemas existentes. Las propuestas de la Academia Francesa fueron aprobadas e introducidas como el Sistema Métrico de Unidades de Francia en 1795. El sistema métrico despertó considerable interés en otras partes del mundo; y finalmente en 1875, 17 países firmaron la llamada Convención del Metro, adoptando legalmente el sistema métrico de unidades.

A mediados del siglo pasado, después de diversos intercambios entre los medios científicos y técnicos del mundo, la Décima Conferencia General de Pesas y Medidas adoptó como unidades fundamentales, el metro, el kilogramo, el segundo, el ampere, el kelvin y la candela.

Finalmente, fue en 1960 que la Décimo Primera Conferencia General de Pesas y Medidas creó, con su famosa resolución 12, el Sistema Internacional de Unidades (SI), basado en las seis unidades fundamentales antes mencionadas, y posteriormente se agregó una séptima: el mol. En este Sistema (SI) las unidades de medida se dividen en tres grupos:

- 1) Las unidades fundamentales que son aquellas que no se componen de otras unidades.

- 2) Las unidades complementarias que son las que no se derivan de una magnitud física pero son necesarias para su comprensión.
- 3) Las unidades derivadas que son todas las que se componen de dos o más unidades fundamentales.

Podemos decir entonces, que la creación del SI es el resultado de una larga historia en donde un gran número de personas- científicos, ingenieros y políticos- han aportado su contribución, estimulados por las exigencias crecientes de una sociedad en evolución. El SI es un sistema adaptado a las necesidades de la ciencia, de la tecnología, de la industria y del comercio y su adopción implica la obligación de conformarse cuidadosamente a la notación, a los símbolos y a las reglas adoptadas por la Conferencia General de Pesas y Medidas.

Para hacer notar la importancia del desarrollo de la metrología en un país, basta observar que toda la manufactura moderna (que cada vez se hace menos con las manos), está basada en la exactitud y en las tolerancias de las magnitudes dimensionales, ya que de ellas se parte para obtener un producto que sea efectivo y competitivo.

Poco a poco los países han impulsado el desarrollo de la metrología y han creado normas que regulan la calidad de los productos basadas en la precisión, exactitud, etc. de estos. México no ha sido la excepción.

1.1.2. Calibraciones y aseguramiento de la calidad

Todo producto terminado a tenido que ver en algún momento de su fabricación con unidades de medida tales como tiempo, masa, volumen, temperatura, presión, entre otras, por ello, en el presente trabajo utilizaremos una serie de conceptos del campo de la metrología que creo es necesario revisar. Empezamos a definir la METROLOGÍA como el campo de conocimiento relativo a las mediciones [9].

La DIMENSIÓN es la expresión numérica en unidades de medida con las que se expresan las características geométricas de un cuerpo tales como longitudes, curvaturas o ángulos, con ellos podemos conocer su forma y tamaño.

La MAGNITUD es el atributo de un fenómeno, cuerpo o sustancia que se puede distinguir y determinar cualitativa y cuantitativamente. Las magnitudes se miden.

La MEDICIÓN es un conjunto de operaciones experimentales que tienen por objeto determinar el valor de una magnitud. Este deberá sustentarse en un fundamento científico.

El PROCESO DE MEDICIÓN se refiere a toda la información, equipo y operaciones, relativos a una medición determinada. Involucra lo relativo a la ejecución de la calidad de la medición. Consta de principios, métodos, procedimientos, valores de las magnitudes de influencia, patrones de medición y cálculos.

El MÉTODO DE MEDICIÓN es el conjunto de operaciones teóricas y prácticas involucradas en la realización de las mediciones de acuerdo a un principio establecido. Los métodos de medición pueden ser de forma Directa, Indirecta, Fundamental, Por comparación directa, Por sustitución, Diferencial o Por cero.

RESULTADO DE UNA MEDICIÓN, es el resultado de una magnitud medida, obtenido en un proceso de medición. Este resultado puede ser: LA INDICACIÓN, el cual es el valor suministrado por nuestro instrumento de medición para la magnitud medida. EL RESULTADO BRUTO, que es el que no ha sido modificado con correcciones por errores sistemáticos de medición. EL RESULTADO CORREGIDO, el cual se da cuando se han efectuado correcciones necesarias por errores sistemáticos y se acompaña generalmente de la incertidumbre de medición o por último si es la medida obtenida por varias OBSERVACIONES.

La EXACTITUD DE LA MEDICIÓN es la proximidad de concordancia entre el resultado de una medición y el valor verdadero de la magnitud medida.

REPETIBILIDAD DE MEDICIONES, es la proximidad de concordancia entre resultados de mediciones sucesivas de la misma magnitud, efectuadas: Por el mismo método, por el mismo observador, con los mismos instrumentos de medición, en el mismo lugar y condiciones de uso y en intervalos cortos de tiempo. A menudo se basa en la incertidumbre de medición, esto es, en cuanto menor sea la incertidumbre, mejor la repetibilidad.

INSTRUMENTO DE MEDICIÓN. Es el dispositivo que nos permitirá efectuar las mediciones. Podemos decir que es una serie de elementos que constituyen la trayectoria de la señal medida que se inician con un sensor (como entrada) y terminan en un indicador (como salida). Este último dará el resultado de la medición o de un valor relacionado. Para ello, en muchos instrumentos este dispositivo se compone de una escala y un índice.

La ESCALA es un conjunto ordenado de trazos los cuales pueden ser líneas o signos grabados que corresponden a valores determinados en una magnitud a medir y tienen una cifra asociada.

La definición de CALIBRACIÓN según el Vocabulario Internacional de metrología (VIM) menciona que es un conjunto de operaciones que establecen, bajo condiciones específicas, la relación entre los valores indicados por un aparato o sistema de medición, o los valores presentados por una medida materializada y los valores conocidos correspondientes de una magnitud medida. Dicho de otra forma, la calibración de un instrumento de medición es encontrar el error que llegara a tener éste respecto a un patrón para así poder eliminar dicho error de las mediciones que se hagan con ese aparato.

PATRÓN, es la medida materializada, aparato de medición o sistema de medición destinados a definir, realizar, conservar o reproducir una unidad de uno o varios valores conocidos de una magnitud, para transmitirlos por comparación a otros instrumentos de medición.

La TRAZABILIDAD es la propiedad de un resultado de medición consistente en poder relacionar a los patrones apropiados, generalmente internacionales o nacionales, mediante una cadena ininterrumpida de comparaciones.

ERROR ABSOLUTO DE MEDICIÓN, es el resultado de una medición menos el valor convencionalmente verdadero de la magnitud medida.

El ERROR ALEATORIO, es un componente del error de medición, que durante un número de mediciones del mismo mensurado, varía de manera imprevisible, este error no es posible corregir.

ERROR SISTEMÁTICO, al igual que el error anterior este, es un componente del error de medición, pero a diferencia del anterior permanece constante o varía de manera constante o en forma previsible.

CORRECCIÓN, es el valor agregado algebraicamente al resultado bruto de medición que compensa un error sistemático supuesto.

La INCERTIDUMBRE DE MEDICIÓN, es la estimación que caracteriza el intervalo de valores dentro de los cuales se encuentra el valor verdadero de la magnitud medida, esta estimación por lo general se basa en la distribución estadística de los resultados de series de mediciones y pueden estar caracterizados por desviaciones estándar experimentales.

La calibración de instrumentos en general resulta ser de gran importancia, ya que de ello depende la presión, exactitud y calidad con la que contara el producto terminado. Todo equipo nuevo debe calibrarse antes de ponerse en servicio, ya sea en su lugar de origen o en el lugar donde va a ser empleado, de cualquier manera, el fabricante

deberá garantizar la trazabilidad al patrón nacional o internacional de medición.

También es posible hacer una recalibración, la cuál se hace con el fin de asegurar la continuidad de la validez de las mediciones hechas por el instrumento en cuestión.

Para lograr la calibración de un instrumento, es necesario encontrar las posibles fuentes de error y asignar cada una de ellas a una variable, al final todas ellas formaran la incertidumbre del instrumento logrando con ello la calibración. Se tienen consideradas tres acciones metrologías para lograr calibrar un instrumento:

- Determinación de las variables que pueden contribuir a formar la incertidumbre del instrumento.
- Medición del error aportado por cada variable.
- Determinar el efecto neto de la interacción de las variables en la capacidad de medición del instrumento.

Una vez obtenido el resultado el instrumento estará en condición de ser utilizado, considerando que los resultados arrojados tendrán que ser corregidos según los datos que haya arrojado la calibración.

Por otro lado, la imprecisión inevitable de los procesos de manufactura hace que una pieza no pueda especificarse con una dimensión fija, sino que dicha dimensión puede darse entre dos dimensiones límite, compatibles con un funcionamiento correcto de la pieza. La diferencia entre esas dos dimensiones constituye la TOLERANCIA. Por norma, debe existir una tolerancia al fabricar un producto que permita el buen funcionamiento de este y se le llegue a considerar un producto de calidad.

Para cualquier medición nominal, está prevista toda una gama de tolerancias. El valor de estas tolerancias está simbolizada por un número de "calidad" que puede ser alguna de las 18 cantidades como 01, 0, 1, ..., 16 correspondientes cada una a los grados de tolerancia fundamentales: ITO1, IT0, ..., IT15 o IT16.

Estas tolerancias deberán ser consideradas a la hora de generar el reporte de la medición. Un instrumento de medición que presente resultados dentro de la tolerancia establecida por alguna norma relacionada, podrá considerarse como un instrumento con un cierto nivel de calidad listo para poder utilizarse.

1.2. Instrumentos básicos de medición dimensional

Los instrumentos básicos de medición son utilizados ampliamente en el ámbito industrial, comercial y científico. Según lo expuesto anteriormente, estos deben ser usados adecuadamente y calibrados periódicamente.

1.2.1. Reglas y escalas graduadas

La medición de longitud es un proceso de comparación de la distancia a ser medida con una dimensión longitudinal conocida y el camino más simple para llevar a cabo esta operación es por comparación directa.

Los cuerpos rectos de longitud definida, como las reglas y las escalas, que tienen graduaciones en subdivisiones conocidas de la unidad básica para medición de longitud, están en un principio bien adaptados para los propósitos de tales comparaciones, figura 1.1. Comenzando en un punto fijo cero en la escala el cual es alineado con un extremo de la distancia a ser medida, la graduación que queda en correspondencia con el otro extremo de la distancia, indicará la longitud. Los números de la escala y las graduaciones grabadas a diferentes alturas, ayudan a atribuir un valor numérico a la graduación correspondiente.

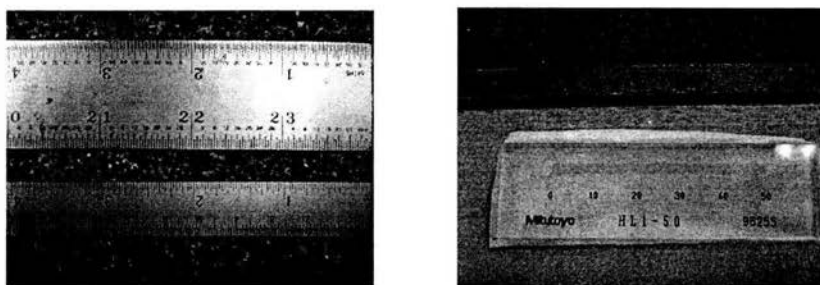


Figura 1.1. Reglas y escalas graduadas.

Este método pareciera ser muy obvio y sencillo, sin embargo se debe tomar en cuenta las diversas limitaciones que son inherentes a las mediciones de longitud por comparación directa. Sin una cierta consideración de estas limitaciones, se puede presentar una falta de exactitud debido a alguno de los siguientes factores:

- Limitaciones del instrumento: Estos pueden presentar deficiencias geométricas resultado de errores de planitud y paralelismo o causadas por deflexión. Falta de exactitud de las graduaciones de la escala. Excesivo grosor o definición pobre de los trazos y/o

sensibilidad limitada por menor incremento de la graduación de la escala, esto es lectura mínima.

- Errores observacionales: Se presentan al existir un alineamiento deficiente debido a una coincidencia impropia de los límites con las graduaciones de la escala seleccionada. También se refieren a errores de paralaje cuando al observar la escala y el objeto en una dirección no esta perfectamente perpendicular a la superficie que está siendo medida.

Los errores observacionales pueden ser reducidos por medio de dispositivos auxiliares, los cuales proveen el alineamiento de la vista y permiten tener bajo control la operación de medir. En general existen normas nacionales e internacionales que definen los errores permitidos en barras y reglas de medición, como por ejemplo podemos observar la tabla 1.1.

LONGITUD DE MEDICION (pulgadas)	ERROR DE GRADUACIÓN PERMISIBLE MAX (pulgadas)			
	Barras de Referencia	Barras de Inspección	Reglas Grado I	Reglas Grado II
4	± 0.00022	± 0.00044	± 0.00088	± 0.0022
8	± 0.00024	± 0.00048	± 0.00096	± 0.0024
20	± 0.00030	± 0.00060	± 0.00120	± 0.0030
40	± 0.00040	± 0.00080	± 0.00160	± 0.0040
FORMULA GENERAL	L = Longitud total de la graduación			
	$\pm (0.0002 + \frac{L}{200000})$	$\pm (0.0004 + \frac{L}{200000})$	$\pm (0.0008 + \frac{L}{200000})$	$\pm (0.002 + \frac{L}{200000})$

Tabla 1.1. Errores en escalas graduadas.

En la tabla 1.1 podemos encontrar una clasificación de las reglas y barras más utilizadas en el mundo de la metrología.

Las barras de referencia graduadas, pueden contener deflexiones que contribuyan a un mayor error, por ello estas barras son de sección transversal "H", "U" y "X" y las marcas están hechas en el plano neutro de la sección transversal. En ningún lado de la barra de referencia, las graduaciones deben extenderse hasta el extremo de la barra mínima. La superficie graduada debe estar plana con pocas diezmilésimas de pulgada de error y las líneas de graduación deben estar precisamente paralelas una de otra. El efecto de "pandeo" se debe tratar de eliminar soportando firmemente estas barras.

Las barras de inspección graduadas tienen como propósito primordial inspeccionar la exactitud de las reglas de medición graduadas comunes. Su sección transversal es cuadrada, la longitud por

lado de éstas varía como una función del largo total de la barra. La longitud de la barra debe extenderse en ambos extremos más allá de las líneas de graduación. Las líneas están hechas a manera de alcanzar el borde de la barra a lo largo de uno de los lados, los números de graduación están derechos en tal posición de la barra donde las líneas de graduación están sobre el borde más lejano. Este arreglo facilitará la comparación de las graduaciones por acercamiento de los bordes de la barra de inspección y de la regla por ser verificada.

Las reglas de trabajo graduadas, figura 1.2, sirven para referencia cuando se transfieren dimensiones longitudinales o para mediciones directas con el borde de la regla en contacto con el objeto a ser medido. La sección transversal de estas reglas es rectangular con una relación de espesor a ancho alrededor de 1:5. De acuerdo con su propósito primordial de uso, las reglas de trabajo con graduación lineal se fabrican en dos diferentes grados de exactitud los cuales se refieren a diferentes grados de precisión según sea necesario.

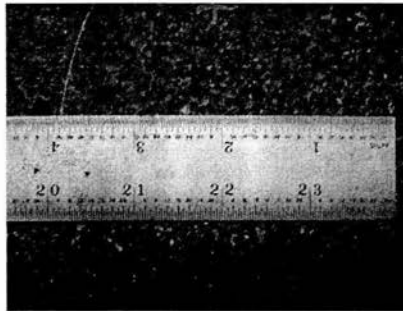


Figura 1.2. Regla de trabajo graduada.

Las graduaciones en reglas de acero de exactitud controlada, se logran con máquinas de graduación especiales para grabar con tolerancia establecidas de las barras patrón del fabricante. Las barras patrón deberán estar contrastadas con los patrones de referencia del organismo nacional de calibración vigente.

1.2.2. Bloques patrón

Los bloques de calibración, como también se les conoce, son los medios por los cuales se da forma física a los patrones de medición de longitud y de los que se dispone para las mediciones prácticas, figura 1.3.

Como comentamos anteriormente, a medida que la producción de bienes se hizo masiva, fue necesario que los patrones universalmente aceptados fueran distribuidos según la demanda solicitada, la cual se incrementaba constantemente. Para ello se tuvieron que perfeccionar

los métodos de manufactura de patrones que pudieran ser usados para tal fin. Se probaron con diferentes materiales y formas hasta llegar finalmente a estandarizarse en una serie de normas que regulan su fabricación y utilización.

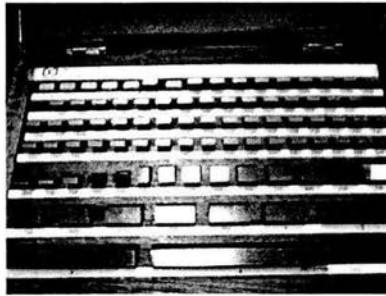


Figura 1.3. Bloques patrón.

Físicamente son piezas rectangulares o cuadradas cuya distancia de los planos de medición, así como su planitud y paralelismo son muy exactos. Son fabricados de acero, carburos o acero cromado y en su manufactura se emplea el método “Johansson” que produce superficies planas con un grado de acabado sumamente fino que tiene el aspecto de plata bruñida. Esto es interesante nombrar, ya que este acabado, se aproxima más al plano perfecto que cualquier otra superficie metálica producida por el hombre. Además se les aplican tratamientos térmicos y de envejecimiento efectivos de los aceros, lo que minimiza sus cambios dimensionales y les confiere alta dureza.

Dado su alto grado de exactitud, estos bloques se emplean para distintos trabajos de verificación y de acuerdo a las normas de referencia, por ejemplo:

- Grado ISO “Calibración” o GGG-0.5. Este tipo de patrones son especiales para trabajo de calibración de precisión extremadamente alta. No se recomiendan para uso general y deberán ser ordenados individualmente. Generalmente se utilizan como patrones de referencia de longitud o “primarios”.
- Grado ISO-00 o GGG-1. Son patrones de referencia de laboratorio. Se utilizan para calibrar los bloques calibradores de inspección y para trabajos de verificación de alta precisión.
- Grado ISO-0 o GGG-2. Generalmente se usan como bloques calibradores de inspección y como patrones en el taller de herramientas.

- Grado ISO-1. Utilizados como patrones para taller, uso más común.

Para efectuar mediciones exactas, los bloques calibradores deberán estar a la misma temperatura que la pieza a medir, si son del mismo material o a 20°C si son de materiales diferentes; también se puede corregir por temperatura cuando se trate de materiales diferentes y temperatura distinta a 20°C. El mantenimiento de los bloques, se hace a intervalos de tiempo según sea su grado de tolerancia, este dato lo da el fabricante, como ejemplo, si se tiene un grado de tolerancia de 0.5 o 1, la verificación y la recalibración, según sea necesario, se hará anualmente, para un grado de tolerancia de 2, se hará mensualmente o semianual dependiendo de la frecuencia de uso.

La recalibración determinará la necesidad de reemplazar uno o más bloques del juego cuando estos se desvían más del doble de las tolerancias establecidas en longitud, planitud o paralelismo. Se hace notar que un solo bloque fuera de tolerancia sitúa a todo el juego fuera de grado.

Los bloques se comprueban en longitud mediante interferometría o mediante comparadores de superficie-exactitud que indiquen cambios de 0.000 0001 pulgadas, su planitud se comprueba con un plano óptico de referencia y el paralelismo se determina usando un interferómetro y dos planos ópticos o con un comparador de alta amplificación. En general la Norma *Federal Specification GGG-G-15C* y la monografía *NBS 163* marcan las guías y procedimientos para efectuar las mediciones de verificación y recalibración de los bloques.

1.3. Principios de visión por computadora

El interés por los métodos de tratamiento digital de imágenes deriva de dos áreas principales de aplicación: la mejora de la información pictórica para la interpretación humana y el procesamiento de los datos de la escena para la percepción autónoma por una máquina. Estas aplicaciones se vienen dando desde principio del siglo XX, con el envío de imágenes a nivel transnacional. Dichas imágenes tenían que ser digitalizadas y procesadas tanto para su envío como para su recepción. A lo largo de los años y más intensamente a partir del de la década de los años 60, estos métodos han sido perfeccionados, sin embargo los elementos de un sistema de procesamiento digital de imágenes siguen siendo los mismos. Estos elementos de los sistemas consisten en 5 etapas diferentes que son: 1) Adquisición 2) Almacenamiento, 3) Procesamiento 4) Comunicación y 5) Presentación. En este trabajo nos enfocaremos principalmente al

procesamiento de imágenes y será aplicado a un sistema de medición. A continuación describiremos algunos conceptos necesarios en el tratamiento de imágenes digitales [10].

1.3.1. Fundamentos de la imagen digital

El término *imagen monocromática* o simplemente *imagen*, se refiere a una función bidimensional de la luz y la intensidad, a la que indicamos por $f(x, y)$, donde el valor o amplitud de f en las coordenadas espaciales (x, y) da la intensidad o iluminación de la imagen en este punto. Se considera que la $f(x, y)$ debe ser estrictamente mayor que cero y finita, ya que la luz es una forma de energía, esto es:

$$0 < f(x, y) < \infty$$

A cada elemento de esta función se le llama elemento de la imagen o más comúnmente *pixeles*.

Para que esta función pueda ser utilizable por el procesamiento por medio de computadoras debe ser digitalizada tanto espacialmente como en su amplitud. La digitalización de las coordenadas espaciales (x, y) se denomina “muestreo” de la imagen y la digitalización de la amplitud se conoce bajo el nombre de “cuantificación” del nivel gris. La función $f(x, y)$ se describe de forma aproximada por una serie de muestras igualmente espaciadas organizadas en forma de matriz $N \times M$ donde cada elemento de la matriz en una cantidad discreta y representa a cada píxel. Esto es:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N,0) & f(N,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Observando la ecuación anterior, para definir una imagen es necesario conocer los valores de N , M y de cada uno de los píxeles, ahora bien, comentamos que los valores de $f(x, y)$ deben estar entre 0 e infinito, sin embargo para fines prácticos, estos valores deben estar limitados en un dominio entre 0 y un número no muy grande. Para una imagen monocroma estos valores nos dirán intensidades ente el blanco absoluto y el negro absoluto, esto es nos dirán los niveles de grises de la imagen. En el procesamiento digital de imágenes es practica común que estas cantidades sean potencias enteras de 2 es decir:

$$N = 2^n \quad , \quad M = 2^k \quad \text{y} \quad G = 2^m$$

Resumiendo N es el número de filas, M el número de columnas y G el número de niveles de gris. Los valores de n , k y m varían según la

aplicación de la imagen o del dispositivo utilizado para su adquisición. En nuestro caso se utilizaran imágenes de 320x240 con 256 niveles de gris, esto es $N=240$, $M=320$ y $G=256$.

1.3.2. Geometría de la imagen

En esta sección veremos las operaciones básicas que se pueden realizar sobre los sistemas coordenados, estos es, las operaciones geométricas que transforman el ambiente global 3D a proyecciones planas [10, 17].

Se han desarrollado una serie de transformaciones básicas sobre la matriz que caracteriza a un sistema coordenado que son de suma importancia ya que al poder trasladar o modificar un punto, se podrá hacerlo también sobre toda la escena.

Para un sistema de tres dimensiones, estas operaciones quedan definidas por la siguiente transformación lineal:

$$[x' \ y' \ z' \ h] = [x \ y \ z \ 1] [T]$$

donde:

x', y', z' = Coordenadas del nuevo sistema de referencia

x, y, z = Coordenadas originales (a las coordenadas $[x \ y \ z \ 1]$ se le conoce como *coordenadas homogéneas*).

T = Matriz de transformación.

La matriz T produce la rotación, translación, reflexión, ajuste global y ajuste local de la imagen. Luego la transformación de coordenadas homogéneas a coordenadas ordinarias estará dada por:

$$[x^* \ y^* \ z^* \ 1] = \begin{bmatrix} x' & y' & z' \\ h & h & h \\ & & 1 \end{bmatrix}$$

La matriz de transformación T , para coordenadas homogéneas de tres dimensiones se puede dividir en 4 partes:

$$T = \begin{bmatrix} & & & 3 \\ & 3 \times 3 & & \times \\ & & & 1 \\ \text{---} & \text{---} & \text{---} & \text{---} \\ 1 \times 3 & & & 1 \times 1 \end{bmatrix}$$

donde:

La matriz de 3 x 3 puede producir la rotación, ajuste de escala, o reflexión.

La matriz de 1 x 3 produce la translación.

La matriz de 3 x 1 puede producir una transformada de perspectiva.

La matriz de 1 x 1 produce un ajuste de escala global.

El poder trabajar con una matriz de transformación cuadrada de 4 x 4 nos permite entre otras cosas concatenar las transformaciones geométricas. Esto es, en el caso de que a un conjunto de puntos se le deba aplicar una secuencia de transformaciones, se puede multiplicar previamente las matrices, aplicando la transformación resultante a los puntos. De este modo se aplica una única transformación a cada punto. Hay que tomar en cuenta que estas matrices generalmente no son conmutativas, por lo que el orden de aplicación es importante. A continuación, una descripción de las transformaciones descritas

Ajuste de la escala

Se le conoce como ajuste de la escala al poder ajustar cada eje independientemente a un tamaño deseado. Este ajuste está dado por la diagonal de la matriz de transformación, esto es:

$$[X'] = [X][T] = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} ax & by & cz & 1 \end{bmatrix}$$

donde:

a , b y c son valores o números reales. Si estos números son:
 menores a 1, el ajuste decrece
 iguales a 1, el ajuste se mantiene
 mayores a 1, el ajuste aumenta

Ajuste global

El ajuste global es el cambio de tamaño igual para los tres ejes. Esto es, el objeto cambia proporcionalmente de tamaño. Este cambio lo da la submatriz de 1 x 1 de la matriz de transformación, esto es:

$$[X'] = [X][T] = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix} = \begin{bmatrix} x & y & z & s \end{bmatrix}$$

por lo tanto:

$$\begin{bmatrix} x^* & y^* & z^* & 1 \end{bmatrix} = \begin{bmatrix} x' & y' & z' & 1 \\ s & s & s & 1 \end{bmatrix}$$

donde:

s es un número real. Entonces si:

$s < 1$	aumenta
$s = 1$	se mantiene igual
$s > 1$	disminuye

Translación

Se produce al modificar la submatriz de 1 x 3, produciendo un desplazamiento en las coordenadas x , y y z .

$$\begin{bmatrix} X' \end{bmatrix} = \begin{bmatrix} X \end{bmatrix} \begin{bmatrix} T \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix} = \begin{bmatrix} x+l & y+m & z+n & 1 \end{bmatrix}$$

en donde l , m , n es la cantidad a desplazar en los ejes x , y , z respectivamente.

Rotación

El caso más simple de la rotación, es la que realiza un punto sobre un eje de coordenadas, figura 1.4. Para el caso de tres dimensiones, las matrices de transformación quedan de la siguiente forma:

$$\begin{bmatrix} T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} T \end{bmatrix} = \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} T \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi & 0 & 0 \\ -\sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotación un ángulo θ en x Rotación un ángulo ϕ en y Rotación un ángulo ψ en z

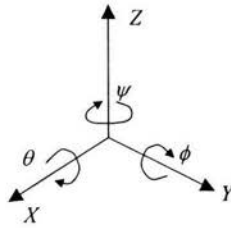


Figura 1.4. Rotación sobre los ejes coordenados.

Reflexión

Puede verse como la generación de una imagen “espejo” a través de cierto plano es equivalente a la rotación en un eje coordenado

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reflexión en xy

$$[T] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reflexión en yz

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reflexión en xz

Transformaciones de perspectiva

Sobre la imagen también se puede hacer transformaciones de perspectiva, las cuales proyectan puntos del espacio tridimensional sobre un plano. Las transformaciones de perspectiva desempeñan un papel central en el procesamiento de imágenes porque proporcionan una aproximación al modo en que se forma una imagen viendo el mundo tridimensional. Estas transformaciones varían a las presentadas anteriormente por que no son lineales en el sentido que implican la división por los valores de las coordenadas.

En la figura 1.5 se presenta el modelo del proceso de la formación de la imagen.

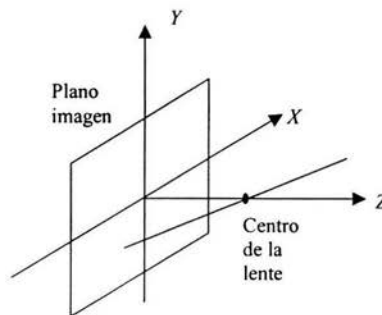


Figura 1.5. Proceso de la formación de la imagen.

El sistema de coordenadas de la cámara (x, y, z) tiene el plano imagen coincidente con el plano xy , mientras que el eje óptico (determinado por el centro de la lente) coincide con el eje z . De este forma el centro del plano imagen está en el origen y el centro de la lente queda en las coordenadas $(0, 0, \lambda)$. Si la cámara está enfocada sobre objetos distantes, λ es la distancia focal de la lente. Aquí se supone que el sistema de coordenadas de la cámara está alineado con el sistema de coordenadas global (X, Y, Z) , mas adelante veremos que esto no es necesariamente así.

Considerando que (X, Y, Z) son coordenadas globales de cualquier punto en un escenario tridimensional, se supondrá que $Z > \lambda$, es decir, que todos los puntos de interés quedan por delante de la lente. El primer paso es obtener una relación que proporcione las coordenadas (x, y) de la proyección del punto (X, Y, Z) sobre el plano imagen. Por semejanza de triángulos tendremos:

$$\frac{x}{\lambda} = \frac{X}{Z - \lambda} \quad \frac{y}{\lambda} = \frac{Y}{Z - \lambda}$$

Por lo tanto, las coordenadas en el plano imagen del punto 3D proyectado se obtienen directamente de las ecuaciones anteriores:

$$x = \frac{\lambda X}{\lambda - Z} \quad y = \frac{\lambda Y}{\lambda - Z}$$

Matricialmente estas ecuaciones se pueden presentar como sigue:

$$c = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{\lambda X}{\lambda - Z} \\ \frac{\lambda Y}{\lambda - Z} \\ \frac{\lambda Z}{\lambda - Z} \end{bmatrix}$$

donde c representa a las coordenadas en el plano imagen de la proyección de un punto (X, Y, Z) del espacio tridimensional. La transformación inversa queda como sigue:

$$w = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{\lambda x}{\lambda + z} \\ \frac{\lambda y}{\lambda + z} \\ \frac{\lambda z}{\lambda + z} \end{bmatrix}$$

sin embargo como se dijo anteriormente el eje z de ambos planos es coincidente, por lo tanto las ecuaciones se pueden expresar de la siguiente forma:

$$\begin{bmatrix} X = \frac{\lambda x}{\lambda + Z} \\ Y = \frac{\lambda y}{\lambda + Z} \\ Z = \frac{\lambda z}{\lambda + Z} \end{bmatrix}$$

despejando z en función de Z en la última ecuación y sustituyendo este resultado en las dos primeras expresiones se obtiene:

$$X = \frac{x}{\lambda} (\lambda - Z) \qquad Y = \frac{y}{\lambda} (\lambda - Z)$$

1.3.3. Reconocimiento de patrones

Mucha de la información que se maneja en la vida real se presenta en la forma de patrones complejos: caras, textos escritos, enfermedades, música, flores, piezas industriales, música, etc [8].

Para la psicología, el problema central en el ámbito del reconocimiento de patrones es el estudio de los mecanismos por los que las señales externas estimulan los órganos sensoriales y se convierten en experiencias perceptuales significativas, o dicho de otra forma, cómo realizamos el etiquetado de estos estímulos tan complejos asignándoles un nombre. Estos procesos continúan siendo desconocidos en su mayor parte y no se ha encontrado un modelo concluyente sobre cómo nuestro sistema nervioso realiza este reconocimiento. No obstante, se admite que esta tarea debe realizarse siguiendo un esquema general como el que se detalla a continuación: Antes del reconocimiento, un patrón debe ser percibido por los órganos sensoriales. Además, el mismo patrón o alguno similar (de la misma clase) debe haberse percibido y recordado previamente. Finalmente, debe establecerse alguna correspondencia entre la percepción actual y lo recordado.

En resumen, esta aproximación al reconocimiento de patrones se centra en el estudio del mecanismo de reconocimiento presente en los seres vivos. Una definición técnica sencilla del reconocimiento de patrones puede ser la siguiente:

“Es la comparación de una señal, que representa un fenómeno, con un patrón almacenado previamente en algún tipo de memoria”

El uso intensivo de computadoras y otros dispositivos electrónicos en los últimos años ha impulsado el estudio y aplicación de técnicas de reconocimiento de patrones. En particular los dispositivos de adquisición de datos (sensores y transductores) y los convertidores Analógico / Digital hacen que una computadora pueda ser “alimentada” con datos (observaciones) del mundo real, utilizarla para almacenar y recuperar información y para establecer la correspondencia entre observaciones pasadas y actuales.

Aunque la aplicabilidad de las técnicas resulta en un principio muy amplia, no hay un método que sea la panacea. Diversas razones hacen que los sistemas de reconocimiento de formas operativas sean muy específicos del problema a resolver, por ejemplo:

- La naturaleza de los patrones: caracteres escritos, símbolos, dibujos, imágenes biomédicas, objetos tridimensionales, firmas, huellas dactilares, espectrogramas, imágenes de teledetección, cromosomas, etc.
- Los requerimientos del sistema: especialmente el tiempo de respuesta hace que algunos métodos de reconocimiento, aún siendo superiores en éxito, no sean aplicables en la práctica.
- Factores económicos: un sistema equipado con diferentes sensores y equipos de procesamiento muy potentes pueden dar resultados muy satisfactorios pero no pueden ser adquiridos por los usuarios.

Estos factores hacen que un sistema adecuado para un problema sea inaplicable para otro, lo que posibilita el estudio y desarrollo de nuevas técnicas.

Los procesos preceptuales del ser humano pueden ser modelados como un sistema de tres estados:

- 1) adquisición de datos sensoriales
- 2) extracción de características
- 3) toma de decisiones

Por lo tanto es conveniente dividir el problema del reconocimiento automático de una manera similar. Gráficamente lo podemos observar en la figura 1.6.

Sensor. Proporciona una representación de los elementos del universo a ser clasificados. Idealmente uno deberá entender completamente las propiedades físicas que distinguen a los elementos en las diferentes clases y usar ese conocimiento para diseñar el sensor, de manera que esas propiedades pudieran ser medidas directamente. En

la práctica frecuentemente esto es imposible porque: no se dispone de ese conocimiento, muchas propiedades útiles no se pueden medir directamente y no es económicamente viable.

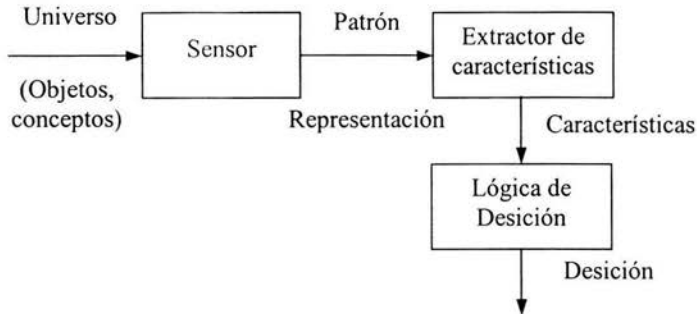


Figura 1.6. Reconocimiento automático de patrones.

Extracción de características. Esta etapa se encarga, a partir del patrón de representación, de extraer la información discriminadora eliminando la información redundante e irrelevante. Su principal propósito es reducir la dimensionalidad del problema de reconocimiento de patrones.

Clasificador. Es la etapa de toma de decisiones en el sistema. Su rol es asignar a la categoría apropiada los patrones de clase desconocida a priori.

En nuestro caso, básicamente utilizamos dos dispositivos que nos sirven para obtener información (sensores), un codificador con una resolución de 2 μm y una cámara digital. La extracción de características se da por medio del software desarrollado, quien se encarga de controlar la posición de la imagen y una vez obtenida dicha imagen, se procede a su corrección y posteriormente a su reconocimiento. Con esta información se podrá entonces clasificar cada “barra” observada y decidir sobre cual de ella se hará la medición.

1.4. Control digital

En años recientes se ha incrementado el uso de controladores digitales en sistemas de control [15, 18, 21]. Los controladores digitales se utilizan para alcanzar el desempeño óptimo – por ejemplo, en la forma de productividad máxima, beneficio máximo, costo mínimo o la utilización mínima de energía. La aplicación de control por computadora ha hecho posible el movimiento “inteligente” en robots industriales, la optimización de economía de combustible en

automóviles y el refinamiento en la operación de enceres y máquinas de uso doméstico. La capacidad en la toma de decisiones y la flexibilidad en los programas de control son las mayores ventajas de los sistemas de control digital.

La tendencia actual de controlar los sistemas dinámicos en forma digital en lugar de analógica, se debe principalmente a la disponibilidad de computadoras digitales de bajo costo y a las ventajas de trabajar con señales digitales en lugar de señales en tiempo continuo.

1.4.1. Fundamentos de control digital

Muchos fenómenos de la naturaleza pueden ser modelados por medio de distintos tipos de señales dependiendo su origen. Una de las características de estas señales es que se definen o son válidas en un intervalo continuo de tiempo, esto es, son señales continuas. La amplitud puede tener un intervalo continuo de valores o solamente un número finito de valores distintos. El proceso de representar una variable por medio de un conjunto de valores distintos se denomina cuantificación. La variable cuantificada solo cambia en un conjunto finito de valores distintos.

En este dominio se encuentran las señales analógicas donde se definen en un intervalo continuo de tiempo cuya amplitud puede adoptar un intervalo continuo de valores.

Por otro lado se tienen las señales en tiempo discreto, las cuales solo son definidas en ciertos valores del tiempo, ver figura 1.7. En una señal en tiempo discreto, si la amplitud puede adoptar valores en un intervalo continuo, entonces la señal se denomina señal de datos muestreados.

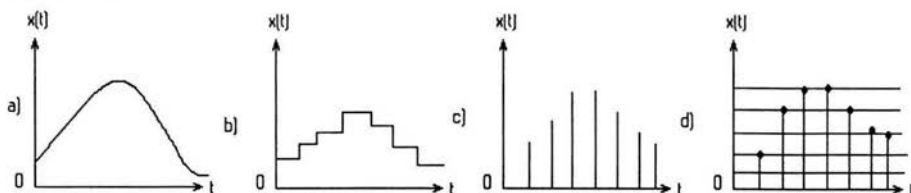


Figura 1.7. Tipos de señales: a) Señal analógica en tiempo continuo, b) Señal cuantificada en tiempo discreto, c) Señal de datos muestreados, d) Señal digital

En este dominio se encuentran las señales digitales, dichas señales se pueden representar mediante una secuencia de números binarios. En

esta señales la cuantificación se da tanto en la amplitud como en el tiempo.

Los sistemas de control en tiempo discreto son aquellos sistemas en los cuales un o más variables pueden cambiar solo en valores discretos de tiempo. Estos instantes se les denomina mediante kT donde $k = (0, 1, 2, \dots)$. En general los sistemas de control digital se pueden representar por el diagrama de bloques mostrado en la figura 1.8.

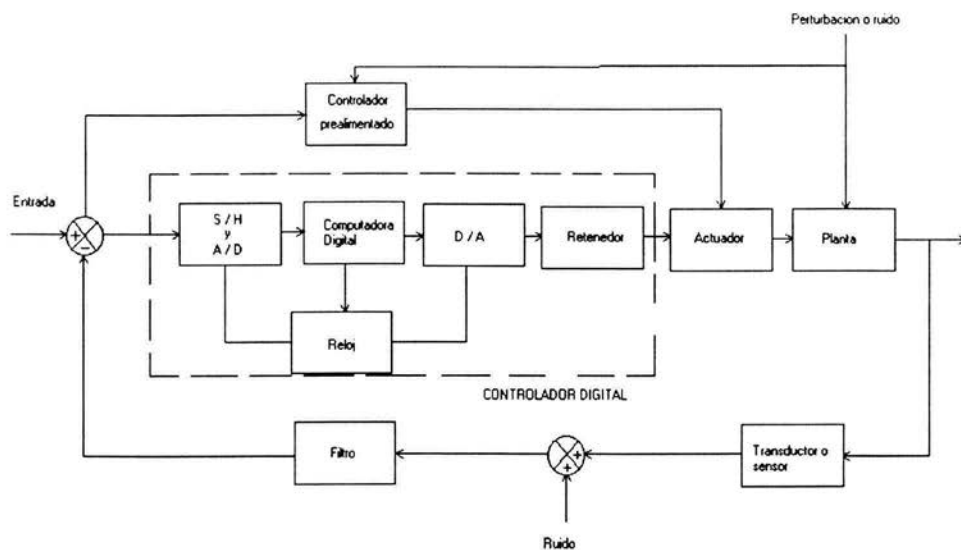


Figura 1.8. Sistema de control digital.

La operación que transforma las señales en tiempo continuo en datos en tiempo discreto se denomina *muestreo* o *discretización*. La operación inversa, que transforma datos de tiempo discreto en una señal de tiempo continuo, se le llama *retención de datos*, esta realiza la reconstrucción de la señal en tiempo continuo a partir de la secuencia de datos en tiempo discreto. El circuito de muestreo y retención y el convertidor analógico-digital (A/D) convierte la señal en tiempo continuo en una secuencia de palabras binarias codificadas numéricamente. A este proceso se le conoce como *codificación*. Al proceso inverso (D/A) se le denomina *decodificación*.

La *planta o proceso*, se le denomina a cualquier objeto físico a ser controlado. En nuestro caso será el sistema de posicionamiento por medio de motores. El *Transductor o sensor* es un dispositivo que convierte una señal de entrada en una señal de salida de naturaleza diferente a la primera, por ejemplo la cámara multimedia utilizada en

nuestro caso, convierte imagen cotidiana 3D a imágenes en la computadora 2D.

1.4.2. Controlador PID utilizado

El sistema de lazo cerrado utilizado en el sistema de medición de reglas graduadas, ha sido implementado y desarrollado en el Laboratorio de Metrología del CCADET con excelentes resultados en máquinas de medición. Este sistema se muestra en la figura 1.9.

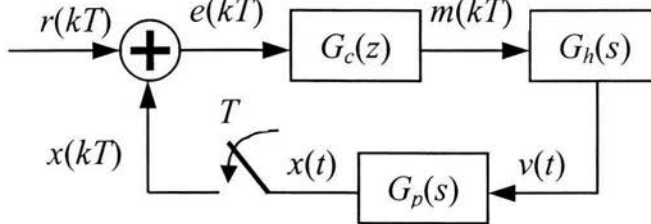


Figura 1.9. Sistema discreto de lazo cerrado para el control de posición.

donde:

$r(kT)$ = señal de referencia discreta o “setpoint” especificado por el operador.

$e(kT)$ = señal discreta de error.

$m(kT)$ = señal de alimentación de voltaje discreta.

$v(t)$ = señal de alimentación de voltaje continua.

$G_c(z)$ = función de transferencia discreta del controlador digital.

$G_h(s)$ = función de transferencia continua del retenedor de orden cero que convierte la entrada discreta en salida continua en el tiempo.

$G_p(s)$ = función de transferencia continua de la planta.

Con estos elementos es posible diseñar un controlador digital que proporcione elementos para el control de posición en la planta introduciendo el menor error posible. Puede verse que este sistema tiene componentes tanto discretos como continuos en el tiempo. Para nuestro caso es necesario tener todos los elementos en forma discreta, para ello se puede tener un sistema como el mostrado en la figura 1.10, que es equivalente al anterior pero completamente discreto.

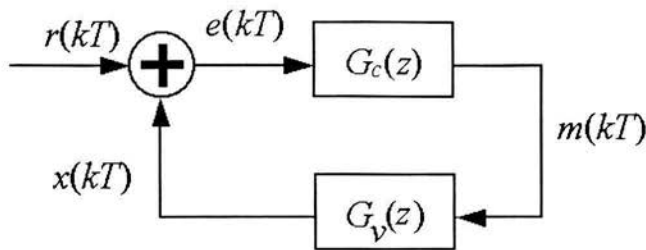


Figura 1.10. Sistema discreto de lazo cerrado para el control de posición.

Donde $G(z)$ representa la respuesta conjunta tanto del retenedor como de la planta en el tiempo discreto.

El principio básico del esquema de un control PID es que actúa sobre la variable a ser manipulada a través de una combinación apropiada de las tres acciones de control: acción de control proporcional (donde la acción de control es proporcional al error); la acción de control integral (donde la acción de control es proporcional a la integral del error) y la acción de control derivativa (donde la acción de control es proporcional a la derivada del error). Cada una de estas acciones tiene un efecto diferente en la respuesta que se tendrá en el sistema. Por ejemplo el término proporcional reduce el tiempo de subida, esto es, el sistema llega al setpoint de una forma muy rápida. La acción integral, reduce el error de estado estacionario, esto es, se trata de estabilizar de una forma rápida. El término derivativo reduce el sobrepaso, esto es no hay un cambio muy brusco entre el setpoint y el valor máximo alcanzado por el sistema. Gráficamente lo podemos ver en la figura 1.11.

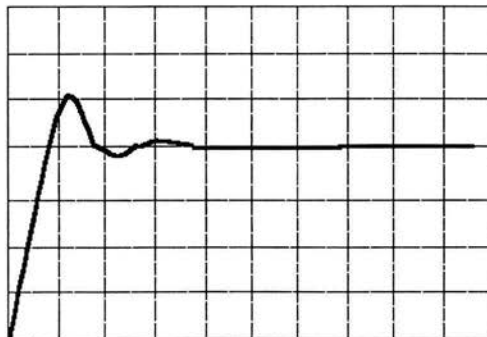


Figura 1.11. Respuesta de un sistema realimentado.

En controladores analógicos en el tiempo la acción de control PID esta dada por la siguiente expresión.

$$m(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right]$$

donde:

K es el término proporcional

T_i es el término integral

T_d es el término derivativo

Sin embargo, es necesario transformar esta expresión a un dominio discreto, esto es al dominio de Z , por lo tanto tenemos

$$G_c = \frac{M(z)}{E(z)} = Kp + \frac{Ki}{1-z^{-1}} + Kd(1-z^{-1})$$

Y desarrollando esta última expresión

$$G_c = \frac{M(z)}{E(z)} = \frac{(Kp + Ki + Kd)z^2 - (Kp + 2Kd)z + Kd}{z^2 - z}$$

en donde :

Kp = constante proporcional

Ki = constante integral

Kd = constante derivativa

Como en nuestro caso la variable a controlar es el voltaje, en la ecuación anterior representado por $M(z)$, es necesario despejar este valor y obtener esta expresión en el tiempo, por lo tanto despejando $M(z)$ esto es

$$M(z) = (Kp + Ki + Kd)E(z) - (Kp + 2Kd)E(z)z^{-1} + M(z)z^{-1}$$

Considerando que la función de transferencia z^{-n} representa un retraso de n unidades de tiempo y transformando al tiempo tenemos

$$m(k) = (Kp + Ki + Kd)e(k) - (Kp + 2Kd)e(k-1) + m(k-1) \quad (1.1)$$

La cuál será nuestra expresión a implementar para el control del motor de corriente directa. En el capítulo 5, mostramos los resultados.

Capítulo 2

Descripción de la máquina para la calibración de reglas y escalas graduadas de alta exactitud

Este capítulo contiene una descripción de la máquina para la calibración de escalas graduadas en su estado original, es decir, sin incluir los sistemas de visión y control de posición abordados en el presente trabajo de tesis. El sistema original es la base para el desarrollo de los algoritmos explicados en el capítulo 3.

Las limitaciones principales que presentaba la máquina en su estado original son las siguientes.

- Carencia de un medio automatizado para el registro de la medición sobre la escala a medir.
- Carencia de un control de lazo cerrado para el posicionamiento automático.

2.1. Introducción

En esencia, la máquina para la calibración de escalas graduadas **originalmente** era una máquina de medición en una coordenada con dos niveles de posicionamiento: uno fino que permite resoluciones de micrómetros en un alcance limitado y uno burdo que permite el posicionamiento en el alcance completo de 1.2 m. En éste sentido, el sistema electrónico permite interconectar una computadora a dos motores usados en el sistema de posicionamiento semi-automático.

Este sistema contiene un codificador óptico con una resolución de 2 μm , del cual se obtienen las lecturas que sirven de base para la calibración. Existe una tarjeta de interfase entre codificadores y PC que es la encargada de obtener dichas lecturas así como enviar los datos necesarios para el movimiento de los motores. Dicho sistema se muestra en la figura 2.1

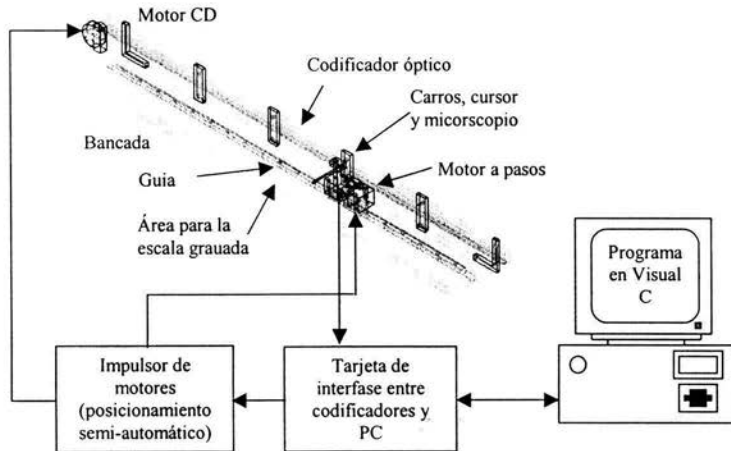


Figura 2.1. Máquina para la calibración de reglas o escalas graduadas.

El proceso que realiza el sistema es el siguiente: una computadora captura las lecturas longitudinales provenientes de un codificador óptico de posición solidario a la bancada de la máquina. Los errores de la escala graduada se obtienen por medio de la diferencia de la longitud de la escala graduada contra la lectura registrada en el codificador. La lectura en la escala graduada la realiza un operador capacitado directamente al observar a través de un microscopio de aumento. Adicionalmente el operador posiciona el conjunto de carros, cursor y microscopio al especificar el intervalo de medición deseado y al observar la posición alcanzada en el programa para PC. En éste sentido, el hardware y software proporcionan los medios para impulsar los motores eléctricos y el operador cierra el lazo de control. Inicialmente, el codificador es calibrado con un sistema de mayor exactitud y resolución (sistema de interferometría láser). El programa para PC, específico para este diseño, procesa longitudes nominales contra longitudes verdaderas.

Los componentes y tareas fundamentales del sistema electrónico y de programación son las siguientes:

- Instalación y adaptación a necesidades particulares de componentes electrónicos comerciales (tarjeta de interfase entre codificadores y PC).
- Programación en Visual C++ específica para la captura, procesamiento de mediciones y posicionamiento semi-automático.
- Tarjeta ISA (PC7266) utilizada para obtener lecturas del codificador y para impulsar los motores en el posicionamiento semi-automático.
- Transporte de resultados a programas de terceros.

A continuación se detallan cada uno de los componentes.

2.2. Sistema mecánico

Como ya se había mencionado antes, utilizamos un diseño mecánico con capacidad hasta 1200 mm de longitud útil, que además de prescindir del patrón (referencia para la calibración), garantiza la confiabilidad de la calibración de reglas o escalas graduadas de alta exactitud, inicialmente en concordancia con la recomendación OIML R35 y la norma JIS B 7516 (aplicables a reglas de uso general) y posteriormente en concordancia con la recomendación OIML R98 (aplicable a reglas de alta exactitud), este mecanismo se muestra en la figura 2.2. Tal diseño mecánico, ha sido desarrollado anteriormente en el Laboratorio de Metrología del CCADET UNAM, y constantemente se le han incorporado mejoras y adecuaciones [1].

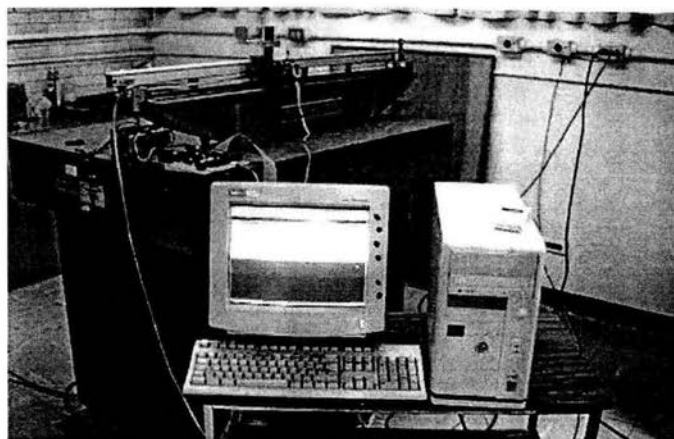


Figura 2.2. Máquina para la calibración de escalas.

El sistema mecánico cuenta con un alineamiento entre dos carros y la guía (por la cual se deslizan), para que las lecturas de

desplazamiento incluyan únicamente la componente longitudinal Y , ver figura 2.3.

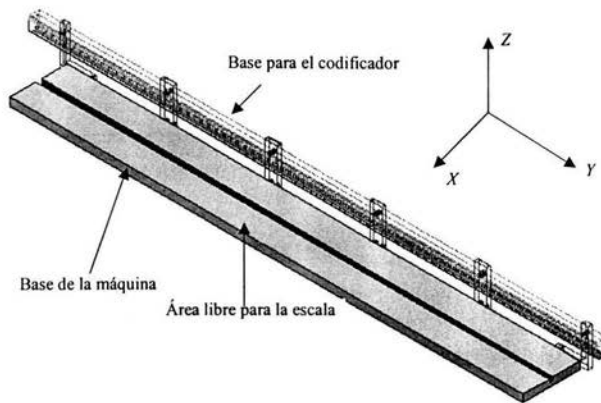


Figura 2.3. Bancada rectificada y codificador óptico.

La base de la máquina está constituida por una bancada rectificada. Por uno de sus lados, se sostiene la base para el codificador óptico. Dos carros se unen a la guía del codificador por ese mismo lado. Por el otro lado de la bancada se dispone de un área para colocar la regla o escala graduada, es decir, el área libre para la calibración.

Al centro de la bancada se encuentra la guía, por la cual se van a deslizar dos carros; uno de los carros (carro 1) está unido al cursor del codificador y el otro (carro 2) se encuentra acoplado al primero mediante un tornillo micrométrico y motor a pasos, figura 2.4. A su vez, en el carro 1 se encuentra un soporte para un pequeño microscopio mediante el cual se hará la inspección ocular para posicionar la medición actual.

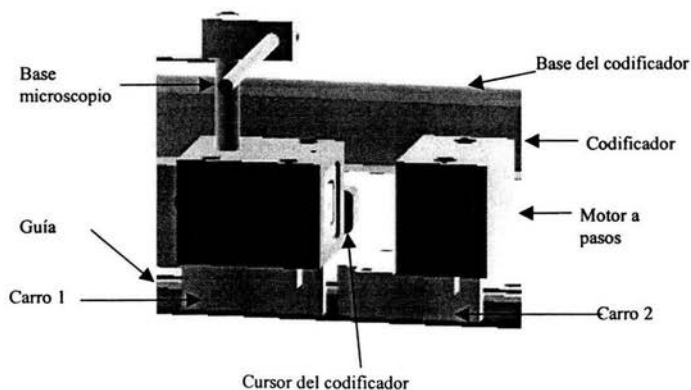


Figura 2.4. Montaje de los carros.

El desplazamiento relativo entre los carros, mediante el tornillo micrométrico, permite el nivel de desplazamiento fino. Cuando el carro 2 este fijo con respecto a la bancada, el movimiento fino se obtendrá al girar un tornillo micrométrico que desplaza el carro 1.

El mecanismo que proporciona posicionamiento burdo y le da movimiento a los carros, se constituye por una polea y un cinta metálica solidaria al carro 2, figura 2.5. El movimiento es proporcionado por un motor de DC. Por lo tanto, el frenado del motor de DC permite el posicionamiento fino descrito anteriormente. La construcción y el diseño mecánico han sido depurados, principalmente cuidando el buen alineamiento entre las piezas.

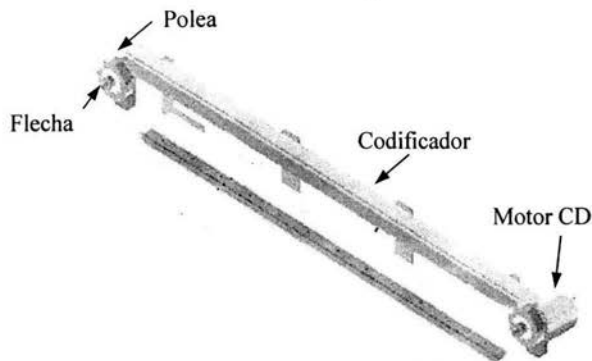


Figura 2.5. Componentes del movimiento burdo.

Adicionalmente, el diseño contempla accesorios mecánicos para la calibración periódica de la máquina mediante un sistema de mayor resolución, por ejemplo un sistema de medición láser. El sistema mecánico ya acoplado se muestra en la figura 2.6.

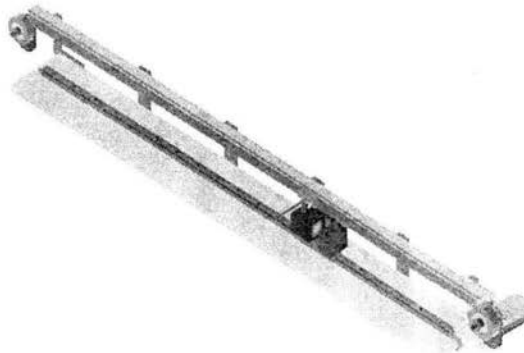


Figura 2.6. Sistema mecánico completo.

2.3. Sistema electrónico

En el desarrollo de la máquina para la calibración de reglas y escalas graduadas fue necesario contar con un sistema electrónico que permitió registrar los datos enviados del codificador óptico hacia la PC así como enviar información a los motores. El sistema electrónico que sirve como interfase entre la máquina y la computadora es una tarjeta de adquisición de datos, la cual dispone de diversos circuitos electrónicos como: contadores, circuitos de almacenamiento, circuitos de retención y multiplexores. La función que desempeña la tarjeta es básicamente el ser un medio intermedio entre el codificador, los motores y la PC, la tarjeta almacena los datos registrados al hacer las mediciones y después son enviados a la computadora para su procesamiento. La figura 2.7 muestra el diagrama a bloques del sistema electrónico.

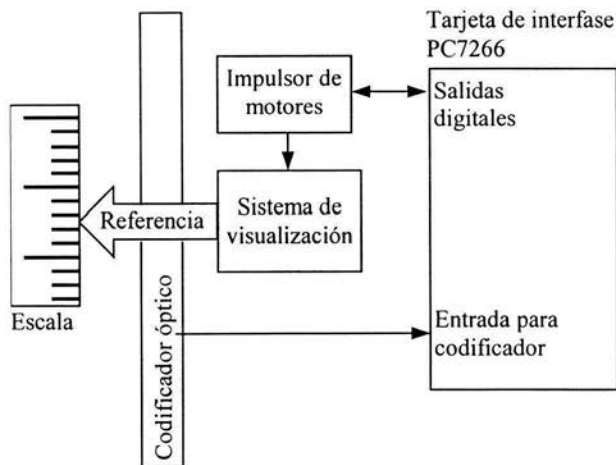


Figura 2.7. Sistema mecánico completo.

En la máquina de calibración de reglas y escalas graduadas el codificador óptico es el patrón de medición y, por lo tanto, es el encargado de proporcionar las lecturas patrón mediante el posicionamiento del carro 1 sobre la lectura deseada. Estas lecturas son capturadas por la tarjeta de interfase entre codificadores y PC. La tarjeta mantiene dichas lecturas en dispositivos electrónicos de retención para que puedan utilizarse como mejor convenga, en el caso de la máquina en su estado original se utilizan en el programa MCREG.EXE que es la interfase entre el usuario y la máquina.

En éste sentido, el sistema electrónico consta de:

- Un patrón de longitud. Es un codificador óptico lineal de la compañía ACU-RITE.
- Una tarjeta de interfase entre el codificador óptico y la PC. Esta contempla las adecuaciones necesarias para su interconexión eléctrica. Dicha tarjeta es la PC7266 con entrada ISA para la PC de la compañía USDigital Corporation. La tarjeta es capaz de capturar lecturas en forma asíncrona con respecto al procesador, de manera que no se pierdan transiciones en el codificador óptico.
- Un sistema electrónico que proporciona el medio para el posicionamiento de dos motores en dos niveles de resolución (fino y burdo), este sistema fue desarrollado en el Laboratorio de Metrología del CCADET, UNAM con el fin de obtener una placa de circuito impreso capaz de obtener la robustez necesaria para su fácil operación, confiabilidad y calidad de producto terminado. Dicha tarjeta es capaz de recibir los datos de la PC y con ellos, alimentar al motor de pasos y al de corriente directa, según se requiera.

En el nivel burdo de posición se controla el movimiento del arreglo de los dos carros mostrado en la figura 2.4 con la inercia que ello implique. Tal situación se logra mediante el control de velocidad de un motor de DC de mediana potencia. En el nivel fino de posición se mueve un tornillo micrométrico por medio de un motor de pasos en ambos sentidos, ambos controlados por es sistema descrito. A continuación una descripción detallada de los elementos que conforman el sistema electrónico.

2.3.1. Codificador lineal Acu-rite

El codificador lineal es el patrón de medición; cuenta con una longitud de 1.375m y una resolución de $2\mu\text{m}$ [2]. Cuando se mueve en cualquier dirección envía señales digitales en cuadratura que son utilizadas para codificar posición y realizar la comparación contra la regla o escala bajo inspección, figura 2.8.

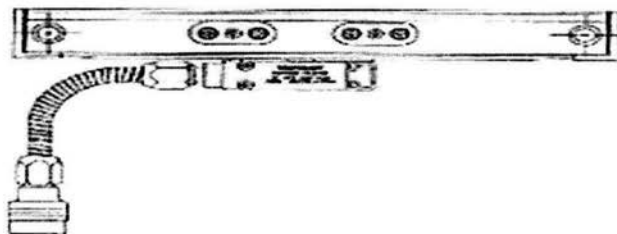


Figura 2.8. Codificador lineal ACU-RITE.

La tecnología basada en cristal del codificador óptico, es el estándar industrial y es preferido por todo el mundo para la realimentación de posición. Por casi cuatro décadas, la tecnología de cristal de los codificadores ópticos de precisión ACU-RITÉ ha probado su confiabilidad superior con exactitud y durabilidad consistente. El cristal es un material intrínsecamente estable que resiste cambios de tamaño, dimensión de una variable o la densidad debido a las variaciones en temperatura o humedad. Su funcionamiento se detalla a continuación.

El codificador óptico de precisión ACU-RITE consiste de una escala de cristal con buena calidad óptica y una pista de lectura foto electrónica. La cabeza lectora o cursor se desliza a lo largo de la escala de cristal iluminando un patrón de líneas de cromo de precisión. Este patrón es reconocido por los foto detectores, que generan una señal electrónica, ver figura 2.9.

Se puede ilustrar este proceso con dos peines de pelo: Si se tomaran dos peines y se pusiera uno encima del otro delante de una fuente de luz, y después se movieran hacia adelante y hacia atrás, se vería un modelo ondulado similar al que los receptores de luz del codificador detectan.

Sin embargo, debido a que los ojos no son tan agudos como receptores electrónicos, algunas condiciones como la lluvia o la niebla obscurecerían la visión. Los foto detectores, sin embargo, siendo mucho más sensibles, pueden detectar la luz infrarroja más intensa a través del líquido, por ejemplo líquido refrigerador, o el aceite.

Solamente con la escala de cristal de precisión ACU-RITE la tecnología puede asegurar que el codificador, en una máquina herramienta, está controlando millares de posiciones reales en cada pulgada del recorrido. Una escala de cristal de precisión $1\mu\text{m}$ ACU-RITE, tiene 1270 líneas exactamente por pulgada, cada una de las cuales es detectada por el lector fotoeléctrico mientras se desliza la escala. Esto da a los codificadores de precisión ACU-RITÉ un intervalo de medición fundamental 10 veces más fino que la tecnología alternativa más cercana del codificador, ver figura 2.10.

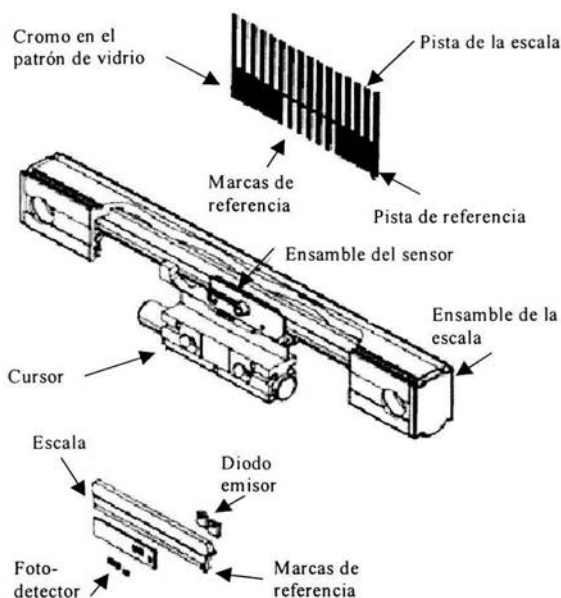
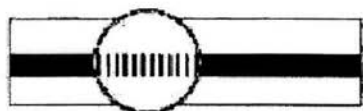


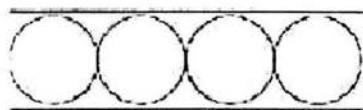
Figura 2.9. Funcionamiento de un codificador óptico.



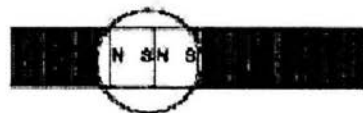
Regla estándar
16 puntos por
medición/pulgada



Codificador óptico ACU-RITE
1270 puntos por
medición/pulgada



Tecnología alterna
4 puntos por
medición/pulgada



Tecnología alterna
127 puntos por
medición/pulgada

Figura 2.10. Comparación entre mediciones longitudinales.

La capacidad de los codificadores ópticos ACU-RITE de medir continuamente posiciones exactas es debido a las excelentes calidades del cristal. El cristal es un material intrínsecamente estable que resiste cambios de tamaño, la dimensión de una variable o la densidad debido a las variaciones de la temperatura.

Esta estabilidad permite la generación de los numerosos patrones de línea ultra-finos; proveyendo a los codificadores de precisión ACU-RITE con la exactitud excepcional para las longitudes del recorrido a partir de la 1"-120" en resoluciones a partir de la 10 μ m hasta un los 0.1 μ m.

Estas características hacen que las lecturas tomadas por la máquina de calibración de escalas graduadas sean consideradas como confiables y cumplan con los estándares correspondientes.

2.3.2. Tarjeta de interfase PC7266

La tarjeta PC7266 mostrada en la figura 2.11, constituye el medio electrónico a través del cual la PC se conecta a los codificadores ópticos en la máquina, también se encarga del envío de los datos necesarios para el movimiento de los motores. El desplazamiento del eje en la máquina es registrado por los codificadores ópticos, estos son enviados a la tarjeta la cual se encarga de contar las transiciones eléctricas ocurridas en los codificadores ópticos y posteriormente son procesados por el software, el cual determinará la acción correspondiente.

De esta forma, la PC puede obtener el contenido de los contadores en la PC7266 para procesar tal cuenta e interpretarla como un desplazamiento lineal en la máquina.

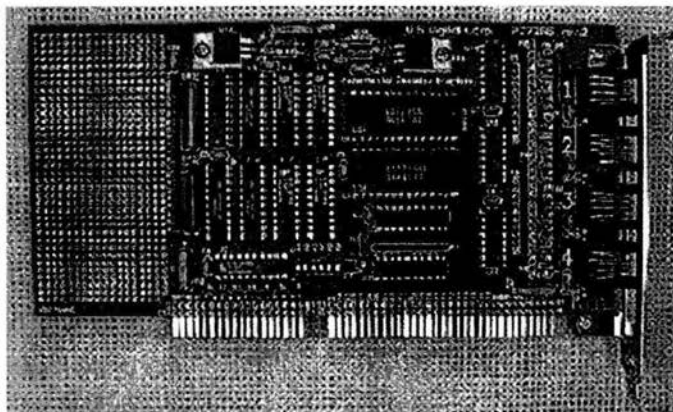


Figura 2.11. Tarjeta PC7266.

La tarjeta PC7266 se conecta en una cavidad ISA estándar de 16 bits de computadoras IBM PC y compatibles. Esta incluye dos contadores duales de 24 bits LS7266R1. La versión particular de la tarjeta acepta entradas compatibles con TTL. Cuatro PAL's reciben las entradas de "índice" y "acarreo" y controlan las señales de "carga de contador", "inicio de contador" e "interrupción". Cada uno de los cuatro canales de conexión a transductores ópticos puede programarse independientemente en uno de ocho modos diferentes. Siete "jumpers" son usados para seleccionar la dirección de interrupción IRQ. Esta tarjeta adicionalmente proporciona polarización 5.5Vcd a los codificadores externos. La tarjeta incluye capacidades de programación bajo Windows 98/00 en Visual C++ 6.0.

Originalmente, la tarjeta PC7266 ha sido probada ejecutando el programa MCREG.EXE en una PC Pentium MMX a 200MHz, 32Mb en RAM y Windows 98 [20]. No obstante, el sistema puede ser transportable a diferentes arquitecturas. La siguiente es una lista de las características más importantes de la PC7266.

- 4 canales.
- Contadores de 24 bits con capacidad de pre-carga.
- Salidas retenidas de los contadores.
- Resolución multiplicada X1, X2 y X4.
- Interfase TTL.
- Generador de interrupciones y entradas con índice programables.
- Puerto de entrada/salida de 8 bits.
- Software para desarrollar aplicaciones.
- Área de desarrollo de prototipos.

Los conectores en la tarjeta son modulares tipo telefónico de 6 pines, uno para cada canal. Al puerto de entrada/salida de 8 bits se le adaptó un conector tipo DB25, ambos se muestran en la figura 2.12.

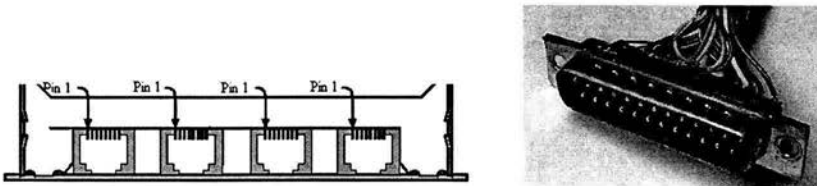


Figura 2.12. Entradas a los canales y conector DB25.


La función en cada pin del conector tipo telefónico se describe en la tabla 2.1.

Pin	Nombre	Descripción
1	GND	Tierra común de polarización y datos
2	PWR	Salida de polarización a los codificadores
3	I-	Entrada en cuadratura, I+ invertida
4	I+	Entrada en cuadratura
5	A-	Entrada en cuadratura, A+ invertida
6	A+	Entrada en cuadratura
7	B-	Entrada en cuadratura, B+ invertida
8	B+	Entrada en cuadratura

Tabla 2.1. Función de los pines.

Parte de la configuración de la tarjeta consiste en seleccionar la dirección de memoria a la cuál hará referencia para el envío y recepción de datos con la PC. La selección de la dirección base se realiza con la ayuda de un conector "dip switch", mostrado en la tabla 2.2. En nuestro caso la dirección utilizada será la 330.

HEX	A9	A8	A7	A6	A5	A4
220	1	0	0	0	1	0
240	1	0	0	1	0	0
250	1	0	0	1	0	1
260	1	0	0	1	1	0
300	1	1	0	0	0	0
310	1	1	0	0	0	1
330	1	1	0	0	1	1
340	1	1	0	1	0	0
350	1	1	0	1	0	1
360	1	1	0	1	1	0



1	A4
2	A5
3	A6
4	A7
5	A8
6	A9

Tabla 2.2. Dirección base y dip switch.

Una vez configurada y obtenida la dirección base, es posible tener completo control de las entradas y salidas de la tarjeta. Para ello, el mapa de puertos de la tabla 2.3, ayuda al programador a establecer las funciones de la PC7266 mediante la lectura/escritura en tales direcciones.

Dir. (+base)	Leer / Escribir	Descripción
0	Leer/Escribir	Dato hacia / desde LS7266R1 Codificador #1
1	Leer	Bandera de estado desde LS7266R1 Codificador #1
1	Escribir	Control al LS7266R1 Codificador #1
2	Leer/Escribir	Dato hacia / desde LS7266R1 Codificador #2
3	Leer	Bandera de estado desde LS7266R1 Codificador #2
3	Escribir	Control al LS7266R1 Codificador #2
4	Leer/Escribir	Dato hacia / desde LS7266R1 Codificador #3
5	Leer	Bandera de estado desde LS7266R1 Codificador #3
5	Escribir	Control al LS7266R1 Codificador #3
6	Leer/Escribir	Dato hacia / desde LS7266R1 Codificador #4
7	Leer	Bandera de estado desde LS7266R1 Codificador #4
7	Escribir	Control al LS7266R1 Codificador #4
8	Leer	Estado desde todas las PAL's (cuatro primeros bits)
8	Escribir	Modo buffer para las PAL's 1 y 2
9	Escribir	Modo buffer para las PAL's 3 y 4
A	Leer/Escribir	Hacia / desde el puerto entrada/salida

Tabla 2.3. Mapa de puertos.

El procedimiento para enviar y recibir datos puede ser el siguiente. Una vez conectada la tarjeta PC7266 al puerto ISA de la tarjeta madre de la PC y configurada el dip switch con la dirección base, se necesita instalar los controladores para que el sistema operativo pueda trabajar con ella. Por medio de un lenguaje de programación, en este caso Visual C++, es necesario inicializar dicha tarjeta, esto es, obteniendo por software la dirección base que se estableció previamente con el dip switch y abriendo un canal de comunicación entre ella y el programa. Esta inicialización arroja un puntero, con el cual es posible acceder a todas las funciones de la tarjeta, como recibir datos de los

codificadores o enviar datos por el puerto de entrada / salida. Esto se hace accediendo a la dirección de memoria correspondiente según la tabla 2.3. Por ejemplo, si se quiere Leer/Escribir en el puerto de entrada / salida es necesario sumar un valor Ahex a la dirección base y enviar o recibir los datos de la dirección resultante. Una vez finalizado el programa es necesario “cerrar” la tarjeta, esto es, liberar las direcciones en memoria utilizadas para evitar posteriores problemas. Para tener un control más fácil y confiable sobre la tarjeta PC7266, se utiliza un archivo tipo DLL, que fue necesario modificar, ya que algunas funciones que se necesitaron para el sistema, no estaban implementadas. La programación de la tarjeta así como de los demás dispositivos se abordan ampliamente en el capítulo 4 de la presente investigación.

2.3.3. Impulsor de motores

Como ya se había mencionado, el Laboratorio de Metrología del CCADET, desarrolló una tarjeta electrónica que fuera capaz de mediar la comunicación entre el puerto de 8 bits de la tarjeta PC7266 y los motores de pasos y el de DC utilizados en la máquina de calibración [22]. Dicho sistema se muestra en la figura 2.13.

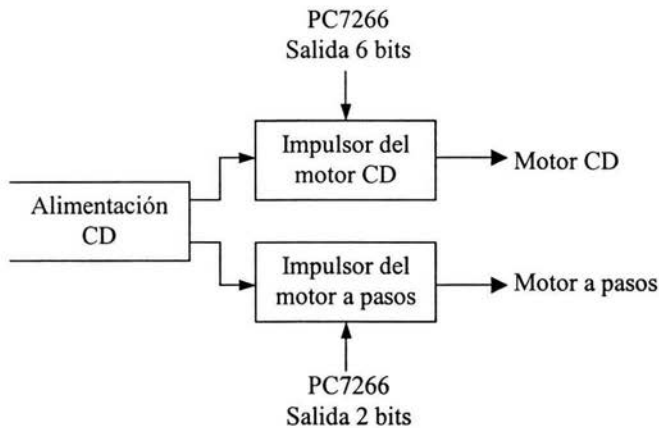


Figura 2.13. Impulsor de motores.

Dicha placa es capaz de recibir información del puerto de 8bits de la tarjeta PC7266 y con ello controlar la velocidad del motor de CD por modulación de ancho de pulso (PWM) y mover el motor de pasos en ambos sentidos. A groso modo puede dividirse en la siguientes tres etapas.

Sistema de alimentación

El sistema de alimentación consta de dos transformadores ambos a 30V pero uno de 1 A y otro de 3 A. Estos transformadores alimentan a los transistores, diodos, capacitores, circuitos integrados y demás componentes de la tarjeta así como a los motores. Al final de este subsistema podemos ver que se tienen 4 fuentes de $\pm 12V$ con los amperajes mencionados, ver figura 2.14.

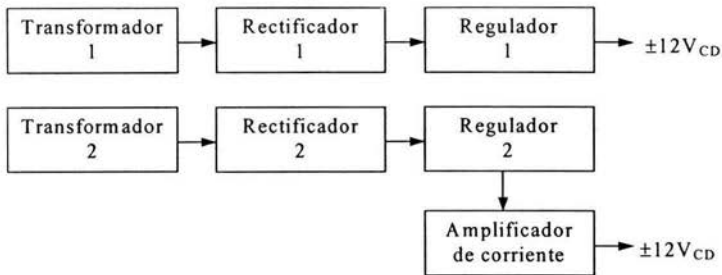


Figura 2.14. Sistema de alimentación.

Impulsor del motor de CD

Para el control del motor de DC, es necesario codificar los datos enviados por la tarjeta PC7266, sin embargo, de los 8 bits enviados por el puerto de la tarjeta, se utilizan solo los 6 últimos bits, estos son enviados a un DAC para su conversión analógica y este a su vez se compara con una señal triangular para obtener la modulación PWM que es enviada al motor de DC, la cual consiste en generar una señal cuadrada de período constante, pero de ciclo de trabajo variable. Con ello la velocidad del motor de DC está determinada por los 6 últimos bits del puerto de la tarjeta PC7266, ver figura 2.15.

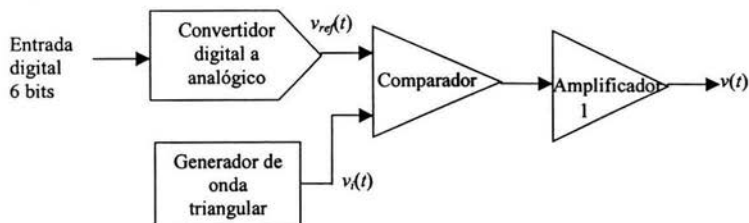


Figura 2.15. Control de motor CD.

Impulsor del motor de pasos

Para el control de posición del motor de pasos se utilizan los primeros dos bits del puerto de entrada / salida de la tarjeta PC7266, estos son los encargados de mover el motor hacia un lado o hacia otro según se requiera, figura 2.16.

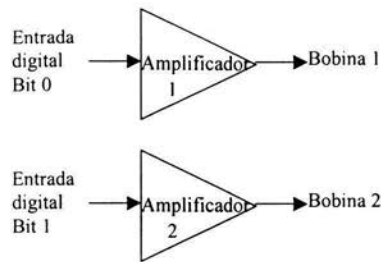


Figura 2.16. Control para el motor de pasos.

2.4. Sistema de visualización

El sistema de visualización de la máquina para calibración de reglas consta de una lente de microscopio de la empresa Edmund Industrial Optics, con la cual pueden amplificarse de una forma nítida las marcas en las reglas, con ello es posible hacer mediciones a instrumentos con resoluciones por debajo de los 0.5 mm. La lente tiene una escala propia cuya longitud es de -2.5 a 2.5 mm con una resolución de 0.05 mm.

Esta lente es montada en el carro 1 de la máquina como se puede ver en la figura 2.17. El acoplamiento entre el carro y la lente es por medio de un soporte que permite tener un movimiento en sentido vertical, transversal y radial (suponiendo un eje z de forma vertical), esto con la finalidad de colocar la lente en la posición más adecuada para la medición de las reglas.

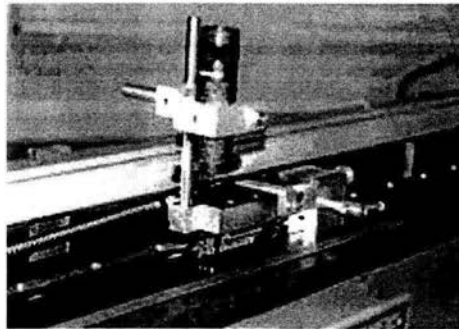


Figura 2.17. Sistema de visualización.

Ello es importante ya que el grosor de las escalas puede variar, se consideran casos en donde el simple enfoque de la lente no sea suficiente para realizar una correcta medición.

La lente es uno de los principales componentes del sistema ya que para hacer una medición es necesario observar a través de ella y comparar la regla a calibrar con la escala graduada propia de la lente, la cuál es utilizada como referencia para obtener el error.

En el sistema propuesto no se sustituye esta lente, sólo se le agrega un dispositivo electrónico que supla al ojo humano, esto es, una cámara digital, la cual será parte indispensable para el sistema de visión y automatización del proceso.

Capítulo 3

Descripción del sistema propuesto para la medición

La máquina para la calibración de escalas graduadas, en su forma original, era capaz de realizar mediciones con buena exactitud pero con una intervención importante por parte del operador. A la máquina original, descrita en el capítulo 2, se le añadieron algoritmos de visión por computadora y control digital para eliminar la intervención humana e incrementar la exactitud. El capítulo 3 describe los algoritmos desarrollados que constituyen la parte medular del presente proyecto de tesis.

3.1. Introducción

Como se vio en el capítulo anterior, se tenía todo un sistema para la calibración de reglas y escalas graduadas que constaba principalmente de un sistema mecánico y un sistema electrónico conectados a una computadora. El sistema de visualización utilizado requería de intervención humana y la exactitud de la lectura dependía del buen juicio del operador. El sistema de visualización era colocado en el intervalo de medición deseado por medio del teclado de la computadora a través del programa MCREG.exe[22]. En estas condiciones, el lazo de control era cerrado por el operador. La labor de posicionar el sistema de visualización demandaba una gran habilidad del operador. Una vez colocado el lente en la posición requerida el operador tomaba las lecturas tanto del patrón de longitud como de la escala a calibrar y las ingresaba al software MCREG en forma automática.

Para realizar todo ello el operador tenía que mantener la vista fija a través de la lente del microscopio durante todo el proceso, esto es, si una escala se calibraba cada centímetro y el tamaño total era de 1 metro, el operador tenía que hacer obtener 100 lecturas y registrarlas en la computadora. Además el procedimiento menciona que se obtengan como mínimo tres series, es decir, el operador tenía que hacer 3 series de 100 lecturas cada una, un total de 300 lecturas para calibrar una regla.

Este sistema impráctico pero válido según las normas establecidas, requería de un tiempo considerablemente grande; aparte, para el operador resultaba ser agotador y tedioso. Además el buen posicionamiento del cursor, dependía principalmente de dos factores: del dispositivo electromecánico y del juicio del operador que esta controlando el proceso a través del sistema de observación.

Debemos agregar que todo sistema de visión tiene en diferente grado cierta distorsión, lo cuál implica otro problema a la hora de obtener las mediciones.

Por ello en el sistema desarrollado se tomaron en consideración los siguientes criterios de diseño:

- Automatización del proceso, reducción al mínimo de la intervención por parte del operador.
- Mínima inversión en el tiempo requerido para la calibración de las escalas y reglas graduadas.
- Corrección de distorsiones en la lente de la cámara.

Para lograrlo se plantearon una serie de soluciones, entre las cuales podemos citar lo siguiente. Para la corrección de imágenes, se utilizó un modelo de calibración de cámara, el cual consiste en utilizar una malla o rejilla patrón y a partir de esta obtener una imagen sin distorsión, esto es, obtener una imagen en donde “las líneas rectas en la escena permanezcan rectas en la imagen”. Lo anterior es necesario ya que al reconocer las líneas sobre las escalas, la distorsión podría provocar información falsa como líneas de diferentes alturas o diferentes grosores. Esto tiene como consecuencia un considerable error en el momento de obtener las diferencias entre la medida patrón y la de la escala a calibrar. Para la automatización del proceso, se desarrolló un software con las siguientes características.

- Es capaz de obtener imágenes de una cámara digital y corregir la distorsión producida por el acoplamiento de las lentes del microscopio y de la cámara que reemplaza la visión humana, como se describe en 3.1.

- Con las imágenes ya corregidas, compensa la iluminación y reconoce los diferentes patrones de las barras que se presentan en las imágenes. Una vez obtenidos estos patrones, se procede a tomar la medición de la barra seleccionada y capturar los datos correspondientes, como se describe en 3.2.
- Posiciona el sistema de visualización automáticamente en el intervalo de medición deseado por medio de control digital automático en lazo cerrado, como se describe en 3.3.

El tiempo en que realiza este proceso, dependerá del número de lecturas que se hagan, sin embargo en cualquier caso es considerablemente menor que el sistema anterior, ver capítulo 5. El sistema propuesto se ilustra gráficamente en la figura 3.1.

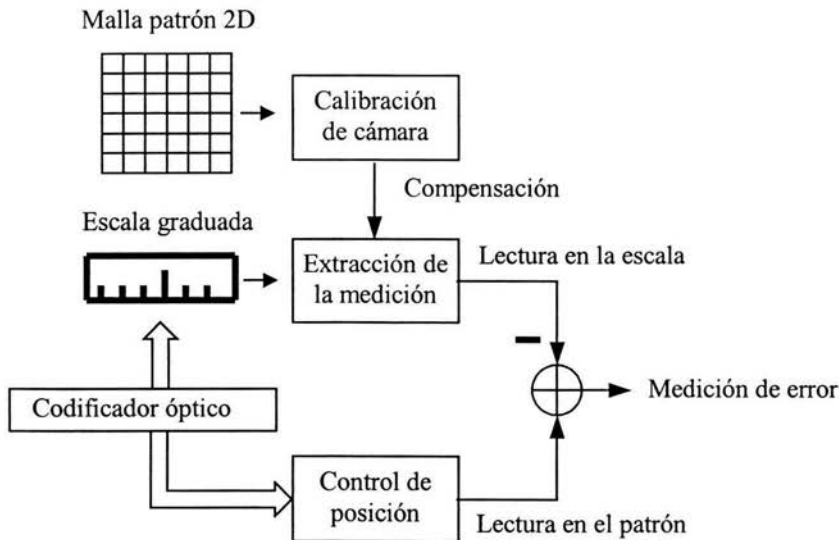


Figura 3.1. Diagrama para la medición por visión.

A continuación se detalla cada uno de los elementos del sistema vistos anteriormente.

3.2. Calibración de la cámara

La idea principal en la calibración de la cámara es derivar un conjunto de características sin distorsión a partir de aquellas con distorsión y las dimensiones de la malla patrón. Gráficamente se puede ver en la figura 3.2.

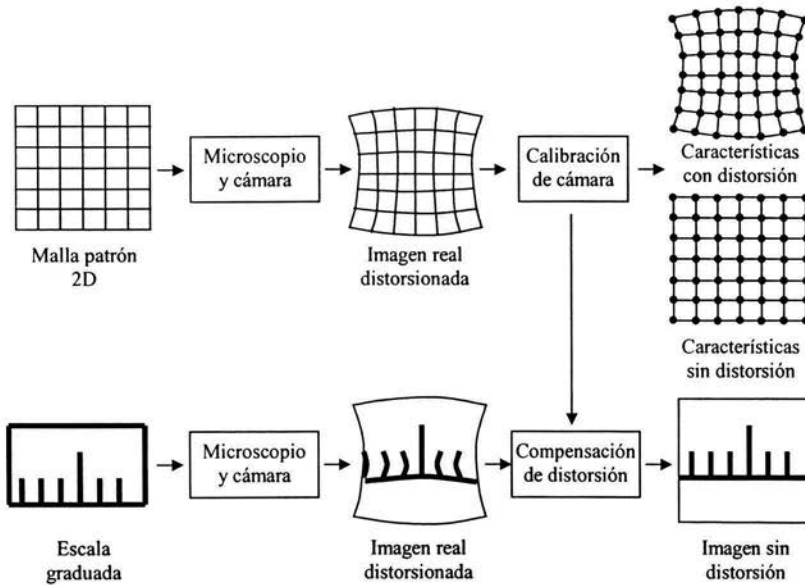


Figura 3.2. Método para la corrección de imágenes.

En el procedimiento de calibración de cámara, utilizamos un enfoque similar al modelo de cámara de orificio [3, 19]. En nuestro modelo existen dos diferencias principales con respecto a los modelos utilizados en visión por computadora. El primero consiste en que la escena es ampliada y no invertida como en el modelo de cámara de orificio [5]. En el segundo, el corto campo de visión del sistema de adquisición impone escenas planas con prácticamente cero profundidad. Entonces se utiliza un patrón de calibración 2D mientras la profundidad permanece como un parámetro libre. El modelo de cámara adoptado en nuestro trabajo se muestra en la figura 3.3.

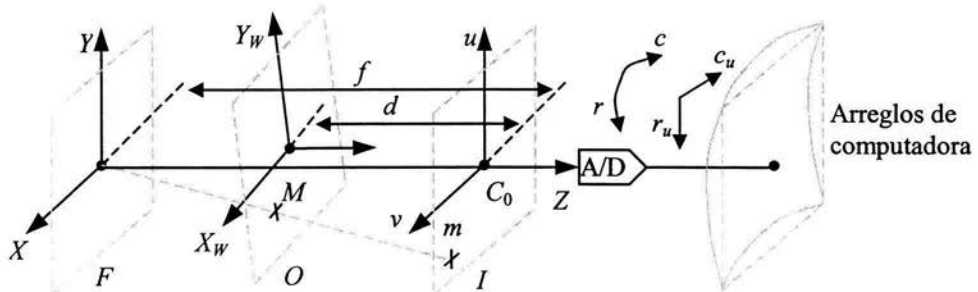


Figura 3.3. Modelo de cámara.

En esta figura, el patrón de calibración (malla) es colocado en el plano objetivo, O , y su proyección se forma en el plano imagen, I . El modelo se completa con el plano focal, F , el cual contiene el punto focal. Entonces, una característica en el patrón de calibración, M , se proyecta en el plano imagen, m . El plano imagen tiene dos componentes: con y sin distorsión, ambas relacionadas por un modelo de distorsión. Con el objetivo de realizar tareas de procesamiento de imágenes, las coordenadas de la imagen son almacenadas en arreglos organizados en renglones-columnas. Por lo tanto, el sistema de visión produce un arreglo con distorsión. El alineamiento paralelo en profundidad (eje Z) entre los planos O e I se alcanza al enfocar por medios mecánicos. Para tal fin, desarrollamos una base mecánica que acopla la cámara al microscopio y que posee diferentes grados de libertad para el enfoque exacto. La base mecánica se muestra en la figura 3.4.

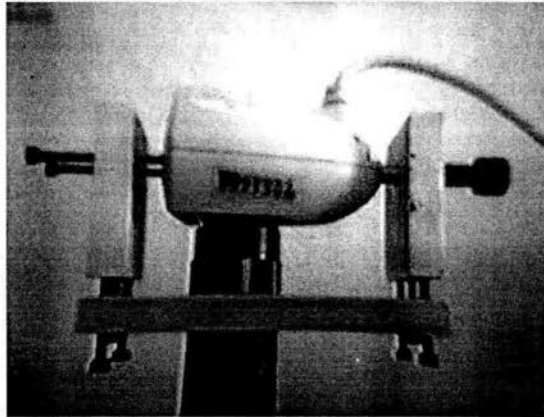


Figura 3.4. Base para acoplar la cámara al microscopio.

El método se divide en tres etapas: extracción de características, estimación de la distorsión y compensación de la distorsión.

3.2.1. Extracción de características

En general la extracción de características se refiere a obtener información de la imagen de la rejilla obtenida con la cámara digital. Para ello utilizamos una malla patrón de 0.5×0.5 mm, como la que se muestra en la figura 3.5.

El objetivo de esta etapa, es obtener los “puntos de los cruces”, y con ellos realizar una correlación entre estos puntos (2D) y puntos globales (3D). Esto es posible, ya que sabemos que la distancia real entre un punto y otro que es de 0.5 mm y al obtener los puntos de los cruces, sabremos que distancia en pixeles existe entre uno y otro. En

un caso ideal donde la imagen tuviera una distorsión nula, la distancia en pixeles debería ser constante entre uno y otro, la cual tendría una correlación directa con los 0.5 mm ya conocidos, sin embargo como toda lente provoca cierta distorsión, esta distancia entre puntos no es constante.

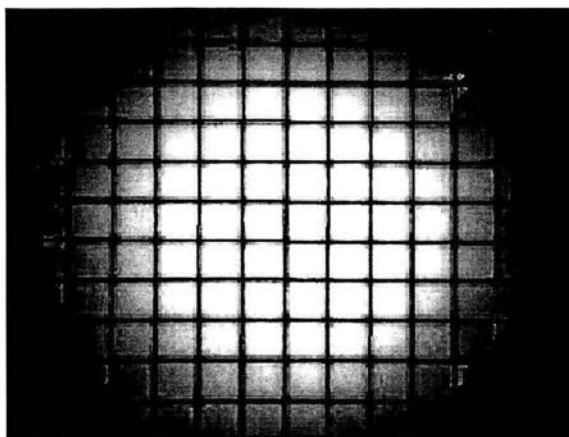


Figura 3.5. Imagen original de la malla patrón.

Esta diferencia es aprovechada para conocer que tan distorsionada está una imagen y con ello proponer un modelo que trate de minimizar esta diferencia, esto es, que elimine la distorsión de la imagen.

Para ello se realizó un software que nos permitiera especificar los cruces de la malla de una manera aproximada para posteriormente refinarlos con mayor exactitud. Los cruces de la malla son ingresados inicialmente por el usuario en una interfase amigable y de fácil uso al presionar el botón izquierdo del ratón, ver figura 3.6.

Los puntos de cruce entre líneas horizontales y verticales son extraídos con exactitud sub-píxel a través de un método basado en un punto de inicio y análisis de un patrón cuadrado alrededor de él [11]. Este método evita el procedimiento de minimización iterativa descrito en [5] y constituye una solución cerrada al problema que contribuye a la rapidez de cálculo sin pérdida de exactitud. La idea principal del algoritmo es ajustar líneas a los datos de la imagen y encontrar el punto de intersección. Se necesitan dos líneas para extraer la característica con exactitud sub-píxel: una horizontal y otra vertical. Las líneas son definidas al analizar la imagen original en escala de gris a lo largo de un patrón cuadrado centrado en los puntos iniciales especificados en forma aproximada por el usuario.

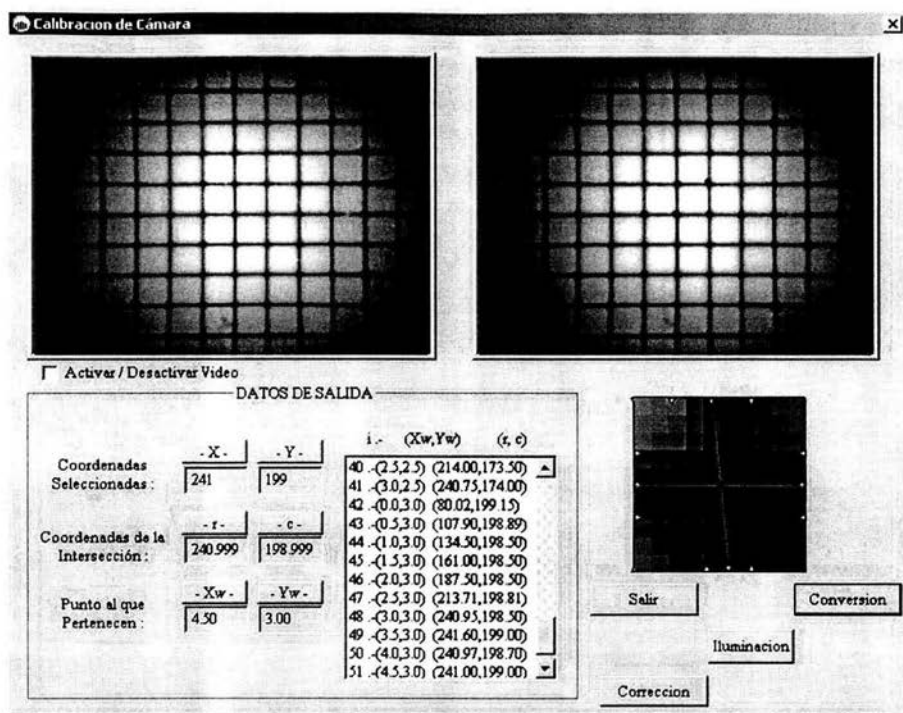


Figura 3.6. Interfase de usuario en la extracción de características.

La figura 3.7 muestra el proceso. El perfil de intensidades del patrón define ocho puntos borde marcados con triángulos en la figura 3.7.a. Los puntos borde son ordenados en pares horizontales y verticales para localizar sus puntos medios. Los puntos medios, denotados con círculos, definen dos líneas. Se utiliza matemática simple para localizar la intersección de estas dos líneas con exactitud sub-píxel. La intersección entonces define la característica final marcada con un punto oscuro. La figura 3.7.b muestra un perfil de intensidades típico de un patrón cuadrado. La derivada de primer orden se calcula de éste perfil. Después, los máximos de la derivada son detectados y agrupados en pares para conformar los puntos borde. Ya que la imagen original está definida en el dominio discreto, el perfil de intensidades del patrón es calculado utilizando interpolación bilineal.

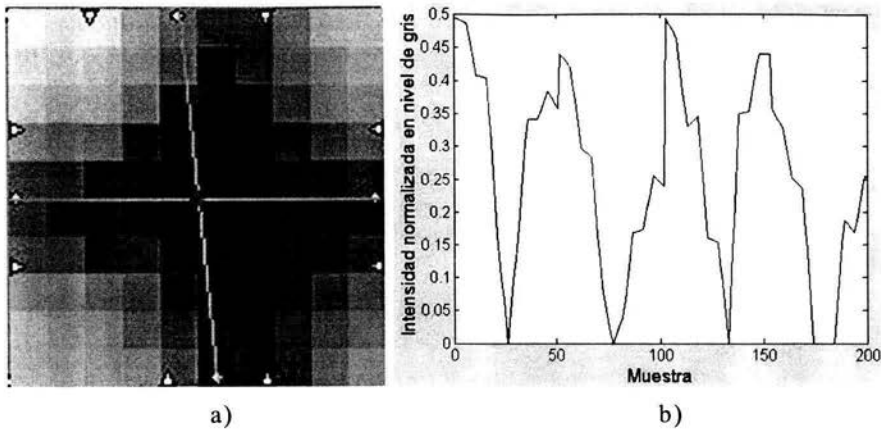


Figura 3.7. Extracción de características con exactitud subpíxel. a) Patrón cuadrado y características. b) Perfil de intensidades del patrón cuadrado.

El proceso para obtener las características con exactitud subpíxel de la malla, incluye también, asignar su coordenada global correspondiente, esto implica que se deberá seguir un orden en la selección de puntos de la imagen. En la extracción se deberá empezar por el primer cruce visible de la parte superior izquierda de la imagen, este valor se tomará como el origen global. Deberá continuarse hacia la derecha y continuar en el renglón de abajo empezando por el primer cruce visible a la izquierda. Esto se repite sucesivamente hasta llegar a la última línea visible de la imagen. Con este procedimiento, se tendrán N características con distorsión, (r_i, c_i) donde: $i = 1, \dots, N$ y N puntos correlacionados 3D, (X_{wi}, Y_{wi}) . La coordenada Z es considerada como 0.

3.2.2. Estimación de la distorsión

En la estimación de la distorsión se genera un conjunto de características sin distorsión, (r_{ui}, c_{ui}) , a partir de las características con distorsión, (r_i, c_i) , su correlación con la posición en la malla, (X_{wi}, Y_{wi}) , y el modelo para la calibración de la cámara. El modelo de calibración de cámara consiste en encontrar una relación entre posiciones globales y el arreglo de computadora sin distorsión [25]. Entonces la estimación de la distorsión relaciona los arreglos con y sin distorsión. La calibración de la cámara se desarrolla en cuatro etapas: transformación de coordenadas globales a coordenadas de cámara, transformación de coordenadas de cámara a coordenadas de imagen, transformación de coordenadas de imagen a coordenadas del arreglo sin distorsión y el modelo de proyección. La última etapa concluye la

relación entre coordenadas globales y su proyección en el arreglo sin distorsión.

Transformación de coordenadas globales a coordenadas de cámara

La geometría de la transformación rota y traslada el sistema (X_w, Y_w, Z_w) hacia el sistema (X, Y, Z) , de la figura 3.3, la rotación solamente tiene componente en Z , θ , mientras que la traslación en Z es $f-d$. Analíticamente

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & t_x \\ \sin\theta & \cos\theta & 0 & t_y \\ 0 & 0 & 1 & f-d \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3.1)$$

Transformación de coordenadas de cámara a coordenadas de imagen

Como se muestra en la figura 3.3, la proyección perspectiva de la cámara de orificio transforma (X, Y, Z) hacia (u, v) , o de otra forma

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{Z}{f} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.2)$$

Transformación de coordenadas imagen a coordenadas de arreglo sin distorsión

La posición sin distorsión del conjunto de píxeles en la imagen digital está relacionada con el plano imagen mediante la siguiente expresión

$$u = \frac{c_u - c_0}{s_u} \quad v = \frac{r_u - r_0}{s_v} \quad (3.3)$$

en donde (r_0, c_0) denota la posición del píxel que corresponde al origen C_0 . Como se puede notar, s_u y s_v son constantes de proporción que introducen un factor de aspecto. En general, los ccd's comunes se construyen simétricamente de tal forma que $s_u = s_v$.

Modelo de proyección

En ésta etapa, el trabajo previo se sintetiza en una ecuación que es utilizada para obtener las características sin distorsión a partir de las coordenadas globales conocidas. Es sencillo demostrar que combinando las ecuaciones (3.1) a (3.3) resulta en

$$\begin{bmatrix} c_u \\ r_u \end{bmatrix} = \begin{bmatrix} a_u & -a_v & t_u \\ a_v & a_u & t_v \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \quad (3.4)$$

en donde:

$$s = \frac{fs_u}{Z_w + f - d} = \frac{fs_v}{Z_w + f - d}$$

$$t_u = st_x + c_0 \quad t_v = st_y + r_0$$

$$a_u = s \cos \theta \quad a_v = s \sin \theta$$

La ecuación (3.4) debe ajustar una serie de N observaciones (c_i, r_i) , (X_{wi}, Y_{wi}) $i = 1, \dots, N$ al modelo de la ecuación con parámetros ajustables $\rho = (a_u, a_v, t_u, t_v)$. Por lo tanto, ρ debe minimizar el siguiente error

$$\gamma(\rho) = \frac{1}{N} \sum_{i=1}^N [(c_i - c_{ui})^2 + (r_i - r_{ui})^2] \quad (3.5)$$

Sustituyendo la ecuación (3.4) en (3.5) y tomando en cuenta que en el mínimo las derivadas parciales de γ con respecto a ρ son cero, se obtiene el siguiente sistema de cuatro ecuaciones

$$\begin{bmatrix} C & 0 & A & B \\ 0 & C & -B & A \\ A & -B & 1 & 0 \\ B & A & 0 & 1 \end{bmatrix} \begin{bmatrix} a_u \\ a_v \\ t_u \\ t_v \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ C_0 \\ R_c \end{bmatrix} \quad (3.6)$$

La ecuación (3.6) puede ser resuelta para $\rho = (a_u, a_v, t_u, t_v)$ tomando como datos de entrada las posiciones globales (X_{wi}, Y_{wi}) correlacionadas con sus proyecciones en el plano imagen (c_i, r_i) , ya que

$$A = \frac{1}{N} \sum_{i=1}^N X_{wi} \quad B = \frac{1}{N} \sum_{i=1}^N Y_{wi} \quad C = \frac{1}{N} \sum_{i=1}^N (X_{wi}^2 + Y_{wi}^2)$$

$$R_c = \frac{1}{N} \sum_{i=1}^N r_i \quad C_0 = \frac{1}{N} \sum_{i=1}^N c_i \quad (3.7)$$

$$D_1 = \frac{1}{N} \sum_{i=1}^N (c_i X_{wi} + r_i Y_{wi}) \quad D_2 = \frac{1}{N} \sum_{i=1}^N (r_i X_{wi} - c_i Y_{wi})$$

Entonces, la ecuación (3.4) se puede usar para generar las características sin distorsión usando las características con distorsión y sus posiciones globales, como se muestra en la figura 3.8.

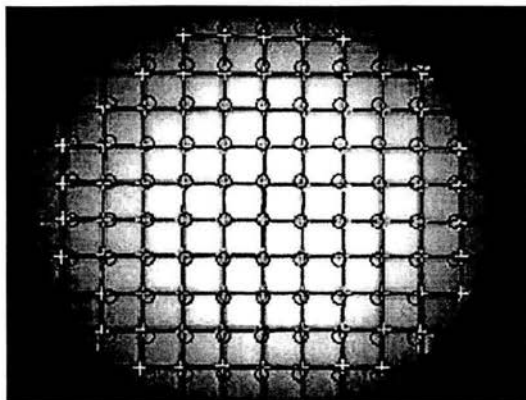


Figura 3.8. Puntos con distorsión y puntos sin distorsión.

En la figura 3.8 se destacan las características con distorsión, marcadas con un signo más, y las características sin distorsión, marcadas con un círculo. De ésta forma se puede apreciar la necesidad de compensar la distorsión en las imágenes del sistema de tal forma que las marcas con el signo más sean desplazadas hacia las marcas con el signo del círculo. De ello se hablará a continuación.

3.2.3. Compensación de la distorsión

Hasta el momento conocemos los puntos originales de la imagen y los puntos sin distorsión. Estos últimos corresponden a los puntos que mejor describen a la escena. Ahora tenemos los elementos para aplicar un algoritmo que sea capaz de corregir estos puntos, esto es, que “mueva” los puntos con distorsión a los puntos sin distorsión y con ello obtener una imagen sin errores de lente.

Para ello se propuso aplicar el algoritmo de corrección basado en regiones [26]. Este algoritmo requiere de dos arreglos de características en forma de enrejados: con distorsión (arreglo origen) y sin distorsión (arreglo destino) mismos que ya se obtuvieron en el tema 3.2.1. El arreglo origen es trasladado al destino utilizando una transformación bilineal para cada región del enrejado.

Como primer paso, y en forma paralela a la extracción de características descrita en 3.2.1, se extraen las regiones cuadrilaterales, es decir, aquellas regiones que están delimitadas por cuatro características o cruces de línea. Aquellas que no lo están, son descartadas. Entonces, cada región cuadrilateral distorsionada es

transformada en una región cuadrilateral sin distorsión, como se muestra en la figura 3.9.

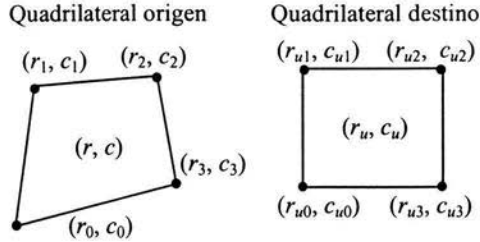


Figura 3.9. Puntos con distorsión y puntos sin distorsión.

La transformación tiene la siguiente forma:

$$r = a_0 + a_1 r_u + a_2 c_u + a_3 r_u c_u \quad c = b_0 + b_1 r_u + b_2 c_u + b_3 r_u c_u \quad (3.8)$$

en donde los parámetros a y b están definidos por el conocimiento de los ocho puntos vértice de ambos cuadriláteros de la siguiente forma

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} 1 & r_{u0} & c_{u0} & r_{u0}c_{u0} \\ 1 & r_{u1} & c_{u1} & r_{u1}c_{u1} \\ 1 & r_{u2} & c_{u2} & r_{u2}c_{u2} \\ 1 & r_{u3} & c_{u3} & r_{u3}c_{u3} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 & r_{u0} & c_{u0} & r_{u0}c_{u0} \\ 1 & r_{u1} & c_{u1} & r_{u1}c_{u1} \\ 1 & r_{u2} & c_{u2} & r_{u2}c_{u2} \\ 1 & r_{u3} & c_{u3} & r_{u3}c_{u3} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Aplicando la transformación inversa definida en la ecuación (3.8) a cada cuadrilátero extraído, se corrige la distorsión. La imagen original distorsionada de la malla patrón se muestra en la figura 3.10 a). La corrección aplicada se muestra en la figura 3.10 b). En la técnica empleada, las regiones no cuadrilaterales son llenadas con negro.

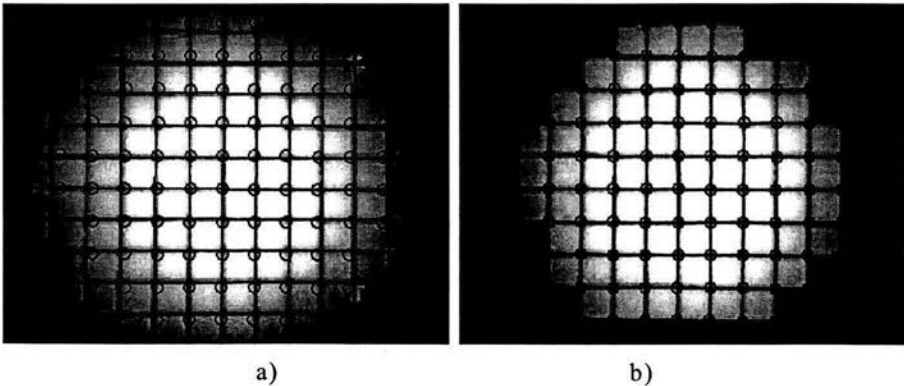


Figura 3.10. Compensación de la distorsión. a) Imagen de la rejilla. b) Imagen corregida.

La idea de obtener la compensación, es aplicarla a las imágenes obtenidas por la cámara para que se tengan trazos rectos y sin distorsión de las líneas de las escalas y reglas graduadas. Mientras la estimación de la distorsión se aplica una sola vez al inicio del proceso, la compensación se aplica en lo consecutivo a cualquier imagen. Las imágenes corregidas se muestran en la Figura 3.11.

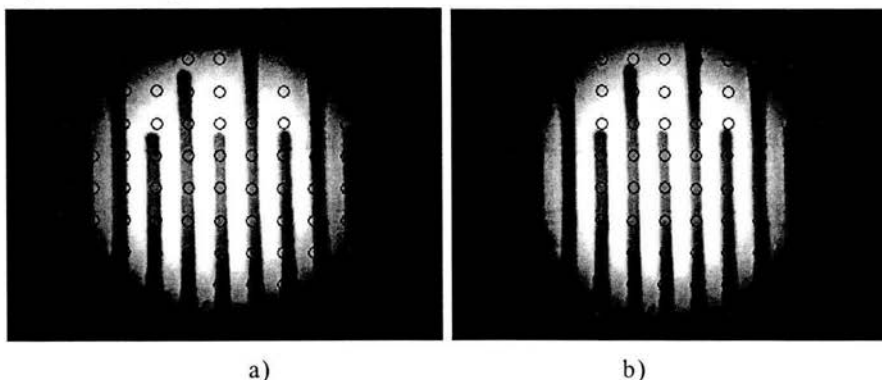


Figura 3.11. Compensación de la distorsión. a) Imagen de una escala. b) Imagen corregida.

3.3. Extracción de la medición

En una imagen de escala graduada es importante reconocer los diferentes patrones que forman las marcas de la escala. Por ejemplo, la figura 3.12 muestra una escala con 3 patrones distintos de marcas a las cuales llamaremos barras. El algoritmo para la extracción de la medición es realizado sobre imágenes libres de distorsión, después de aplicar el procedimiento para la calibración de la cámara discutido en 3.2. El procedimiento toma como base la siguiente situación ilustrativa que ocurre cuando se miden escalas en condiciones reales. Primero, la escala es inspeccionada y el patrón de longitud son puestos en concordancia con el cero inicial; posicionando la referencia en el dispositivo de observación sobre el cero de la escala. Entonces el dispositivo de observación es posicionado por el sistema cerca de un intervalo de medición previamente especificado, por ejemplo 10mm en la figura 3.12. Debido al error inherente del sistema de posicionamiento, éste es posicionado en 9.95mm. Entonces se extrae la magnitud real de la barra de 10mm sobre la imagen ampliada de la escala. En el ejemplo de la figura 3.12, y asumiendo un patrón de longitud libre de error, el valor real de la barra de 10mm en la escala es de $9,95+0,03=9,98\text{mm}$, resultando en un error de $10-9,98=0,02\text{mm}$.

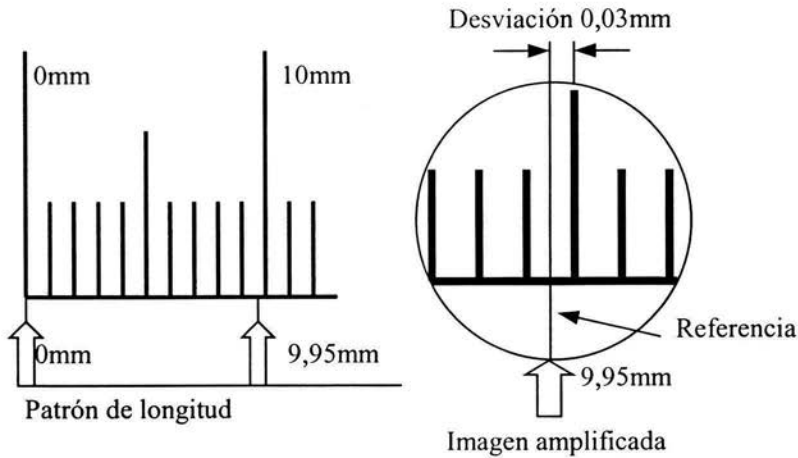


Figura 3.12. Escala graduada y patrones de barras.

La generalización del procedimiento descrito nos conduce a la siguiente ecuación para el cálculo del error:

$$\text{Error} = \text{Intervalo de medición} - (\text{Patrón de longitud} \pm \text{Desviación}) \quad (3.9)$$

En donde el signo \pm es determinado de acuerdo a la posición de la referencia. Si la referencia está a la derecha de la barra principal, entonces el signo es positivo. Por otra parte, si la referencia está a la izquierda de la barra principal, entonces el signo es negativo.

El procedimiento anterior para la extracción de la medición es automatizado utilizando técnicas de visión por computadora en las siguientes dos etapas: simplificación de la imagen y análisis de la imagen, descritas a continuación.

3.3.1. Simplificación de la imagen

El primer paso, es tratar de aislar la información de la imagen que realmente necesitamos, en nuestro caso, esta información son las barras de las escalas. Para ello eliminamos el exceso de iluminación y el trasfondo, esto con el propósito de aislar en la imagen solamente las barras que son de nuestro principal interés. Para eliminar la iluminación y trasfondo es necesario tomar una imagen de referencia de la superficie a medir, pero libre de cualquier marca, ver figura 3.13.

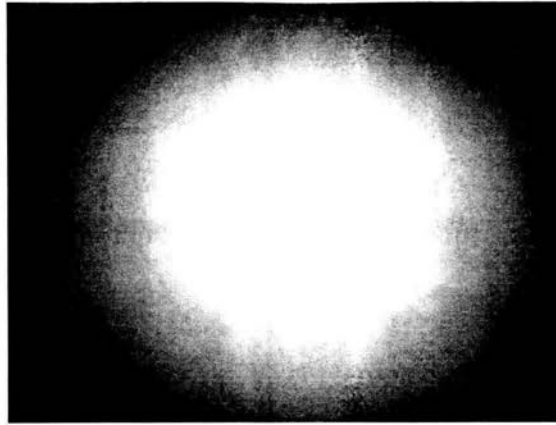


Figura 3.13. Imagen de iluminación de referencia.

Esta imagen nos servirá como referencia durante el proceso de medición, esto es, a cada imagen capturada durante el proceso de medición se le restará la imagen de referencia; con ello se obtendrán las barras en video inverso pero apropiadas para continuar con el proceso, ver figura 3.14.

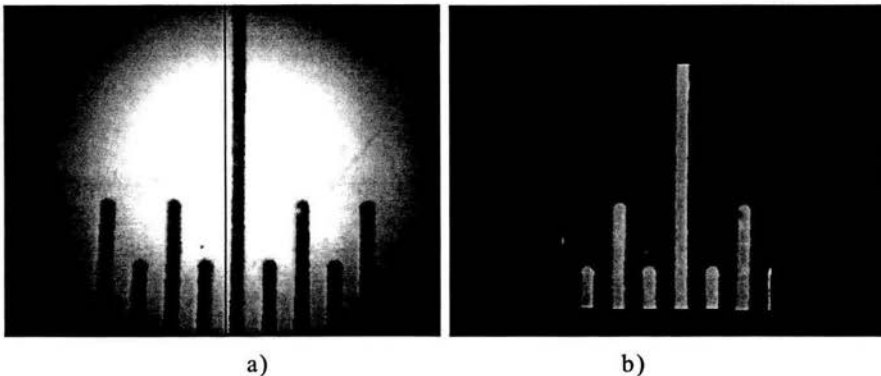


Figura 3.14. Eliminación del trasfondo. a) Imagen original. b) Imagen simplificada.

3.3.2. Análisis de la imagen

Como ya se había mencionado las imágenes simplificadas son convertidas a escala de grises, esto nos permite tener una matriz con valores de intensidad. Para que a la información obtenida le podamos asignar operaciones matemáticas, es necesario primero obtener una función matemática que contenga la información de la imagen. Al tener una imagen simplificada podemos hacer un barrido de izquierda a derecha e ir detectando por columnas estos niveles de intensidad [4].

Cada vez que se encuentre un nivel de intensidad por arriba de un cierto umbral en la columna x , se le irá incrementando en uno al valor $f(x)$. Entonces obtendremos una función tipo $f(x)$ donde x serán las columnas y $f(x)$ las intensidades de iluminación por arriba del umbral seleccionado.

Como a esta imagen ya se le quitó la iluminación y el trasfondo, los valores por encima del umbral serán únicamente los valores de las barras, quedando la función $f(x)$, como se muestra en la figura 3.15.

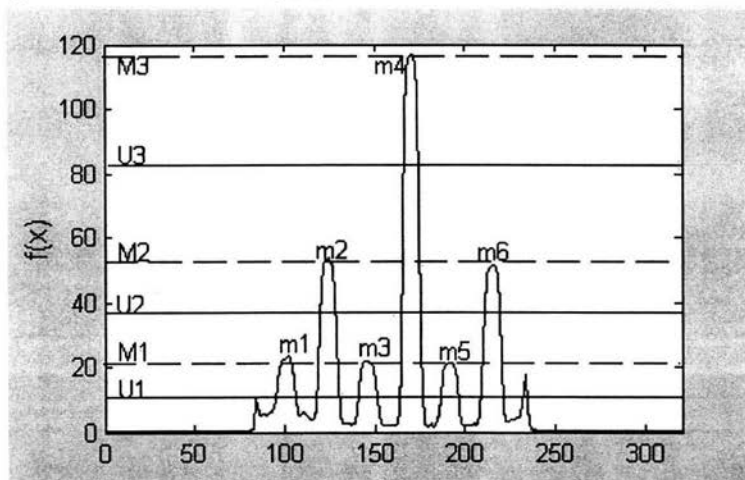


Figura 3.15. Función $f(x)$.

La función $f(x)$ proporciona información con el objetivo de establecer valores de umbral para la clasificación de barras de interés. En la figura 3.15, se extraen valores máximos locales, m_i , al rastrear $f(x)$ en ventanas pequeñas. La información estadística de los m_i 's proporciona tres tipos de puntos de control, $M_{1,2,3}$, resultando en tres umbrales $U_{1,2,3}$, y tres tipos de barras. La primera derivada de $f(x)$, $f'(x)$, contiene información acerca de los bordes de las barras, figura 3.16. Los bordes positivos están definidos por los máximos locales y los bordes negativos por los mínimos locales. Esta información es útil en la determinación de la distancia entre barras, d_i . El promedio de las distancias entre barras, d , en pixeles, tiene una correspondencia en unidades de longitud, ya que de antemano es conocida la resolución de la escala.

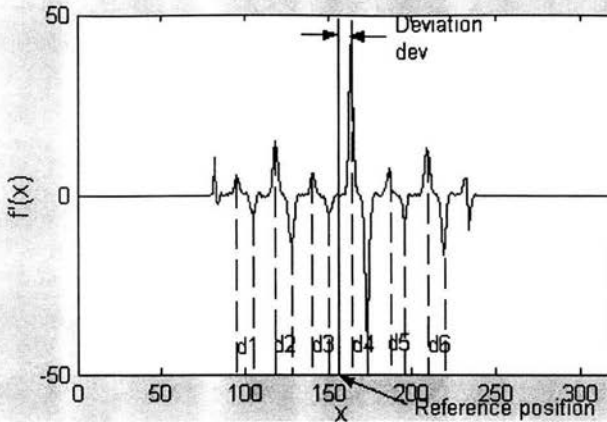


Figura 3.16. Función $f(x)$.

La desviación de una barra predefinida de la posición de referencia en el dispositivo de observación, dev , es extraída y trasladada a unidades de longitud utilizando la siguiente relación:

$$Desviación = \frac{dev(\text{pixels})}{d(\text{pixels})} \text{resolucion}(\text{unidades de longitud})$$

Entonces, para un intervalo de medición predefinido, el sistema de medición realiza los siguiente: Posiciona el dispositivo de observación, captura el valor en el patrón de longitud, procesa la imagen para extraer la desviación y el error es calculado utilizando la ecuación (3.9).

3.4. Control de posición

El posicionamiento del dispositivo de observación requiere de un algoritmo de control digital con el fin de alcanzar la automatización del proceso. Con esta idea en mente, se implementó un controlador PID en modo posicional de acuerdo al esquema de la figura 3.17.

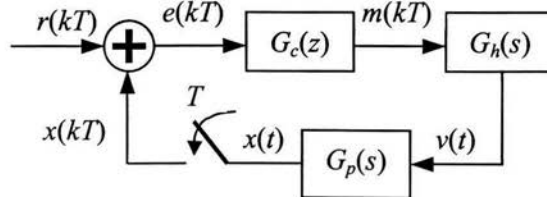


Figura 3.17. Diagrama a bloques para el sistema digital de control de posición.

En donde $G_c(z)$ es el controlador PID digital, $G_h(s)$ es un retenedor de orden cero y $G_p(s)$ es la planta [18, 21]. En el dominio discreto el sistema en conjunto resulta en la siguiente ecuación.

$$G(z) = Z \{ G_h(s) G_p(s) \}$$

$$G_c(z) = K_p + \frac{K_i}{1-z^{-1}} + K_d(1-z^{-1})$$

Entonces la respuesta en lazo cerrado es

$$G_l(z) = \frac{G_c(z)G(z)}{1+G_c(z)G(z)}$$

Esta ecuación es utilizada para sintonizar los parámetros del controlador (K_p , K_i y K_d), utilizando funciones de MatLab. La implementación actual para el control de posición es mediante el esquema de la figura 3.18 y software específico. El software incluye temporizadores de alta resolución con el objeto de cumplir con un periodo de muestreo de $T = 10\text{kHz}$. Por otra parte, la implementación utiliza la infraestructura hardware original de la máquina descrita en el capítulo 2.

En la figura 3.18, la planta está conformada por el sistema electromecánico que forma el conjunto de sistema mecánico y motor CD descritos en 2.2.. Con el objetivo de conseguir operación en tiempo real, la implementación opera el codificador óptico mediante la tarjeta de interfase para PC descrita en 2.3.2. Se utilizó el circuito electrónico impulsor CD, descrito en 2.3.3, que suministra las señales PWM a la planta.

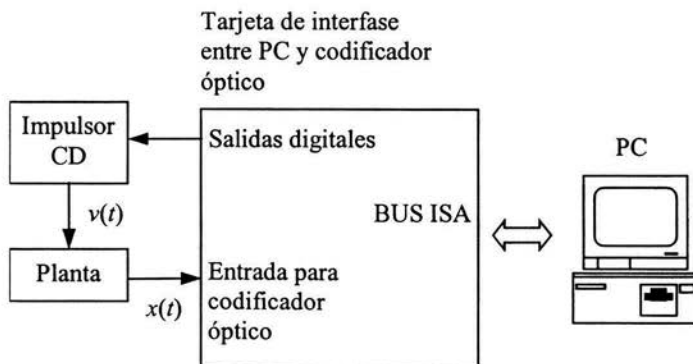


Figura 3.18. Implementación del control de posición.

De la función de transferencia del controlador digital, ecuación (1.1), tenemos la siguiente implementación

$$m = ((k_p + k_i + k_d) * (r - \text{teta})) -$$
$$((k_p + 2 * k_d) * (r_1 - \text{teta}_1)) +$$
$$(k_d * (r_2 - \text{teta}_2)) +$$
$$m_1$$

donde:

k_p , k_i y k_d , son constantes cuyos valores asignados son $k_p = 20.0$, $k_d = 10000.0$ y $k_i = 0.0$.

r es el intervalo de medición.

teta es el valor del codificador transformado a milímetros.

r_1 , r_2 , teta_1 , teta_2 y m_1 son los valores previos de las variables anteriores.

Este controlador una vez iniciado es actualizado cada $T = 10\text{kHz}$, esto y los valores de las variables k_p , k_i y k_d permiten posicionar el sistema con un mínimo de error. Sin embargo, aún este pequeño error es considerado en el momento de obtener los datos finales, logrando con ello un exactitud bastante aceptable.

Capítulo 4

Descripción del software desarrollado

Se realizaron dos programas que implementan el sistema propuesto para la medición descrito en 3.1. En este capítulo se explicarán estos programas desde dos puntos de vista: la del operador y la del programador. En forma adicional, se expondrá el proceso de medición que un usuario típico tiene que realizar para operar adecuadamente el instrumento de medición.

4.1. Introducción

La primera aplicación llamada “Calibración de Cámara 2.exe” se encarga de los procesos de extracción de características, estimación de la distorsión y compensación de la distorsión, descritos en la sección 3.2. El segundo, llamado “MCREG Final 2.exe”, se encarga de los procesos de simplificación de la imagen, análisis de la imagen y control de posición, como fue descrito en las secciones 3.3 y 3.4. Cabe mencionar que el programa MCREG Final usado en la versión anterior de la máquina de calibración, fue modificado para agregarle el código que realiza todas las acciones antes mencionadas [1].

Ambos programas están desarrollados en “Visual C++” versión 6.0 [6, 7, 13, 16]. El programa Calibración de Cámara 2 es una aplicación basada en diálogo sin utilizar la arquitectura típica vista-documento. En el caso de MCREG Final 2, es una aplicación de múltiples documentos (MDI), y sí utiliza la arquitectura vista-documento. Ambos programas utilizan una serie de librerías proporcionadas por Microsoft llamadas VisionSDK, estas librerías proveen un amplio conjunto de funciones que permiten la manipulación y despliegue de imágenes. Las

funciones son aprovechadas para obtener imágenes directamente de una cámara multimedia, las cuales son procesadas posteriormente para su reconocimiento y medición. En el caso de MCREG Final 2, es necesario manejar una librería dinámica, DDL, que permite la comunicación entre el programa y el sistema electrónico descrito en la sección 2.3.

En general los requerimientos mínimos para ejecutar y compilar los programas son los siguientes.

- Computadora PC Pentium I o superior (recomendado) con una ranura tipo ISA, 64 MB en Ram y al menos un puerto USB.
- Sistema operativo Windows 98 o 2000.
- Sistema electrónico que comprende un codificador óptico lineal ACU-RITE, una tarjeta ISA PC7266 y una tarjeta impulsora de motores.
- Cámara multimedia USB compatible con Video for Windows, VFW.
- Visual C++ 6 con VisionSDK instalado [24].
- Sistema mecánico que comprende el prototipo de máquina para medir escalas, motores eléctricos y sus acoplamientos, microscopio y su base para acoplamiento y rejilla patrón.
- Cables y conectores.

Las conexiones eléctricas necesarias para operar la máquina y realizar mediciones se muestran en la figura 4.1.

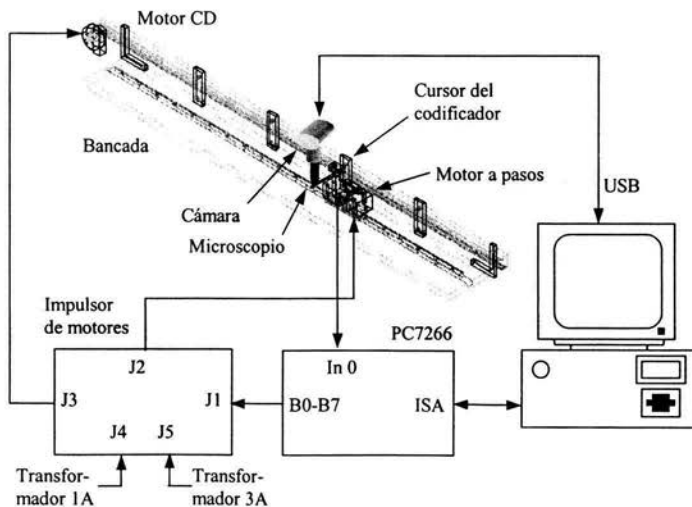


Figura 4.1. Conexiones en el sistema para la calibración de escalas graduadas.

En las próximas secciones se describirán con detalle cada uno de los programas.

4.2. Descripción en el ámbito del operador

La descripción en el ámbito del operador tiene como objetivo servir como una guía de usuario para operar los programas desarrollados y realizar mediciones con la máquina para la calibración de escalas.

Un proceso típico para medir con la máquina implica realizar los siguientes dos subprocesos.

- Calibrar la cámara. Se utiliza el programa Calibración de cámara 2 para estimar la distorsión del dispositivo de captura y así poder generar imágenes libres de distorsión que incrementen la exactitud del siguiente subproceso.
- Medir escalas. Se utiliza el programa MCRReg Final 2 para realizar mediciones completamente automatizadas sobre escalas graduadas.

4.2.1. Programa “Calibración de cámara 2”

La pantalla principal de dicho programa se muestra en la figura 4.2.

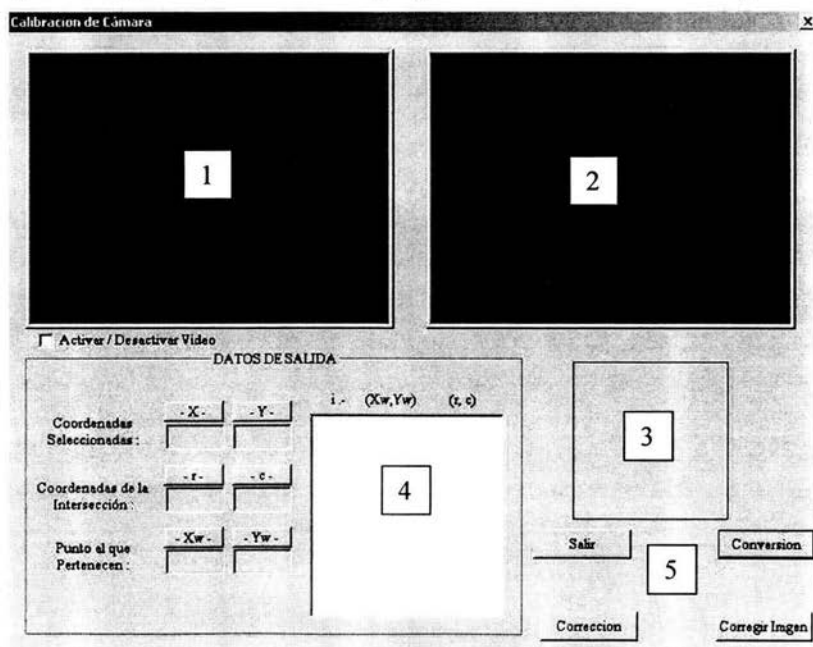


Figura 4.2. Programa “Calibración de cámara 2.exe”.

Como puede observarse, la pantalla está dividida en tres áreas.

- Sección de visualización de imágenes obtenidas de la cámara multimedia.
- Sección de visualización de los “Datos de salida”.
- Sección de controles.

La sección de visualización de imágenes consta de tres áreas, números 1,2 y 3. En el área número 1 se tiene el video en tiempo real de la cámara, con el propósito de posicionar la rejilla de calibración. En el área número 2 se obtiene una imagen fija a partir del video en la cual el usuario podrá definir las intersecciones de las líneas en la imagen de la rejilla patrón. En el área número 3 aparece un detalle de acercamiento, también en tiempo real, de la imagen del área número 1, esto con la finalidad de ayudar al operador a seleccionar de una mejor forma la intersección entre las líneas. Esto es necesario ya que se pueden trabajar con mallas de alta densidad que a simple vista podría ser complicado acertar en la intersección.

La sección de “Datos de salida”, número 4 en la figura 4.2, consta de 4 partes.

- Coordenadas Seleccionadas (x, y). En estas casillas aparecen las coordenadas de la imagen del área 2 al mover el ratón por encima de ella.
- Coordenadas de la Intersección (r, c). Aquí aparecen las coordenadas calculadas a nivel subpixel, esto es, el centro exacto de la intersección de líneas.
- Punto al que pertenecen (X_w , Y_w). Estas son las coordenadas globales a las cuales el punto seleccionado pertenece. En la sección 3.2.1 se mencionó que era necesario seleccionar los puntos de intersección de una manera previamente establecida, esto es, de izquierda a derecha y de arriba hacia abajo. Como por definición se utiliza una malla con cuadrados de 0.5x0.5 mm, las coordenadas asignadas mantienen esta misma relación, esto es el primer valor será (0,0), el segundo (0,0.5) y el tercero (0,1), etc.
- Resumen de coordenadas. En esta área (parte blanca) se tendrá un listado completo de todos los puntos que se fueron seleccionando así como su correspondiente coordenada global.

En la sección de controles, marcada con el número 5, aparecen los cuatro botones descritos a continuación.

- **Conversión.** Este botón toma una imagen del video correspondiente al área 1, la convierte a escala de grises y la coloca en el área 2, donde es procesada para la extracción de características. En este sentido, es común que la imagen capturada sea la de la malla patrón.
- **Corrección.** Una vez que el operador asignó a cada punto de la intersección un valor global, el proceso continúa al presionar este botón con el reconocimiento de todos los cuadriláteros en la imagen, se calculan y guardan los parámetros de la corrección para su uso futuro.
- **Corregir Imagen.** Este botón permanece oculto hasta que los procesos anteriores se hayan terminado. Su objetivo es mostrar que los resultados de la calibración sobre cualquier imagen sean satisfactorios, es decir, en este momento la malla utilizada para la calibración puede reemplazarse por cualquier otro objeto para obtener una imagen corregida.
- **Salir.** Termina con la ejecución del programa.

4.2.2. Proceso de calibración

Al iniciar el programa Calibración de cámara 2 aparecerá el cuadro de diálogo con el video en tiempo real en el área 1, figura 4.3. En este momento el usuario deberá colocar la malla patrón completamente paralela y alineada con respecto a la lente de la cámara. También podrá ajustar el foco de la lente del microscopio y la posición de la cámara para evitar imágenes borrosas. La iluminación también es parte fundamental para una correcta calibración; es necesario que no haya un exceso de ella ya que provoca que algunas líneas se pierdan y con ello las intersecciones deseadas. Una vez que se tienen todos los elementos aceptados se procede dando clic al botón Conversión.

El botón Conversión, capta inmediatamente la imagen actual y la convierte de un tipo RGB a escala de grises, esta última es colocada en el área 2. La imagen capturada por el sistema de visión es la proyección original de la malla patrón, por lo tanto se le considera distorsionada. Con esta imagen distorsionada, es posible iniciar el proceso de corrección, encontrando primero las intersecciones de la malla. Esto se realiza manualmente colocando el puntero del ratón sobre la imagen en escala de grises y presionando el botón izquierdo lo más acercado a la intersección más próxima a la esquina superior izquierda. El detalle de acercamiento del área 3 puede ayudar al operador en la definición de intersecciones.

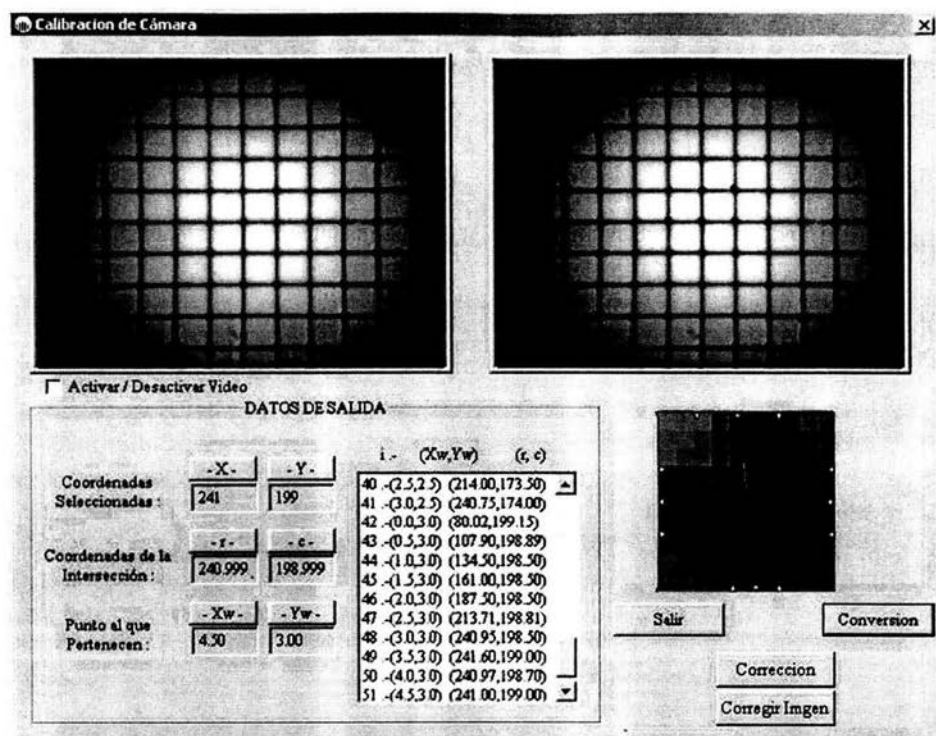


Figura 4.3. Ventana del programa Calibración de Cámara 2.

Cabe recordar que se deberá seguir un orden al seleccionar los siguiente puntos de las intersecciones, esto es, de izquierda a derecha y de arriba hacia abajo, ver figura 4.4.a. Aunque el programa tiene la capacidad de obtener todos los puntos observados en la imagen y procesarlos correctamente, es necesario tomar en cuenta que esta corrección será utilizada en el reconocimiento y medición de las líneas de las escalas, por lo tanto los puntos seleccionados deben formar un cuadrilateral, ya que cualquier otro polígono puede acarrear información errónea en el momento de su procesamiento, ver figura 4.4.b.

Como puede verse en la figura 4.4.a, al dar clic manualmente en la intersección, se presentará gráficamente en el área número 3, el reconocimiento realizado por el programa, esto es, se podrá observar, mientras el ratón no se mueva, los puntos encontrados conteniendo las líneas que obtienen el centro de la intersección a nivel subpíxel. Entonces la lista de coordenadas se irá llenando conforme el usuario ingrese nuevas intersecciones de la siguiente forma: i es el número de intersección, (X_w, Y_w) son las coordenadas globales estimadas de la

intersección y (r, c) son las coordenadas imagen de la intersección calculada a nivel subpixel.

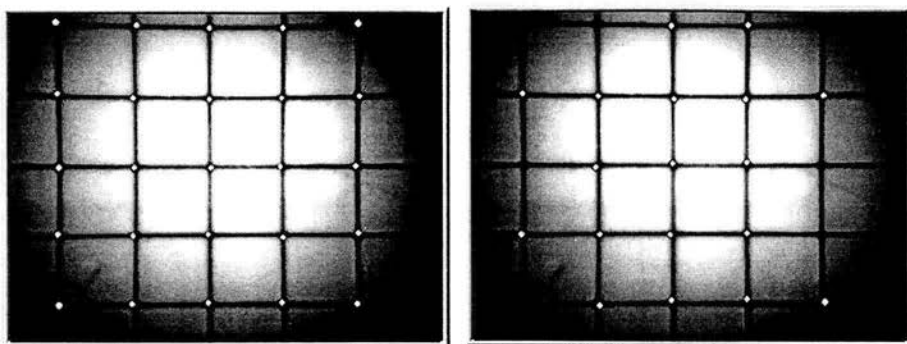


Figura 4.4. a) Selección de puntos de intersección, b) Selección errónea de puntos de intersección.

Una vez ingresados todos los puntos manualmente, es necesario dar clic al botón Corrección. Este botón se encarga de formar los arreglos de coordenadas (X_w, Y_w) (r, c) y su ordenamiento en cuadrilaterales, ver figura 4.5.a. El proceso que desencadena el botón Corrección es el siguiente:

- Se resuelve la ecuación 3.6 para encontrar los parámetros de la calibración (a_u, a_v, t_u, t_v) .
- Se resuelve la ecuación 3.4 para generar las coordenadas imagen sin distorsión (r_u, c_u) .
- Se resuelve la ecuación 3.8 para encontrar los parámetros de la distorsión $(a_{0,1,2,3} b_{0,1,2,3})$.

Una vez encontrados estos valores, la ecuación 3.8 es aplicada a todas las regiones cuadrilaterales de la imagen original y con ello corregir la distorsión. Lo anterior se realiza al presionar el botón Corregir imagen y el resultado de la corrección se aprecia en la figura 4.5.b.

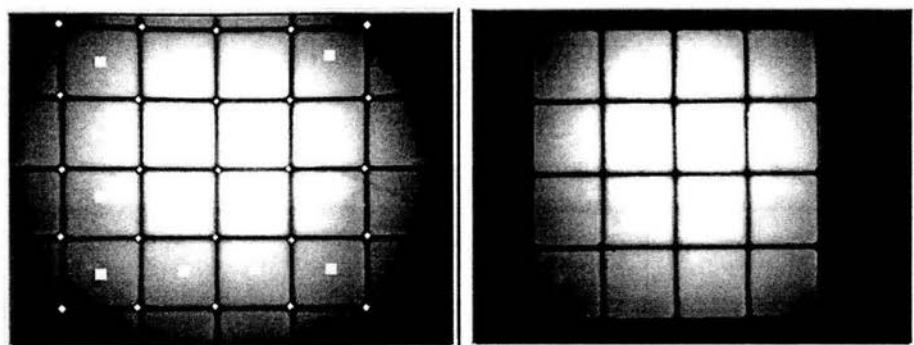


Figura 4.5. a) Arreglo de cuadrilaterales, b) Imagen corregida.

La calibración finaliza guardando la información necesaria para la calibración en tres archivos, estos son:

- **datos.txt:** Contiene las coordenadas globales (X_w , Y_w), las intersecciones a nivel subpixel encontradas (r , c) y los puntos corregidos (r_u , c_u).
- **R_c.txt:** Contiene las coordenadas imagen (r , c) que forman los vértices de los cuadrilaterales con distorsión.
- **Ru_cu.txt:** Contiene las coordenadas imagen (r_u , c_u) que forman los vértices de los cuadrilaterales sin distorsión.

Estos archivos son leídos por este programa y por MCREG Final 2 para corregir cualquier imagen deseada.

4.2.3. Programa “MCREG Final 2”

Este programa consta principalmente de tres interfases: interfase inicial, interfase principal e interfase de visualización y automatización.

Interfase inicial

Cuando el programa inicia, se muestra una pequeña caja de diálogo preguntando por los siguientes datos: Valor Inicial, Valor Final, Incremento y Resolución, todos estos valores serán considerados en milímetros (figura 4.6). El valor inicial y el valor final indican los valores primero y último respectivamente de la escala a ser medidos, por lo general el valor inicial será 0 ya que las mediciones se harán del origen de la regla. El valor final dependerá del tamaño y/o de la sección que se quiera calibrar, esto es, si una regla es de un metro y se quiere conocer su error a todo lo largo, el valor final será 1000, si solo se quiere calibrar la mitad será 500 y así sucesivamente. En la casilla Incremento, se debe escribir los saltos o incrementos que hará el

sistema. Una regla según su resolución y número de marcas diferentes se necesita calibrar en diferentes intervalos, por ejemplo se pueden considerar los errores de las líneas que señalan los decímetros, si se selecciona toda la regla de un metro, el sistema tendrá que hacer 10 mediciones por serie. El número de líneas y el tamaño depende de cada fabricante, es por ello que este dato necesita ser adaptado a cada regla. Por último, el valor Resolución nos servirá para conocer la relación existente del espacio entre barras, entre píxeles y milímetros. También este dato dependerá del tipo de escala y del fabricante.

Figura 4.6. Interfase inicial.

Interfase principal

Una vez aceptados los datos de la interfase inicial, aparece la pantalla principal del programa, Figura 4.7. Esta pantalla consta de cuatro áreas: Barra de Menús, Barra de estado, Área de Información y Barra de Herramientas.

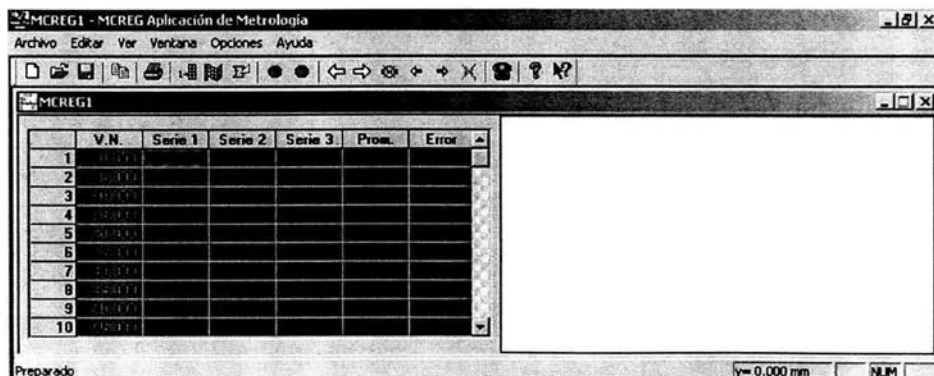


Figura 4.7. Interfase principal.
















Barra de Menús: En esta barra se encuentran 6 menús diferentes.

- Menú Archivo. Se encarga de abrir, guardar, imprimir, ver vista preliminar, cerrar un documento y salir del programa.
- Menú Editar. Permite copiar la información de la tabla de datos al portapapeles.
- Menú Ver. Oculta o muestra las barras de herramientas y estado.
- Menú Ventana. Organiza las ventanas abiertas.
- Menú Opciones. Permite poner a cero el codificador, pre-cargar un valor, calcular los promedio y obtener la grafica.
- Menú Ayuda. Muestra la ventana de ayuda del programa.

Barra de Estado: Aquí se muestra la información de los botones de la barra de herramientas y la lectura del codificador.

Área de Información: Esta área es subdivida en dos partes. En el lado izquierdo aparece la tabla de datos con los errores registrados durante el proceso de reconocimiento y medición. Una vez llenadas las tres series de datos y seleccionada la opción para obtener la gráfica, aparecerá automáticamente la gráfica que muestra los errores promedio de la calibración del lado derecho.

Barra de Herramientas: Esta sección muestra los siguientes botones:

-  Para iniciar una calibración nueva.
-  Abre una calibración existente.
-  Guarda calibración.
-  Copia información de la tabla al portapapeles.
-  Imprime el documento actual.
-  Toma lectura de forma manual.
-  Pone el codificador a cero ("resetea" el codificador)
-  Grafica de forma manual.
-  Mueve el motor de corriente directa hacia la izquierda.
-  Mueve el motor de corriente directa hacia la derecha.
-  Des-energiza el motor de corriente directa.
-  Mueve el motor de pasos hacia la izquierda.
-  Mueve el motor de pasos hacia la derecha.
-  Activa la caja de diálogo para la calibración automática.
-  Muestra la ayuda del programa.

En esta pantalla es posible mover los motores con la ayuda de los botones del menú de herramientas o con las flechas de dirección del teclado –Flecha izquierda y derecha para motor DC y flecha superior e inferior para motor de pasos-.

Pantalla de visualización y automatización

Para continuar con la calibración automatizada es necesario dar clic en el botón "Activar Cámara" de la barra de herramientas o utilizar el menú Opciones. En este momento aparecerá la caja de dialogo mostrada en la figura 4.8 con el video activo. Esta pantalla se divide en cuatro secciones: Área de visualización, Datos de Entrada, Datos de Salida y Botones de Acción.

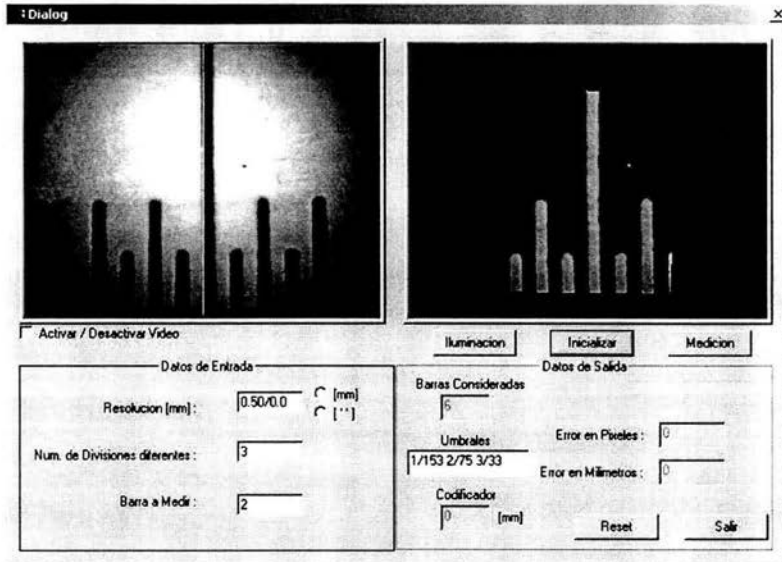


Figura 4.8. Interfase de visualización y automatización.

Área de visualización: Se conforma de dos rectángulos en la parte superior de la pantalla. En el izquierdo podemos observar el video en tiempo real donde se mostrará la escala ampliada. En el derecho aparecerán las imágenes a ser procesadas.

Datos de Entrada: Estas casillas deben ser llenadas para la medición de la regla. La Resolución es un dato que se estableció en la pantalla inicial de la figura 4.6, sin embargo, en éste cuadro de texto aparece en el momento de la medición su correspondiente valor en pixeles antecedido por una diagonal. Por ejemplo, el valor 0.5 / 24 indica una resolución de 0.5 milímetros y una resolución equivalente de 24 pixeles. El Numero de Divisiones Diferentes, se refiere a la cantidad de líneas con tamaños diferentes. Algunas escalas tienen una línea para milímetros, otra para centímetros y otra para decímetros. Otras tienen barras intermedias para indicar medios centímetros o medios decímetros. Ya que cada barra es diferente, es necesario

indicar que línea es la que corresponde a la medición que se va a tomar. Este dato se tiene que poner en el área Barra a medir.

Datos de Salida: En esta sección pueden verse 5 cuadros de texto.

- **Barras Consideradas.** Este dato indica el número de líneas que se están considerando dentro de la imagen. La información proporcionada es importante, ya que si alguna línea aparece cortada es descartada, aunque una parte de ella aparezca en la imagen.
- **Umrales.** La información que aquí aparece hace referencia al umbral asignado para el reconocimiento de cada tamaño diferente de barra. En la figura 4.8 se puede observar el texto 1/153 2/75 3/33, lo que indica que la barra número 1 (la mas grande) tendrá en promedio 153 valores de intensidad, la barra dos, 75 y la más pequeña 33.
- **Codificador.** Aparece la lectura actual del codificador en milímetros.
- **Error en píxeles.** Muestra el error encontrado al hacer la medición en píxeles.
- **Error en milímetros.** Muestra el error encontrado en milímetros.

Botones de Acción: Se muestran 5 botones de acción siguientes:

- **Iluminación.** Con este botón es posible obtener una imagen de iluminación de referencia para ser compensada de las imágenes a medir.
- **Inicializar.** Para comenzar con la medición, es necesario que se tengan los valores asignados de los distintos tamaños de las barras en la escala. Con la ayuda de las flechas de dirección del teclado para mover los motores se puede colocar la imagen de tal manera que se vayan visualizando todas las barras diferentes en la regla para obtener su tamaño y asignarlo a un tipo correspondiente.
- **Medición.** Una vez completadas la iluminación e inicialización, se puede proceder a la medición dando clic a éste botón, el cual iniciará el proceso hasta terminar con las tres series establecidas.
- **Reset.** Con este botón es posible borrar la información de los umbrales o detener el proceso en cualquier momento.
- **Salir.** Ocultará el cuadro de dialogo y regresará a la pantalla principal.

4.2.4. Proceso de medición

Al iniciar el programa MCRReg Final 2, aparece la pantalla mostrada en la figura 4.6, donde hay que decidir en que valor empezar, en cual terminar y en que intervalos se hará la medición. Estos datos nos darán como resultado el número de mediciones que se realizarán, por ejemplo si se tiene un Valor Inicial de 0, un Valor Final de 50 y un Intervalo de 1, se tendrán 50 mediciones de 1 milímetro cada una, si el Intervalo cambia a 10, se tendrán 5 mediciones de 10 milímetros cada una. En el dato Resolución simplemente se pondrá la resolución en milímetros que tenga la regla o escala graduada.

Al Aceptar esta ventana, se mostrará la ventana principal del programa mostrada en la figura 4.7. Es necesario dar clic en el icono de la cámara en la barra de herramientas de esta ventana para que aparezca el cuadro de diálogo para la medición, figura 4.8. Es necesario, en el área correspondiente, colocar el “Número de Barras Diferentes” que tiene la regla, posteriormente hay que escribir el número de la barra sobre la cuál se hará la medición. Los números asignados comienzan en el “1” para la barra más grande y se van incrementando para tamaños menores.

La iluminación de la regla tiene una especial importancia ya que de ello depende en mucho que los datos encontrados sean precisos. Es necesario que se localicé en la regla una sección libre de líneas y sea capturada la imagen mediante el botón Iluminación, ver figura 4.9. Esta imagen será utilizada para el reconocimiento de toda la regla, por ello si no se elimina de manera eficiente puede provocar que exista información falsa y provoque errores en la medición.

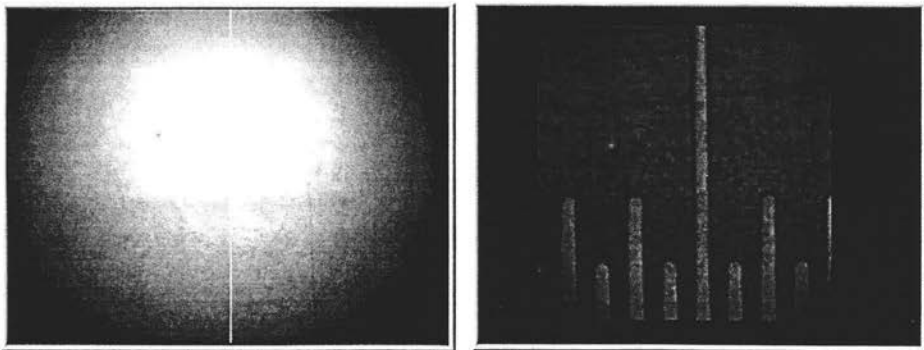


Figura 4.9. a) Iluminación en una escala, b) Imágenes con una incorrecta eliminación de iluminación.

Con el botón Inicializar es posible encontrar los umbrales que se le asignaran a cada tamaño diferente de línea. Es común que en una solo

imagen no se logren tener todas las líneas que hay en una escala, por ello es necesario desplazar el sistema de visualización por medio de los motores y capturar en diferentes puntos los umbrales de las barras. Esto se puede repetir tantas veces como sea necesario para lograr que se tengan todos los tamaños de las barras y sus respectivos umbrales. Los valores asignados a las líneas irán apareciendo en la caja de texto llamada “Umbrales”.

Hasta este momento se puede comenzar con el proceso de medición automática. Por medio de las teclas de dirección del teclado es necesario ubicar el sistema de visualización en la barra considerada como 0 u origen (generalmente al principio de la regla) y “resetear” el codificador para que también se considere esta posición como 0. Una vez hecho esto, se puede dar clic en el botón Medición para inicial el proceso.

A partir de este momento el programa toma el completo control del sistema. Comenzará a procesar la primera imagen, esto es eliminar la iluminación, corregir la distorsión de ésta, reconocer la barras encontradas en la imagen, decidir en cual de ellas se hará la medición y finalmente obtener el error existente y mandar el dato a la pantalla principal. Después de esto el programa mandará la orden para que el sistema de visualización se mueva a la siguiente barra, una vez estabilizado en ella se repite el proceso anterior. Cuando finaliza con la primera serie, el programa regresa el sistema al punto considerado como origen y repite todo lo anterior hasta completar tres series.

Al completarlas, la ventana se cerrará y mostrará los datos obtenidos y los resultados de la medición, tanto numérica como gráficamente.

Si se quiere realizar otra calibración, se tiene que cerrar esta ventana y usando la barra de herramientas o el menú edición Abrir un Nuevo Documento e ingresar los datos necesarios según la nueva regla a calibrar.

4.3. Descripción en el ámbito del programador

La siguiente es la explicación de los programas Calibración de cámara 2 y MCRég Final 2 en el ámbito del programador. Omitimos el código fuente en su totalidad debido a la gran extensión de éste. En lugar de ello, nos limitamos a introducir las porciones del código que sean de verdadera utilidad.

4.3.1. Programa “Calibración de cámara 2”

El programa Calibración de cámara 2, consta principalmente de cinco clases: CAboutDlg, CCalibraciondeCamara2App, CCalibraciondeCamara2Dlg, CCorrección y PUNTO, figura 4.10.

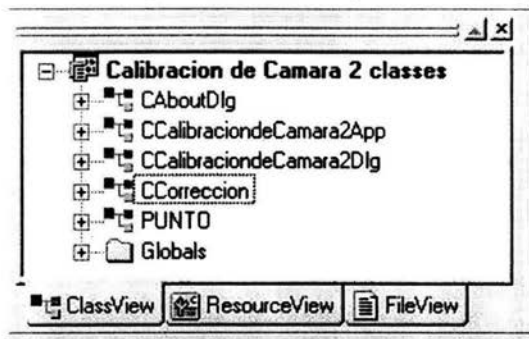


Figura 4.10. Clases del programa Calibración de Cámara 2.

Las tres primeras son las típicas clases que se generan al crear una aplicación basada en diálogo utilizando el AppWizard de Visual C++. En nuestro caso se eligió un proyecto tipo VisSDK, ya que, como se ha mencionado, contiene una serie de librerías para el manejo de imágenes y uso de cámaras digitales. Las clases de mayor interés son CCalibraciondeCamara2Dlg (figura 4.11) y CCorrección (figura 4.12), ya que contiene todos los métodos utilizados para la calibración,.

Al comenzar la aplicación una de sus principales tareas, es la de inicializar la cámara, esto es, ubicar si existe una cámara digital configurada en el sistema operativo y de ser así se tiene que verificar si está conectada y disponible. Estas acciones se realizan en los métodos InitInstance() de la clase CCalibraciondeCamara2App y OnInitDialog() de la clase CCalibraciondeCamara2Dlg.

El código mandará un error en los casos en que la cámara estuviera siendo usada por otra aplicación o estuviera desconectada.

El comienzo de la aplicación también es utilizado para asignar valores a variables claves que se necesitaran a lo largo del programa y empezar a “correr” el video en tiempo real de la cámara multimedia.

Una vez verificados estos datos, aparecerá el cuadro de diálogo mostrado en la figura 4.2. Las funciones principales para realizar lo mencionado se muestran a continuación:

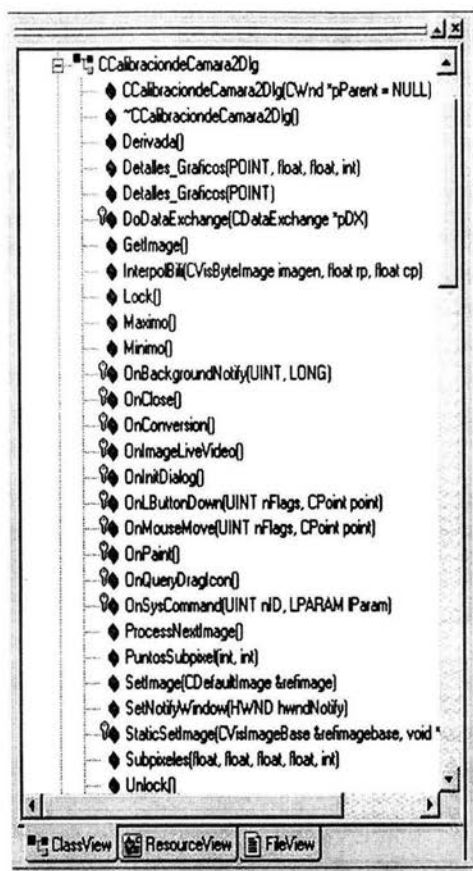


Figura 4.11. Métodos de la clase CCalibraciondeCamara2DIg.

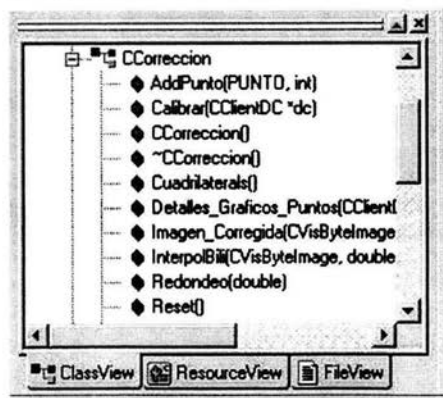


Figura 4.12. Métodos de la clase CCorrección.

```

CVisImageSource m_imagesource;
m_imagesource = VisFindImageSource();

CVisImageHandler<CVisRGBABYTEPixel> m_imagehandler;
m_imagehandler.ConnectToSource(((CCalibraciondeCamara2App*)AfxGetApp())-
>m_imagesource);

```

donde:

VisFindImageSource(): En la función que se encarga de localizar la cámara digital, abrir y mantener un canal de comunicación entre ella y la aplicación.

ConnectToSource(): Es la encargada de controlar la comunicación con la cámara digital, así como recibir y mandar la información hacia ella.

Una vez realizada esta inicialización se ejecuta un subproceso (Thread) que se encarga de mantener un orden y enviar las imágenes capturadas por la cámara. Los mensajes recibidos por este subproceso son recibidos por la función **OnBackgroundNotify()**, cuyo código principal se muestra a continuación:

```

LONG CCalibraciondeCamara2Dlg::OnBackgroundNotify(UINT, LONG)
{
    if ((!m_imagehandler.IsBackgroundProcessing()) &&
        (!m_image.IsValid() || (m_fImageDisplayed)))
    {
        // Process the image here if it wasn't processed in the
        // document's background thread.
        ProcessNextImage();
    }
    Lock();
    CDefaultImage image = m_image;
    CClientDC dc(this);
    image.DisplayInHdc(dc, m_rect1, 16, 16, SRCCOPY, 0);
    m_fImageDisplayed = true;
    Unlock();
}

```

donde:

IsBackgroundProcessing(): Es una función que retorna un número diferente de cero si existe una imagen capturada por el subproceso.

ProccesNextImage(): Es la función encargada de asignar a una variable la imagen capturada.

Lock(): Detiene el proceso. Manteniendo la imagen en memoria

UnLock(): Continúa el proceso liberando la imagen de memoria

En el momento en que se tenga en el video la imagen deseada, el botón **Conversión** es el encargado de hacer la transformación a escala de grises y colocar la imagen en el área correspondiente, esto es mediante el método **OnConversión()**. Las funciones principales en este método son las siguientes:

```
void CCalibraciondeCamara2Dlg::OnConversion()
{
    CClientDC dc(this);
    CVisByteImage imagenGris2(320,240);
    imagenGris2=VisBrightnessFromRGBA(m_imagenRGB);
    imagenGris2.DisplayInHdc(dc,rect1,m_offsetX,
        m_offsetY,SRCCOPY,0)
}

```

donde:

VisBrightnessFromRGBA(): Se encarga de hacer la transformación de píxeles entre distintos tipos de imágenes, en nuestro caso de un tipo RGB (variable `m_imageRGB` asignada por la función `ProcesNextImage()`) a un `ScaleGray` (variable `imagenGris2` tipo escala de grises).

DisplayInHdc(): Es un método, al igual que el anterior, de la librería `VisionSDK`, que permite desplegar una imagen en un área rectangular tipo `RECT` de un contexto de dispositivo (`DC`).

Una vez que la imagen en escala de grises aparece, el método que recibe el mensaje de movimiento del ratón, `OnMouseMove()`, cobra un mayor sentido. A partir de este momento cada vez que el puntero del ratón se desplace encima del área 2, la coordenada de esta posición será registrada por este método y con ella se actualizará el cuadro de texto "Coordenadas Seleccionadas", de la Figura 4.1.

Al dar clic en algún punto de la imagen, el método `OnLButtonDown()`, revisará si es una coordenada válida para procesar una imagen de esa zona. Toda coordenada válida será aquella que encuentre 6 píxeles alejados del margen de la imagen. Si el punto es validado, el método crea una imagen ampliada de 11 x 11 píxeles del área seleccionada y lo coloca en el área 3. El código principal es el siguiente:

```
void CCalibraciondeCamara2Dlg::OnMouseMove(UINT nFlags, CPoint point)
{
    CClientDC dc(this);
    CVisRGBABYTEImage SubImagenGris=m_imagenRGB;
    HDC SubImagen_hdc=SubImagenGris.Hdc();
    //Crea un Hdc para la subimagen
    if(SubImagen_hdc!=0)
    {
        HBITMAP SubImagen_bitmp=SubImagenGris.
        HBitmapCreateForHdc (SubImagen_hdc);
        SelectObject (SubImagen_hdc,SubImagen_bitmp);
        ::StretchBlt (dc.m_hDC,x_offset2,y_offset2,140,140,
        SubImagen_hdc,(point.x-m_offsetX)-5,(point.y-
        m_offsetY)-5,11,11,SRCCOPY);
        ::DeleteDC(SubImagen_hdc);
        ::DeleteObject(SubImagen_bitmp);
    }
}

```

donde:

`dc`: es una instancia del contexto de dispositivo usado.

SubImagenGris: es un objeto de un tipo de imagen RGB.

SubImagen_hdc: es un "handler" del contexto de dispositivo adecuado para el tipo de imagen RGB.

StretchBlt(): Es la función que permite colocar la imagen en el área seleccionada.

El usuario está en libertad de seleccionar cualquier punto dentro de la imagen, sin embargo vamos a asumir que seleccionará puntos muy cercanos a la intersección de dos líneas por ser propio a nuestro trabajo. Una vez que se colocó la imagen de detalle ampliado, el método llama a la función PuntosSubpixel() que será la encargada de encontrar el punto central a nivel subpixel dentro de la imagen ampliada.

El método PuntosSubpixel(), se encarga de delimitar los cuatro bordes de la imagen ampliada y mandar los datos necesarios al método Subpíxeles(), este método encontrará los puntos medios de los bordes y regresará la información al método PuntosSubpixel(), ver círculos de la figura 4.13. Aquí cada borde ya delimitado es subdividido en 5 partes por píxel, esto es, un borde de 10 píxeles será dividido en 50 partes consideradas como subpíxeles, esto con el afán de tener un mejor acercamiento a la intersección real. El código se muestra a continuación:

```
PUNTO CCalibraciondeCamara2Dlg::Subpíxeles(float ml_fl, float ml_fin, float
ml_diffcp, float ml_diffrp, int ii)
{
    PUNTO Aux;
    float rp[50],cp[50];
    float Subpixel_Max,Subpixel_Min;
    int count=0,iMax,iMin;
    CString datos;
    for(float ifl=ml_fl;ifl<=ml_fin;ifl=ifl+0.2)
    { //CREA UN ÁREA VECINA DE 50 PUNTOS(esto es 10/0.2)
        EN FORMA HORIZONTAL
        rp[count]=ifl;
        cp[count]=ml_diffcp;
        pp[count]=InterpolBili(imagenGris,rp[count],cp[count]);
        //HACE LA INTERPOLACION BILINEAL
        count=count+1;
    }
    Derivada();
    // DERIVADA, MÁXIMO Y MÍNIMO DE LA "FUNCION" PP[]
    iMax=Maximo();
    iMin=Minimo();
    Subpixel_Max=rp[iMax];// VALOR DEL SUBPIXEL ENCONTRADO
    Subpixel_Min=rp[iMin];
    Aux.x=ml_diffcp;
    Aux.y=(fabs(Subpixel_Max-
        Subpixel_Min)/2)+min(Subpixel_Max,Subpixel_Min);
    Detalles_Graficos(Aux,iMax,iMin,ii);
}
```

De este código cabe resaltar la siguientes funciones:

InterpolBili(): Este método se encarga de realizar la interpolación bilineal del punto enviado.

Derivada(): Crea la función derivada (dpp[i]) del vector pp[j].

Máximo(): Regresa el índice del valor máximo de la función dpp[i].

Mínimo(): Regresa el índice del valor mínimo de la función dpp[i].

Detalles_Graficos(): Este método se encarga de dibujar en el zoom los puntos y líneas para indicar el proceso del punto encontrado.

En la imagen 4.13, se pueden observar los puntos máximos y mínimos encontrados (triángulos) así como los puntos medios (círculos). La función PuntosSubpixel(), recibe estos últimos y se encarga de trazar las líneas auxiliares para encontrar el punto central en nivel subpixel, ver flecha blanca de la figura 4.13. Cabe señalar que el punto negro visto en esta figura es el píxel seleccionado por el usuario, no el punto central necesariamente.

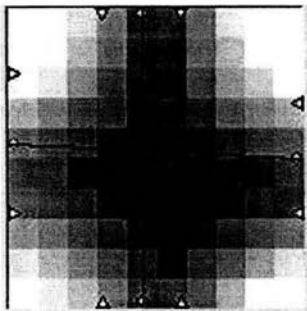


Figura 4.13. Detalles de la imagen de detalle ampliada.

Este proceso se repite toda vez que el operador haga clic en alguna parte de la imagen válida. Estos puntos son enviados a al método AddPunto() de un objeto de la clase Ccorrección. Este método se encarga de guardar cada uno de las intersecciones encontradas y asignarles sus respectivos valores globales. El código se encuentra a continuación.

```
PUNTO Ccorreccion::AddPunto(PUNTO centro, int ii)
{
    PUNTO xx;
    Vector_Puntos[indice][2]=centro.x;
    Vector_Puntos[indice][3]=centro.y;
    if(indice==0) aux2=Vector_Puntos[indice][2];
    if(aux+10<Vector_Puntos[indice][3])
    {
        yGlobal++;
        //TENER EN CUENTA QUE LOS "BRINCOS" NEGATIVOS EN X TIENEN
        QUE SER DE UNO EN UNO, ESTO ES: 0,-1,-2 no 0,-1,-3
        if(Vector_Puntos[indice][2]<aux2-10)
        {
```

```

    xGlobal_Aux--;
    aux2=Vector_Puntos[indice][2];
}
if (Vector_Puntos[indice][2]>aux2+10)
{
    xGlobal_Aux++;
    aux2=Vector_Puntos[indice][2];
    xGlobal=xGlobal_Aux;
}
else xGlobal++;Vector_Puntos[indice][0]=xGlobal*0.5;
Vector_Puntos[indice][1]=yGlobal*0.5;
aux=Redondeo(Vector_Puntos[indice][3]);
}

```

Como puede observarse, los puntos encontrados así como su correspondiente valor global son almacenados en un vector llamado `Vector_Puntos`, esta variable cobra gran relevancia al hacer la calibración ya que contiene la principal información que se necesita para ella. También se puede observar que en este caso se considera una malla de 0.5 milímetros entre una intersección y otra, por lo tanto las coordenadas globales serán múltiplos de 0.5.

Una vez que el usuario terminó de asignar las intersecciones de la malla, puede continuar con la calibración haciendo clic en el botón `Corrección`. El método que recibe este comando es `OnCorrección()`, el cual se encarga de llamar a los métodos de la clase `CCorrección` que hagan posible la calibración. Una vez corregida se muestra en el área 2 esta imagen, el código es.

```

void CCalibraciondeCamara2Dlg::OnCorreccion()
{
    CClientDC dc(this);
    CVisByteImage Imagen_Corregida;
    correccion.Calibrar(&dc);
    correccion.Cuadrilaterals();
    Imagen_Corregida=correccion.Imagen_Corregida(imagenGris);
    Imagen_Corregida.DisplayInHdc(dc,rect1,m_offsetX,
        m_offsetY,SRCCOPY,0);}

```

Las funciones principales de este código son `Calibrar()`, `Cuadrilaterals()` e `Imagen_Corregida()`. Todas ellas parte de la clase `CCorreccion`. A continuación explicamos cada una de ellas.

`Calibrar()`: En esta función se encuentra implementado el método propuesto para la corrección de distorsiones visto en el capítulo 3. Las ecuaciones 3.6, 3.7 y 3.8 son utilizadas aquí con los valores requeridos por ellas, esto es.

```

bool CCorreccion::Calibrar(CClientDC *dc)
{
    PUNTO cu_ru;fstream io("datos.txt",ios::out);
    CString aux,aux1;
    double A=0.0;
    double B=0.0;
    double C=0.0;double Rc=0.0;
    double C0=0.0;
    double D1=0.0;

```

```

double D2=0.0;
for(int i=0; i<indice;i++)
{
    A=A+Vector_Puntos[i][0];
    B=B+Vector_Puntos[i][1];
    C=C+((Vector_Puntos[i][0]*Vector_Puntos[i][0])+
        (Vector_Puntos[i][1]*Vector_Puntos[i][1]));
    Rc=Rc+Vector_Puntos[i][3];
    C0=C0+Vector_Puntos[i][2];
    D1=D1+((Vector_Puntos[i][2]*Vector_Puntos[i][0])+
        (Vector_Puntos[i][3]*Vector_Puntos[i][1]));
    D2=D2+((Vector_Puntos[i][3]*Vector_Puntos[i][0])-
        (Vector_Puntos[i][2]*Vector_Puntos[i][1]));
}
A=A/indice;B=B/indice;
C=C/indice;
Rc=Rc/indice;
C0=C0/indice;
D1=D1/indice;
D2=D2/indice;
CVisDMatrix V(4,1,0),M(4,4,0),P(4,1,0),PP(4,4,0);
V[0][0]=D1; V[1][0]=D2; V[2][0]=C0; V[3][0]=Rc;
M[0][0]=C; M[0][1]=0; M[0][2]=A; M[0][3]=B;
M[1][0]=0; M[1][1]=C; M[1][2]=-B; M[1][3]=A;
M[2][0]=A; M[2][1]=-B; M[2][2]=1; M[2][3]=0;
M[3][0]=B; M[3][1]=A; M[3][2]=0; M[3][3]=1;
PP=M.Invert();
P=M*V;
cu=new double[indice];
ru=new double[indice];
for(i=0;i<indice;i++)
{
    cu[i]=(P[0][0]*Vector_Puntos[i][0]-
        (P[1][0]*Vector_Puntos[i][1])+P[2][0];
    ru[i]=(P[1][0]*Vector_Puntos[i][0]+(P[0][0]*
        Vector_Puntos[i][1])+P[3][0];
    cu_ru.x=cu[i];
    cu_ru.y=ru[i];
    Detalles_Graficos_Puntos(dc,cu_ru,0);
    aux.Format("%f%f%f%f%f",
        Vector_Puntos[i][0],Vector_Puntos[i][1],
        Vector_Puntos[i][2],Vector_Puntos[i][3],
        cu[i],ru[i]);
    auxl=aux+"\n";
    io.write(auxl,auxl.GetLength());
}
}
io.close();
}

```

Como se puede ver en el código anterior la mayor parte de las operaciones son del tipo matemáticos y creo no necesitan mayor explicación, sin embargo hago notar que en las últimas líneas la información es almacenada en un archivo llamado "datos.txt", este archivo es uno de los utilizados por el programa MCREG Final 2.

Cudrilaterals(): Este método se encarga de encontrar, a partir de la información guardada en la variable Vector_Puntos, los cuadriláteros que requiere el método visto en la sección 3, un ejemplo de ello se encuentra en la figura 4.5.a.

Imagen_Corregida(): Esta función se encarga de implementar la ecuación 3.8 vista en la sección 3.2.3 y aplicarla a toda imagen que se requiera, basándose en la información generada en las funciones anteriores. El resultado será una imagen sin distorsiones. Su código se presenta a continuación.

```

CVisByteImage CCorreccion::Imagen_Corregida(CVisByteImage imagenGris)
{
    int j,i,iii;
    CVisDMatrix Dx(4,1,0),Dy(4,1,0),U(4,4,0),A(4,1,0),B(4,1,0);
    CVisByteImage Imagen_Corregida(320,240);
    for (i=0;i<ii;i++)
    {
        Dx[0][0]=r_c[i][0][0];
        Dx[1][0]=r_c[i][0][1];
        Dx[2][0]=r_c[i][0][2];
        Dx[3][0]=r_c[i][0][3];
        Dy[0][0]=r_c[i][1][0];
        Dy[1][0]=r_c[i][1][1];
        Dy[2][0]=r_c[i][1][2];
        Dy[3][0]=r_c[i][1][3];
        U[0][0]=1;
        U[0][1]=ru_cu[i][1][0];
        U[0][2]=ru_cu[i][0][0];
        U[0][3]=ru_cu[i][0][0]*ru_cu[i][1][0];
        U[1][0]=1;
        U[1][1]=ru_cu[i][1][1];
        U[1][2]=ru_cu[i][0][1];
        U[1][3]=ru_cu[i][0][1]*ru_cu[i][1][1];
        U[2][0]=1;
        U[2][1]=ru_cu[i][1][2];
        U[2][2]=ru_cu[i][0][2];
        U[2][3]=ru_cu[i][0][2]*ru_cu[i][1][2];
        U[3][0]=1;
        U[3][1]=ru_cu[i][1][3];
        U[3][2]=ru_cu[i][0][3];
        U[3][3]=ru_cu[i][0][3]*ru_cu[i][1][3];
        B=U.Invert()*Dx;
        A=U*Dy;
        double dx,dy;
        int aux=Redondeo(ru_cu[i][1][3]);
        int aux1=Redondeo(ru_cu[i][0][1]);
        for (iii=Redondeo(ru_cu[i][1][0]);iii<=
            Redondeo(ru_cu[i][1][3]);iii++)
        {
            for (j=Redondeo(ru_cu[i][0][0]);j<=
                Redondeo(ru_cu[i][0][1]);j++)
            {
                dy=A[0][0]+(A[1][0]*iii)+(A[2][0]*j)+(A[3][0]*j*iii);
                dx=B[0][0]+(B[1][0]*iii)+(B[2][0]*j)+(B[3][0]*j*iii);
                Imagen_Corregida(j,iii)=
                    InterpolBili(imagenGris,dx,dy);
            }
        }
    }
    return Imagen_Corregida;
}

```

Finalmente el botón “Corregir Imagen” se utiliza para corregir la distorsión de cualquier otra imagen que se coloque en la lente del microscopio, eso con el fin de revisar si la calibración es la adecuada,

en caso contrario utilizando el botón Conversión se puede borrar toda la información y comenzar de nuevo todo el proceso para tratar de mejorar la calibración. El método que recibe el comando de este botón, es `OnCorreccionImg()`, y el código es muy parecido al de la función `Imagen_Corregida()`, por lo tanto no será detallado.

Con ello se completa la calibración de la cámara y se da paso al reconocimiento y medición del programa MCREG Final 2.

4.3.2. Programa “MCREG final 2”

El programa MCREG Final 2, mantiene las nueve clases originales del programa MCREG Final y se agregó una mas, `CDlg_Rec_y_Med()`, para lograr la automatización y la calibración, ver figura 4.14 [].

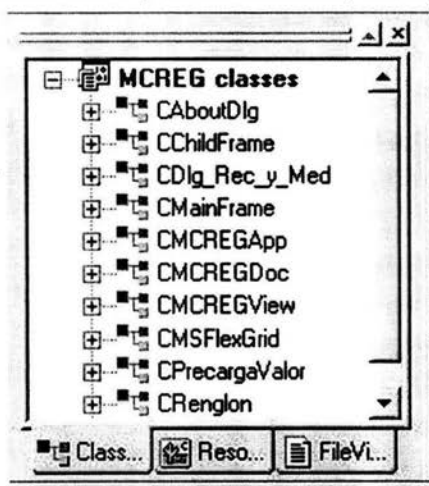


Figura 4.14. Clases del programa MCREG Final 2.

La clase `CMCREGView()`, fue modificada en algunos métodos y algunos otros fueron agregados, los métodos de la clase se muestra en la figura 4.15. Puede verse que este programa maneja la arquitectura Vista-Documento. Varias funciones de la clase `CMCREGView()` serán explicadas con detalle al igual que la clase `CDlg_Rec_y_Med()` de la figura 4.16, el resto puede consultarse en la referencia [].

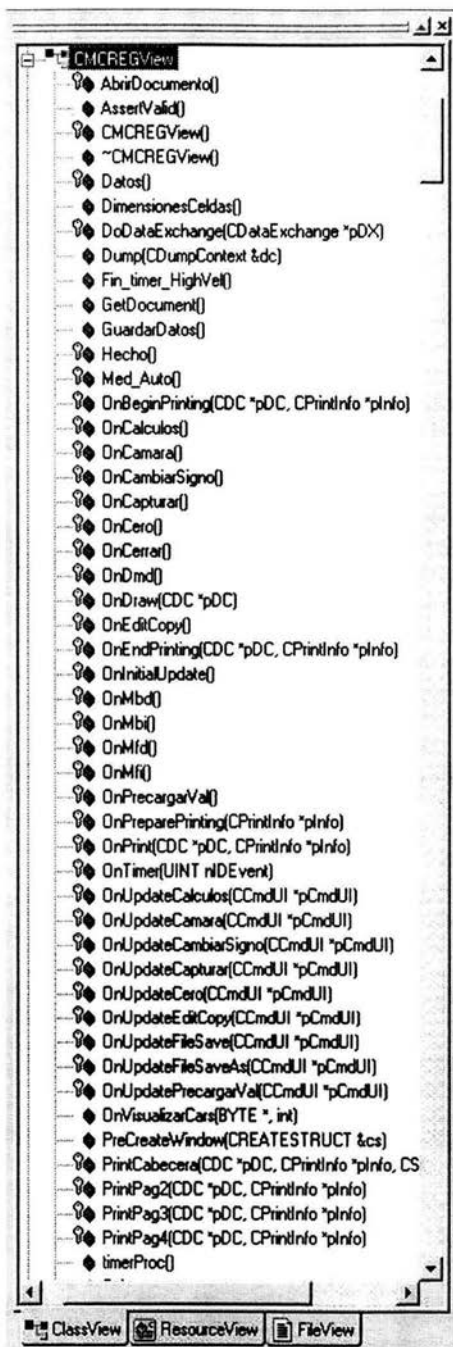


Figura 4.15. Clase CMCREGView.

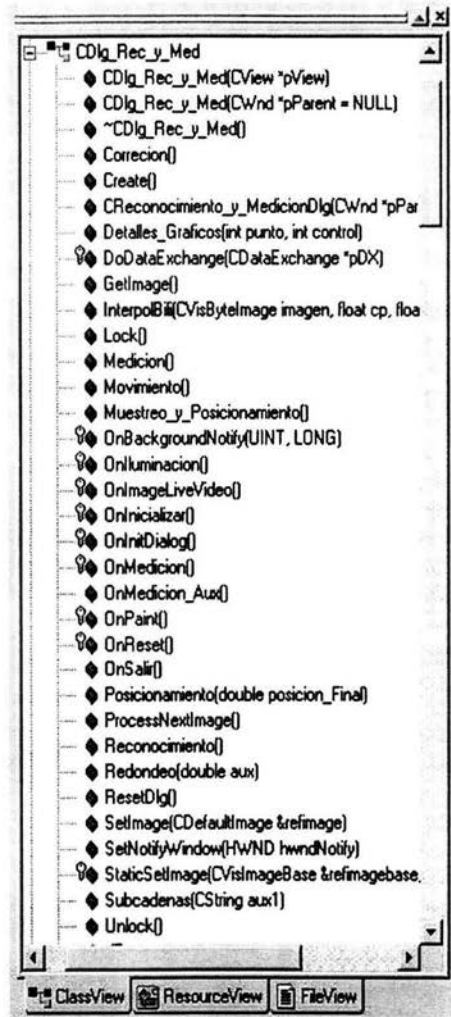


Figura 4.16. Clase CDlg_Rec_y_Med.

La primera caja de dialogo, figura 4.6, es manejada por la clase `CRenglon()`, cuya única finalidad es la de obtener los datos Valor Inicial, Valor Final, Incremento y Resolución. Estos son guardados en la variables `m_ValorInicial`, `m_ValorFinal`, `m_Incremento`, `m_Resolución` respectivamente. En algunos métodos más adelante se hará referencia a estas.

Al Aceptar la caja de dialogo inicial, aparece maximizada la pantalla principal de la aplicación, ver figura 4.7. En este momento ya se tiene pleno control tanto de la cámara multimedia, como de los motores por

medio de la tarjeta PC7266. Hasta este momento la cámara digital todavía no muestra las imágenes en pantalla y los motores no muestran movimiento alguno, sin embargo las funciones de inicialización ya están activas y listas para recibir las ordenes necesarias para el control. En el caso de la cámara, la configuración inicial es la misma que en el programa Calibración de Camara 2, vista en la sección 4.3.1, en el caso de la tarjeta PC7266 las funciones principales para la inicialización están implementadas en el método OnInitialUpdate() y se muestran a continuación.

```
void CMCREGView::OnInitialUpdate()
{
..if (PC7266INITIALIZE()==0)
{
//Iniciar direccion
if(PC7266SETBASEADDRESS(0x0300)!= 0)
    AfxMessageBox("Error al iniciar dirección
    en la PC7266", MB_ICONSTOP);
//Iniciar el puerto de salida
int salida=ContMDC;
salida=ContMDC<<2;
int out8=(salida&255)|(Secuencia[indiceM]&3);
if(PC7266OUTPUT(out8)!=0) AfxMessageBox("Error al
    escribir en el puerto", MB_ICONSTOP);
//Poner la lectura a cero
if(PC7266RESETCOUNTER(0)!=0) AfxMessageBox("Error en el
    reset", MB_ICONSTOP);
//Poner el Multiplicador X 4
if(PC7266SETMULTIPLIER(4)!=0) AfxMessageBox("Error al
    fijar modo X4", MB_ICONSTOP);
}
else
{
    if(AfxMessageBox("No se encuentra la 32pc7266", 49) ==
        IDOK) return;
    else PostQuitMessage(0);
}
}
}
```

donde:

PC7266INITIALIZE(): Es la función encargada de inicializar la tarjeta regresando un valor 0 si fue satisfactoria.

PC7266SETBASEADDRESS(): Se encarga de asignar la dirección en memoria a la que se hará referencia para el envío y recepción de información.

PC7266OUTPUT(): Es la encargada de enviar datos (números enteros) al puerto de salida de la tarjeta.

PC7266RESETCOUNTER(): Esta función pone a cero el codificador en la posición actual. En este trabajo se llama a esta función cuando se quiere "resetear" el codificador.

PC7266SETMULTIPLIER(): Se encarga de asignar el multiplicador de la tarjeta. Puede ser 1, 2, 3 o 4. En nuestro caso para que se obtuviera la máxima resolución se seleccionó el multiplicador 4.

Una vez realizada esta inicialización, el siguiente paso para iniciar la calibración es dar clic al botón “Activar Camara” en la barra de herramientas, ver sección 4.2. El método que procesa este mensaje es: OnCamara() y su único objetivo es mostrar la caja de diálogo no modal de la figura 4.8. Todo el control de esta caja la maneja la clase CDlg_Rec_y_Med(). En este momento las imágenes enviadas por la cámara son controladas por el subproceso cuyos mensajes son recibidos por la función OnBackgroundNotify(), que fue explicada en el programa Calibración de Camara 2.

En dicha caja de dialogo se tienen los botones correspondientes para iniciar la medición. Se había mencionado que antes de iniciar una medición eran necesario eliminar la iluminación y asignar valores a los umbrales, los botones Iluminación e Inicializar de la caja de dialogo permiten hacerlo mediante los métodos OnIluminacion() y OnInicializar() respectivamente. El método OnIluminacion() toma una imagen “limpia” (sin barras) de la regla a medir y la guarda para ser compensada posteriormente en el reconocimiento.

El código en general del método OnInicializar() que realiza el proceso explicado en el subtema 4.2.4 se muestra en el siguiente código.

```
void CDlg_Rec_y_Med::OnInicializar()
{
    int i,ii,iii,Ubral_Aux=0,j;
    int tolerancia1=15;//para la funcion F
    int tolerancia2=8;//para los umbrales
    int limiteInf=0,limiteSup=0,numero_barras=0,flat=0;
    double coor_max_y[2],aux2,coor_ultima=0;
    char cadena[90];
    CString aux1;

    ifstream datos("D:\\Gabriel\\Programas en C++ y Visual C++\\Calibracion de
    Camara 2\\datos.txt",ios::in);
    Fondo.FReadFile(szFile);//Obtiene la imagen Fondo del Archivo correspondiente

    m_imagenRGB=GetImage();//Obtine una imagen del video

    m_imagenGris=VisBrightnessFromRGBA(m_imagenRGB);//La convierte a escala de grises

    for( i=0;i<320;i++)//Se Quita el fondo a la imagen obtenida para
        for( j=0;j<240;j++)//obtener informacion "suavisada"
            ImagenGris_Uni(i,j)=abs(Fondo(i,j)-
            m_imagenGris(i,j));
    //////////////////////////////////////
    //////////////////////////////////////
    m_coor_primera_x=10000.0;
    coor_max_y[1]=0.0;
}
```

```

j=0;
datos.getline(cadena,90);//Se pone al principio por la forma de operar del while
while(datos){//Obtiene el punto x, mas a la izquierda
    pl=0;
    for(i=0;i<=4;i++){//obtiene la quinta cadena de la linea
        aux1=cadena;
        aux2=Subcadenas(aux1);
    }
    if(m_coor_ultima_x<aux2){
        m_coor_ultima_x=Redondeo(aux2);
    }

    if(m_coor_primera_x>aux2)
        m_coor_primera_x=Redondeo(aux2);
    datos.getline(cadena,90);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

CString aux_n;
UpdateData(true);

m_dato_res_mm=m_edit_str_resolucion.Mid(0,m_edit_str_resolucion.Find("/",0));
divisiones=strtol(m_edit_str_divisiones,NULL,10);
if(m_flat_umbrales==0){
    m_arreglo_barras=new int*[divisiones];//Se inicializa el tamaño de
los vectores
    m_umbrales=new int[divisiones];
    for (i=0; i<divisiones; i++){//Reserva memoria para cada renglon
        m_arreglo_barras[i] = new int[2];
        m_umbrales[i]=0;//Se inicializan los umbrales a cero
    }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int flat2=0;

Correccion();//Corrige Distorsion

Reconocimiento();//Saca la funcion F(x) y F'(x)

for(i=m_coor_primera_x; i<m_coor_ultima_x; i++){//Recorre la funcion F, para
encontrar barras

    //Solo el área corregida
    Umbral_Aux=0;
    //Si una barra esta cortada la descarta CASO IZQUIEDO
    if(i>=m_coor_primera_x && i<=m_coor_primera_x+4)
        if(F[i]>tolerancia || F[i+1]>tolerancia ||
F[i+2]>tolerancia)
            while(F[i]>tolerancia) i++;

    if(F[i]>tolerancia){//Encuentra el rango de la barra
        limiteInf=i;
        while(F[i]>tolerancia){

            if(i>=m_coor_ultima_x-3 && i<=m_coor_ultima_x){//Si una barra esta
cortada la descarta
CASO DERECHO
                if(F[i]>tolerancia || F[i+1]>tolerancia || F[i+2]>tolerancia){
                    flat=1;
                    break;
                }
            }
        }
    }
}

```

```

        i++;
    }
    if(flat==1) break;
    limiteSup=i-1;

    numero_barras++;
    for(ii=limiteInf,iii=1;ii<=limiteSup;ii++,iii++){//Saca el promedio
de la "barra"
        Umbral_Aux=Umbral_Aux+F[ii];
    }
    Umbral_Aux=Umbral_Aux/(iii-1);

    for(ii=0;ii<divisiones;ii++){
        //si entra en alguno de los umbrales ya encontrados
        //los promediará
        if(Umbral_Aux>=m_umbrales[ii]-tolerancia2 &&
            Umbral_Aux<=m_umbrales[ii]+tolerancia2){
            m_umbrales[ii]=(m_umbrales[ii]+Umbral_Aux)/2;
            break;
        }
        if(m_umbrales[ii]==0){//Caso en que no se haya inicializado
            //con algun umbral
            m_umbrales[ii]=Umbral_Aux;
            break;
        }
    }
    //fin del primer if
    m_flat_umbrales=1;//Umbrales inicializados
} //fin del for que recorre la imagen corregida

////////////////////////////////////
////////////////////////////////////

m_edit_str_umbrales.Empty();

Umbral_Aux=m_umbrales[0];//Acomoda los umbrales de mayor a menor
for(i=0;i<divisiones;i++){
    Umbral_Aux=m_umbrales[i];
    for(ii=i;ii<divisiones;ii++){
        if(Umbral_Aux<m_umbrales[ii]){
            Umbral_Aux=m_umbrales[ii];
            m_umbrales[ii]=m_umbrales[i];
            m_umbrales[i]=Umbral_Aux;
        }
    }
    aux1.Format("%d/%d ",i+1,m_umbrales[i]);//Los muestra en su
respectivo cuadro de dialogo
    m_edit_str_umbrales=m_edit_str_umbrales+aux1;
}
m_edit_num_barras=numero_barras;
UpdateData(FALSE);
datos.close();
}

```

Puede verse que este código está dividido en tres partes. La primera se encarga de eliminar la iluminación o fondo de la imagen simplemente restando una de otra. La segunda parte se encarga de corregir la imagen con la función Corrección() cuyo código es el mismo utilizado en el programa Calibración de Camara 2 por el método Calibrar() de la

clase CCorrección. En esta misma sección se hace un pre-reconocimiento de las líneas para obtener sus tamaños, el método que lo realiza se llama Reconocimiento() y su código es el siguiente.

```
void CDlg_Rec_y_Med::Reconocimiento()
{
    int i,j,n;
    CString aux,aux1;
    fstream datos1("datos1.txt",ios::out);
    derivada_F("derivada_F.txt",ios::out);
    for(i=0;i<320;i++)
    {
        n=0;
        for(j=239;j>=0;j--)
        {
            if(Imagen_Corregida2.Pixel(i,j)>=50)
            {
                F[i]++; //CADA VEZ SE AUMENTA UNO
            }
        }
        aux.Format("%d %f",i,F[i]); //SE MANDA LA FUNCION F A UN
ARCHIVO DE TEXTO
        aux=aux+"\n";
        datos1.write(aux,aux.GetLength());
    } //PRIMERA DERIVADA
    dF[319]=F[319]-F[318];
    for( i=1,j=0;i<=319;i++,j++)
    {
        dF[j]=F[i-1]-F[i];
        aux.Format("%d %f",i,dF[j]);
    } //SE MANDA LA FUNCION dF A UN ARCHIVO DE TEXTO
    aux=aux+"\n";
    derivada_F.write(aux,aux.GetLength());
}
}
```

La información es guardada en las variables $F[i]$ y $dF[j]$, donde $F[i]$ es la función de las barras y $dF[j]$ es su derivada. Estas variables son utilizadas en la tercera parte del método OnInicializar(), donde, por operaciones algebraicas, son delimitados los tamaños y anchos de las barras y son asignados los umbrales correspondientes.

Por último el botón Medicion(), comenzará el proceso automático de calibración. Mostramos el código del método OnMedición() encargado de dicha acción y posteriormente se explicará con detalle:

```
void CDlg_Rec_y_Med::OnMedicion()
{
    CClientDC dc(this);
    CString aux1;
    UpdateData(true);
    //INICIA TIMER DE ALTA VELOCIDAD Y CON ELLO EL PROCESO
    if(m_flat_timer==0)
    {
        m_pView->SendMessage(WM_MENSAJE_1, IDOK);
        m_flat_timer=1;
        return;
    }
    SetTimer(3,500,NULL);
    while(GetMessage(&msg,NULL,NULL,NULL))
```

```

{
    if(msg.message==WM_TIMER)
    {
        if(msg.wParam==3)
        {
            KillTimer(3);
            break;
        }
    }
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
if(m_edit_str_resolucion.IsEmpty()||
m_edit_str_lecturas.IsEmpty())
{
    MessageBox("Por favor, ingrese los Datos de Entrada
completos", "Faltan datos", MB_ICONINFORMATION);
    return;
}
Fondo.FReadFile(szFile);
//OBTIENE LA IMAGEN FONDO DEL ARCHIVO CORRESPONDIENTE
m_imagenRGB=GetImage();
//OBTIENE UNA IMAGEN DEL VIDEO
m_imagenGris=VisBrightnessFromRGBA(m_imagenRGB);
//LA CONVIERTE A ESCALA DE GRISES
for(int i=0;i<320;i++)
//SE QUITA EL FONDO A LA IMAGEN OBTENIDA PARA
for(int j=0;j<240;j++)
    ImagenGris_Uni(i,j)=abs(Fondo(i,j)-m_imagenGris(i,j));
Correccion();
//CORRIGE DISTORCION
Reconocimiento();//SACA LA FUNCION F(X) Y F'(X)
Medicion();//COMIENZA LA MEDICION DE LAS BARRAS ENCONTRADAS
}

```

Este método será llamado toda vez que el sistema de visualización esté posicionado en el lugar correcto para hacer la medición. Al inicio del código se puede observar que es enviado un mensaje, el cual es recibido por la clase CMCREGView donde se inicia un timer de alta velocidad (resolución de hasta 1 milisegundo) y con ello el proceso de automatización.

El método que recibe el mensaje es Med_Auto(), el cual al iniciar el timer de alta velocidad, comienza el muestreo necesario para el controlador digital. El código para configurar e inicializar el timer es el siguiente:

```

extern "C"void CALLBACK internalTimerProc(UINT id, UINT msg, DWORD dwUser,
    DWORD dw1, DWORD dw2)
{
    CMCREGView *timer=(CMCREGView *)dwUser;
    timer->timerProc();
}

void CMCREGView::Med_Auto()
{
    ////////////////////////////////////////
    MMRESULT result;
    period=1;
    timerRes=1;
}

```

```

result=timeSetEvent(period, timerRes,internalTimerProc,
(DWORD)this, TIME_PERIODIC);
if(NULL!=result)
{
    timerID=(UINT)result;
    timerInicio=timeGetTime();
}
}

```

donde:

internalTimerProc(): Es la función encargada de recibir un mensaje cada vez que se cumpla el evento de un periodo realizado.

TimeSetEvent(): Se encarga de inicializar el timer con los parámetros enviados, esto es, la resolución (timerRes) será de un milisegundo y habrá un evento cada periodo.

Puede observarse que cada vez que se ejecute un periodo, el proceso llamará a la función timerProc(), en donde se incluye el código del controlador digital. Dicha función se muestra a continuación:

```

void CMCREGView::timerProc()
{
    int salida;
    double out,kp,kd,ki;
    kp=20.0;
    kd=10000.0;
    ki=0.0;
    teta=(double)PC7266READPOSITION(0)/500;
    r=setPoint;
    m=((kp+ki+kd)*(r-teta))-((kp+(2*kd))*(r1-teta1))+((kd)*(r2
-teta2))+m1;
    if(m>10.4)
    {
        v=10.4;
    }
    else if(m<-10.2)
    {
        v=-10.2;
    }
    else v=m;
    out=(12.3*v)+128.0;
    salida=(int)out;
    m_posicion=teta;
    errortipo = PC7266OUTPUT(salida);
    if (errortipo !=0) AfxMessageBox("Error al escribir en el
puerto", MB_ICONSTOP);
    m2=m1;
    m1=m;
    v2=v1;
    v1=v;
    r2=r1;
    r1=r;
    teta2=teta1;
    teta=teta;
    ////////////////////////////////////////
    if (abs(teta-r)<0.01 && m_pDlg_Rec_y_Med->m_flat_done==0)
    {
        m_pDlg_Rec_y_Med->m_flat_done=1;
        SetTimer(20,300,NULL);
    }
}
//RETARDO

```

```

}
}

```

donde:

PC7266READPOSITION(): Obtiene la posición actual del codificador.

PC7266OUTPUT(): Manda los datos a la tarjeta para el movimiento de los motores.

En este código la ecuación 1.1 es implementada. La variable teta contiene el valor actual del codificador. El SetPoint (variable r), es obtenido y actualizado según los incrementos introducidos al inicio del programa.

Al aplicarse la ecuación, el resultado m, es delimitado entre el valor 10.4 y 10.2, esto con el fin de mandar la información correcta a los motores y no sobrepasar el voltaje permitido de estos.

Cuando se llegue a un valor considerado como aceptable (muy cercano al setpoint), el controlador es detenido por medio del timer de alta velocidad y continua el proceso de medición.

Regresando a la función OnMedicion(), puede verse al inicio de ésta un retardo de 500 milisegundos, esto con el fin de estabilizar el sistema y obtener una imagen adecuada. Posteriormente es eliminado el fondo de la imagen, se corrige y se reconoce por las funciones vistas en el código anterior.

El siguiente método que llama la función OnMedicion() es Medicion(), la cuál se encarga de obtener el error encontrado en la regla graduada con el código siguiente:

```

void CDlg_Rec_y_Med::Medicion()
{
    int i,ii,iii,Umbrales Aux=0,j,ini=0;
    int tolerancia1=10;//para la funcion F
    int tolerancia2=5;//para los umbrales
    int limiteInf=0,limiteSup=0,barra_medir,punto_medio=0,aux=0;

    float max[2],min[2];

    UpdateData(true);

    m_res_pixeles=0;

    barra_medir=strtol(m_edit_str_lecturas,NULL,10)-1;

    if(barra_medir+1>strtol(m_edit_str_divisiones,NULL,10) ||
       barra_medir<0) MessageBox("Introduzca un número entre 1 y el No. de Divisiones
Diferentes","Fuera de Rango");

    //ENCUENTRA LA RESOLUCION EN PÍXELES LOCAL
    for(i=0; i<320; i++){//Recorre la función F, para encontrar la barra a
medir

```

```

Umbral Aux=0;
//Si una barra esta cortada la descarta CASO IZQUIEDO
if(i>=m_coor_primera_x && i<=m_coor_primera_x+4)
    if(F[i]>tolerancial || F[i+1]>tolerancial ||
F[i+2]>tolerancial)
        while(F[i]>tolerancial) i++;

if(F[i]>tolerancial){//Encuentra el rango de la barra
    limiteInf=i;
    while(F[i]>tolerancial) i++;
    limiteSup=i-1;

    //Encuentra el Máximo de esta área
    max[0]=dF[limiteInf-1]; //Valor
    max[1]=0; //Indice
    for(j=limiteInf-1;j<=limiteSup;j++){
        if(dF[j]>max[0]){
            max[0]=dF[j];
            max[1]=j;
        }
    }

    //Encuentra el Mínimo de esta área
    min[0]=dF[limiteInf-1];//Valor
    min[1]=0; //Indice
    for(j=limiteInf-1;j<=limiteSup;j++){
        if(dF[j]<=min[0]){
            min[0]=dF[j];
            min[1]=j;
        }
    }

    punto_medio=(max[1]+min[1])/2;
    aux=abs(aux-punto_medio);//Distancia entre barras

    if(ini==1){
        m_res_pixeles=aux;
        ini++;
    }
    if(ini==0){//para primer punto, por lo tanto distancia 0
        m_res_pixeles=0;
        aux=punto_medio;
        ini++;
    }
    m_res_pixeles=(m_res_pixeles+aux)/2;
    aux=punto_medio;
}
} //fin del For

int res=Redondeo(m_res_pixeles);//Actualiza valores en la Interfaz
CString aux_n;
aux_n.Format("%d",res);
m_dato_res_mm=m_edit_str_resolucion.Mid(0,m_edit_str_resolucion.Find("/",0));

m_edit_str_resolucion=m_dato_res_mm+"/"+aux_n;

int flat=0;
int error_aux=321;//Solo se inicializa con un valor grande
////////////////////////////////////
////////////////////////////////////

for(i=0; i<320; i++){//Recorre la funcion F, para encontrar la barra a
medir

```

```

    Umbral_Aux=0;
    //Si una barra esta cortada la descarta CASO IZQUIEDO
    if(i>=m_coor_primer_x && i<=m_coor_primer_x+4)
        if(F[i]>tolerancial || F[i+1]>tolerancial ||
F[i+2]>tolerancial)
            while(F[i]>tolerancial) i++;

    if(F[i]>tolerancial){//Encuentra el rango de la barra
        limiteInf=i;
        while(F[i]>tolerancial){

            if(i>=m_coor_ultima_x-3 && i<=m_coor_ultima_x){//Si una barra
esta cortada la descarta CASO DERECHO
                if(F[i]>tolerancial || F[i+1]>tolerancial ||
F[i+2]>tolerancial){
                    flat=1;
                    break;
                }
            }
            i++;
        }
        if(flat==1) break;
        limiteSup=i-1;

        for(ii=limiteInf,iii=1;ii<=limiteSup;ii++,iii++){//Saca el
promedio de la "barra"
            Umbral_Aux=Umbral_Aux+F[ii];
        }
        Umbral_Aux=Umbral_Aux/(iii-1);
        if(Umbral_Aux>=m_umbrales[barra_medir]-tolerancia2){
            //Encuentra el Máximo de esta área
            max[0]=dF[limiteInf-1]; //Valor
            max[1]=0; //Indice
            for(j=limiteInf-1;j<=limiteSup;j++){
                if(dF[j]>max[0]){
                    max[0]=dF[j];
                    max[1]=j;
                }
            }
            //Encuentra el Mínimo de esta área
            min[0]=dF[limiteInf-1]; //Valor
            min[1]=0; //Indice
            for(j=limiteInf-1;j<=limiteSup;j++){
                if(dF[j]<=min[0]){
                    min[0]=dF[j];
                    min[1]=j;
                }
            }
            punto_medio=(max[1]+min[1])/2;
        }
    }
    if(abs(159-punto_medio)<error_aux){
        error_aux=abs(punto_medio-159);
        m_edit_str_error_pixeles=punto_medio-159;
    }
} //Fin de for que recorre toda la funcion F

if(m_edit_str_error_pixeles<0) Detalles_Graficos(m_edit_str_error_pixeles+159,3);
else
    Detalles_Graficos(m_edit_str_error_pixeles+159,2);

m_edit_str_error_mm=m_edit_str_error_pixeles*m_res_mm/m_res_pixeles;

```

```
UpdateData(FALSE); //Muestra los errores en pantalla  
}
```

Básicamente este código está dividido en dos partes. La primera parte se encarga de encontrar la distancia que hay entre las barras, esto con la finalidad de encontrar la relación entre la resolución en pixeles (distancia entre barra y barra) y la resolución en milímetros (dato obtenido inicialmente). Cabe señalar que este código eliminará las barras que se encuentren en los extremos de la imagen, ya que no pueden ser consideradas como buenas para la obtener una distancia entre barras ni para obtener una medición.

La segunda parte se encarga de reconocer las barras, tomar la medida entre un borde y el centro de la imagen, mandar los resultados y graficarlos.

Al final se obtiene el valor solicitado, esto es, el error que existe entre la barra y el valor del codificador, recordando que este último es utilizado como patrón.

Este método será llamado tantas veces sea necesario para completar las tres series solicitadas. Al final los datos aparecerán en la pantalla de datos con su respectiva gráfica.

Capítulo 5

Resultados y conclusiones

En el capítulo tres se describió el sistema propuesto para la calibración de reglas utilizando visión por computadora, lo cual proporciona una idea cualitativa de su desempeño. En el presente capítulo describimos resultados cuantitativos obtenidos en las diversas etapas del proyecto, considerando cada etapa de forma independiente. No obstante, el desempeño global del sistema es evaluado al realizar una calibración de una regla propiedad del Laboratorio de Metrología utilizando el sistema propuesto en este proyecto de tesis. El desempeño global entonces contempla los resultados parciales en las etapas independientes para realizar una medición con gran exactitud y alto grado de automatización.

Se presentan también las conclusiones finales y una serie de propuestas que pueden orientar el trabajo a futuro con el objetivo de realizar mejoras al instrumento.

5.1. Resultados

Para la obtención de resultados cualitativos consideramos tres subprocesos que influyen directamente en el desempeño del sistema completo: calibración de cámara, extracción de la medición y control de posición. Cada uno de ellos es evaluado aisladamente de acuerdo a ciertos objetivos propuestos. Posteriormente, en la operación en tiempo real se verifica el desempeño del instrumento en su conjunto, es decir, incluyendo los resultados parciales que aseguran la confiabilidad de la medición.

5.1.1. Calibración de cámara

En el sentido de obtener resultados cualitativos, desarrollamos dos experimentos para probar que una escena ideal aparece en el plano imagen corregido con el menor grado de distorsión. En este sentido, una malla patrón conformada por líneas rectas y espaciamiento uniforme, debe aparecer como una imagen de líneas rectas y espaciamiento uniforme.

En el primer experimento, ajustamos características sin distorsión a líneas rectas horizontales y verticales y calculamos la suma de los cuadrados de los errores residuales. Después, comparamos el resultado contra aquel obtenido para las características con distorsión. Los resultados se sintetizan en la tabla 5.1.

	Sin distorsión (píxeles ²)	Con distorsión (píxeles ²)
Horizontal	6.4324X10-16	7.70127
Vertical	6.9364X10-16	8.98562

Tabla 5.1. Suma de los cuadrados de los errores residuales.

Los resultados de la tabla 5.1 muestran que las imágenes sin distorsión representan mejor las líneas rectas de la escena.

En el segundo experimento comprobamos la uniformidad para la distancia de separación entre líneas verticales y horizontales para la proyección de la malla patrón en el plano imagen con distorsión y sin distorsión. En este experimento, las características son ajustadas a líneas verticales y horizontales para posteriormente estimar la uniformidad entre sus separaciones. El criterio para verificar la uniformidad lo desarrollamos en dos partes: paralelismo entre rectas y uniformidad en la separación entre ordenadas al origen. El paralelismo entre rectas se verifica al graficar las pendientes de las rectas horizontales y verticales, como se muestra en la figura 5.1.

En la figura 5.1 se puede apreciar que las pendientes en las rectas sin distorsión se mantienen iguales, lo que indica que nunca se cruzan y que mantienen una separación constante o paralela. Por lo contrario, las pendientes de las rectas con distorsión varían indicando que existen cruces entre ellas y que su separación no es uniforme.

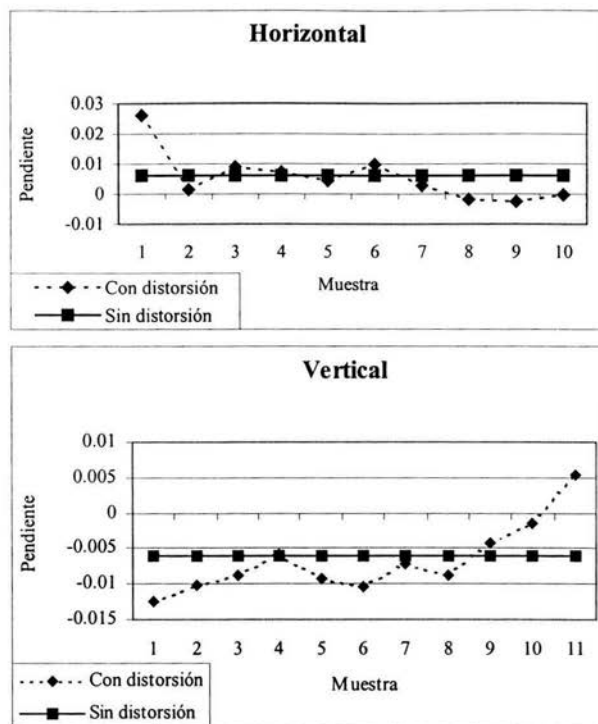


Figura 5.1. Paralelismo.

Por otra parte, la separación entre rectas es evaluada al graficar la diferencia entre ordenadas al origen para rectas adyacentes. Las gráficas se muestran en la figura 5.2. La figura muestra que existe una separación uniforme entre rectas sin distorsión horizontales (23,2342 pixeles) y verticales (23,2342 pixeles). Por lo contrario, las rectas que ajustan características con distorsión no mantienen uniformidad en la separación.

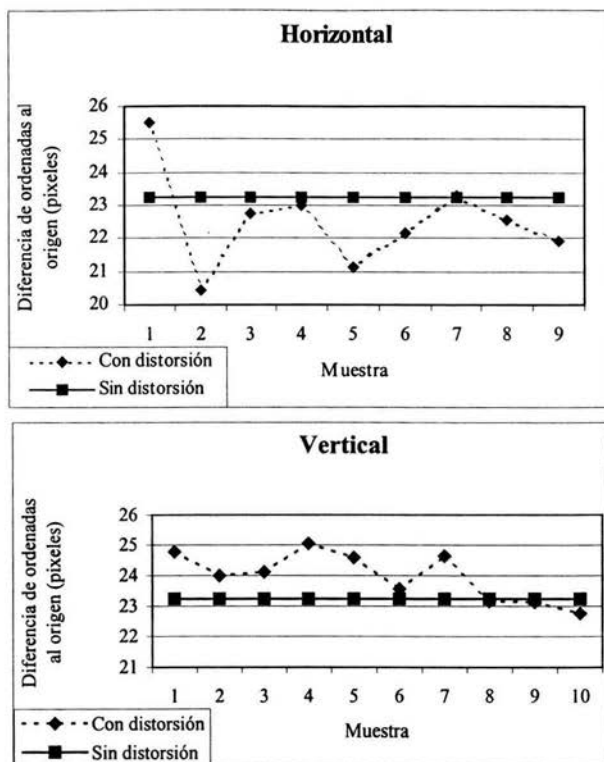


Figura 5.2. Uniformidad en la separación.

Los resultados anteriores indican que, corrigiendo la distorsión, la unidad de longitud proyectada en el plano imagen resulta en un número de píxeles determinado sin importar su orientación (vertical u horizontal) o su posición relativa al centro óptico de la imagen. En la práctica, la idea anterior es de gran utilidad ya que la relación que existe entre unidades reales y unidades proyectadas puede ser estimada y realizar mediciones con gran exactitud.

Un aspecto cualitativo en la corrección de la distorsión en imágenes puede apreciarse en las figuras 5.3 y 5.4. La figura 5.3 muestra una malla patrón corregida con el software desarrollado. La figura 5.4 muestra un objeto común antes y después de la corrección. En ambas figuras, en la parte izquierda se muestra la imagen original con distorsión tal cual es capturada por el sistema de visión. En la parte derecha de las figuras aparece la imagen corregida utilizando el procedimiento descrito en la sección 3.2 y cuya implementación se describió en las secciones 4.2.1, 4.2.2 y 4.3.1.

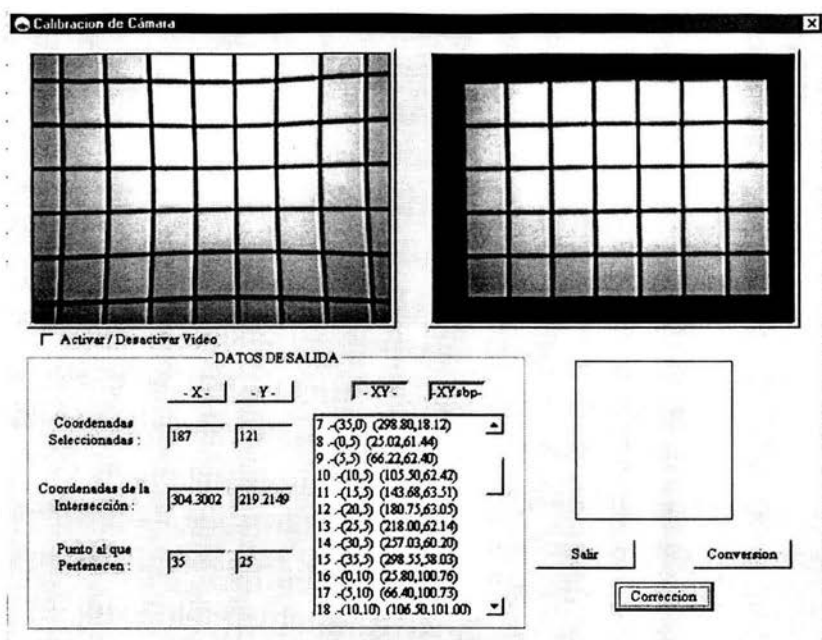


Figura 5.3. Ejemplo de la calibración de cámara para una rejilla.

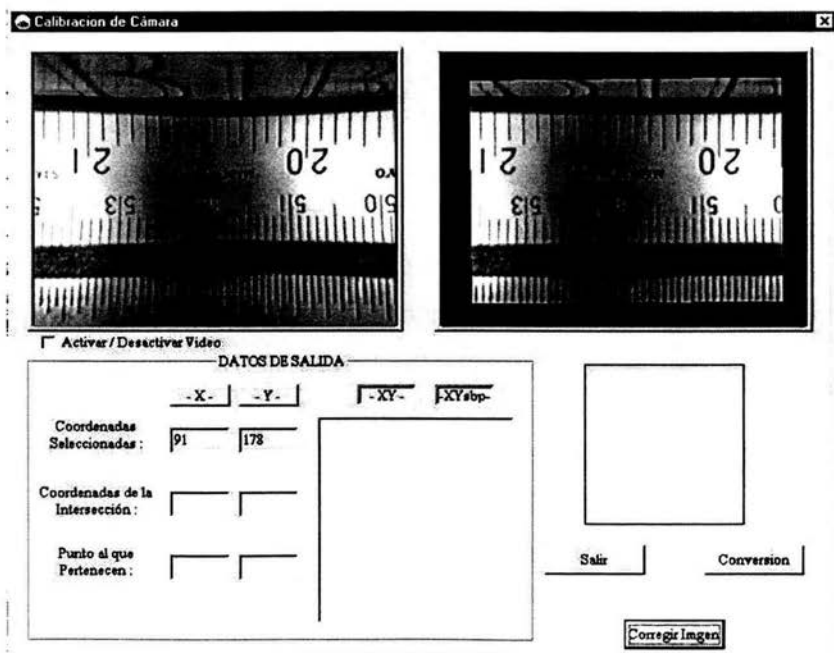


Figura 5.4. Ejemplo de la calibración de cámara para un objeto común.

5.1.2. Extracción de la medición

En la evaluación de la extracción de la medición consideramos dos aspectos fundamentales: El análisis de la imagen y la confiabilidad de la medición.

El análisis de la imagen es vital para el cálculo de la desviación y el reconocimiento de los diferentes tipos de marcas impresas sobre una escala graduada. Para verificar el desempeño cualitativo se realizó el siguiente experimento. Se obtuvieron mediciones de desviaciones de forma manual y automática a lo largo de una regla de 600 mm de largo, como es mostrado en la técnica para la extracción de la medición descrita en la sección 3.3 e ilustrada en la figura 3.12. En la forma manual un operador se encargó de realizar la medición de la desviación directamente de la imagen obtenida haciendo uso de sus propias habilidades y experiencia. En la forma automática el software se encargó de extraer la medición de la desviación.

Las figuras 5.5 a 5.7 muestran la extracción de la medición de forma automática para tres tipos diferentes de marcas en una escala graduada. Estas imágenes fueron utilizadas por el operador para obtener la medición. La tabla 5.2 resume los resultados de la medición de la desviación obtenidos.

	Manual [mm]	Automática [mm]
Barra del tipo 1	1.4444	1.5106
Barra del tipo 2	1.3500	1.4316
Barra del tipo 3	-0.4444	-0.4278

Tabla 5.2. Extracción de la medición (cálculo de la desviación).

Aclaremos que las desviaciones obtenidas son exageradas y nunca se presentan en una escala, su único fin fue el de comparar ambos resultados.

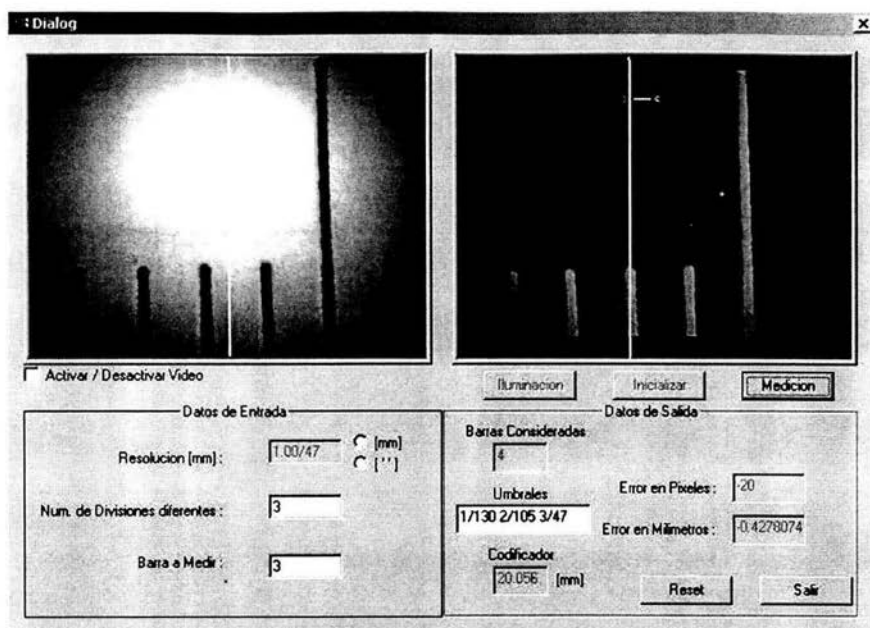


Figura 5.5. Extracción de la medición en forma automática (barra del tipo 3).

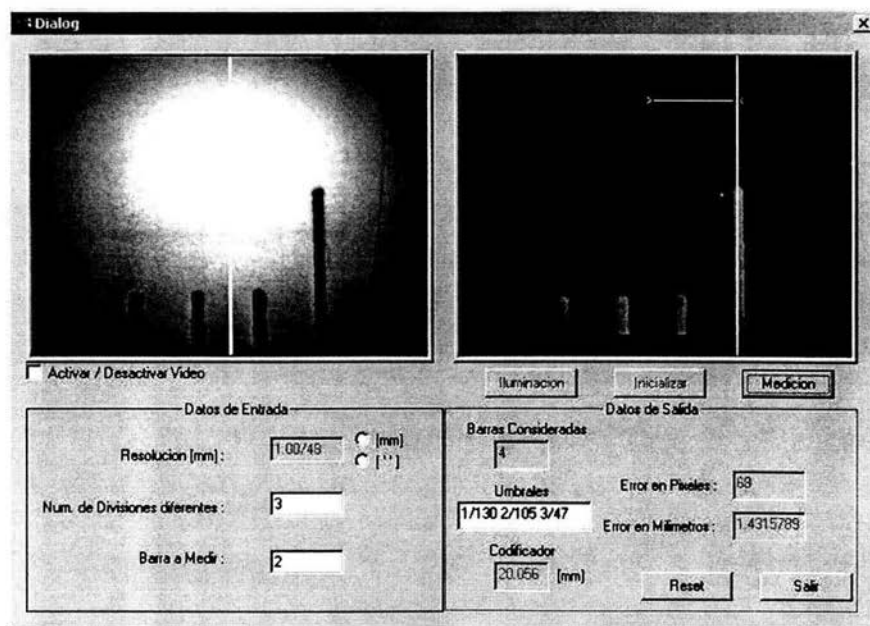


Figura 5.6. Extracción de la medición en forma automática (barra del tipo 2).

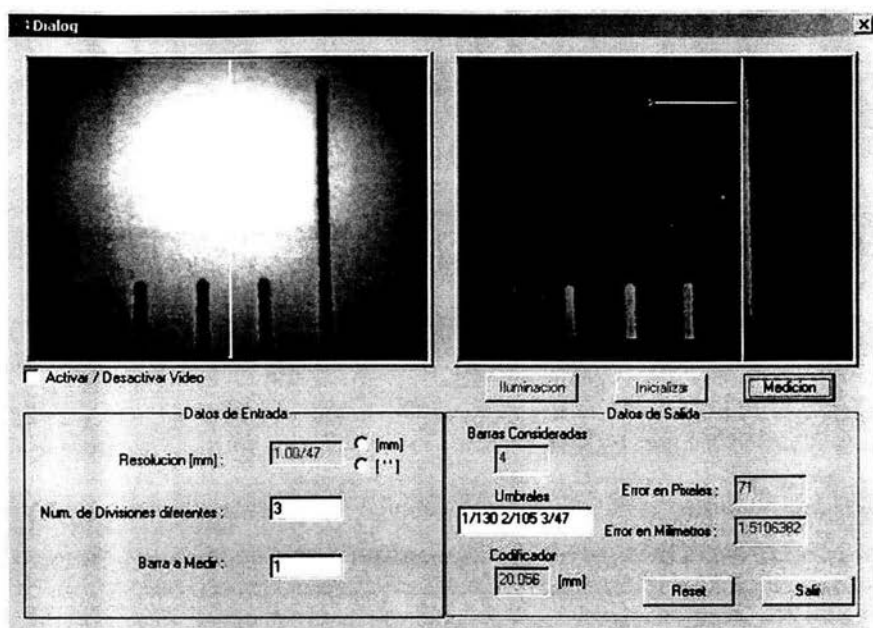


Figura 5.7. Extracción de la medición en forma automática (barra del tipo 1).

Para evaluar la confiabilidad del reconocimiento, se realizó la medición de una escala graduada de 600mm en alcances e intervalos de medición establecidos, consistente de tres series de datos. La escala se mostraba uniforme, es decir, carecía de daños locales y artefactos que obstaculizaran el procesamiento de la imagen. Posteriormente se totalizaron los aciertos alcanzados en el reconocimiento preciso usando el método automático. La tabla 5.3 resume los resultados

Tipo de barra	Intervalo de medición	Alcance de medición	Número de aciertos / Total de eventos	Porcentaje de acierto %
1	10 mm	200 mm	58 / 60	96
2	5 mm	50 mm	30 / 30	100
3	1 mm	50 mm	150 / 150	100

Tabla 5.3. Extracción de la medición (confiabilidad).

Cabe destacar que las marcas de la escala que permanecen en los límites de la imagen no son consideradas para la medición, logrando con todo ello un sistema confiable en cuanto al reconocimiento se refiere.

5.1.3. Control de posición

El experimento realizado sobre el control de posición fue el siguiente. Se realizó una medición de forma arbitraria y se observó la repetibilidad en el posicionamiento automático caracterizada por los datos de la figura 5.8. La gráfica fue obtenida al mover el dispositivo de observación tres veces de 0 a 600mm y verificar el error en la posición de origen.

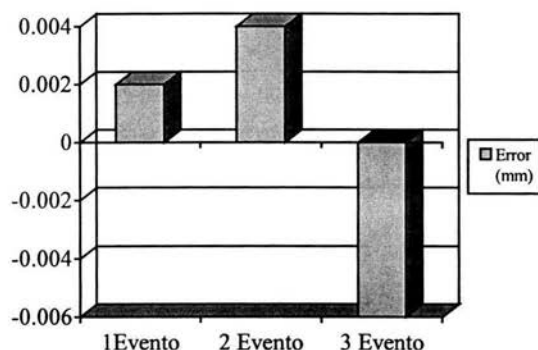


Figura 5.8. Error de posición.

5.1.4. Operación en tiempo real

Se realizó una medición a una regla graduada utilizada en el Laboratorio de Metrología con el sistema implementado. Los resultados pueden observarse en la figura 5.9. En esta figura podemos apreciar la calibración de la regla en su tramo de 0 a 40 cm., con una resolución de 0.5.

La medición de una regla incluye una hoja de datos numéricos y la curva de calibración que caracteriza al objeto en inspección. La medición fue realizada en forma automática. Solo se requiere la intervención humana en el proceso de alineamiento de la regla y en la definición del origen de la medición (puesta a cero).

Se puede apreciar que esta regla obtuvo un error máximo de 0.133 milímetros y un error mínimo de 0.103 mm a lo largo del tramo. El tiempo empleado en la medición, una vez realizados los alineamientos y puesta a cero, es aproximadamente de 5 min. El tiempo que toma la alineación y puesta a cero dependen de las habilidades del operador, pero típicamente demora 20 min.

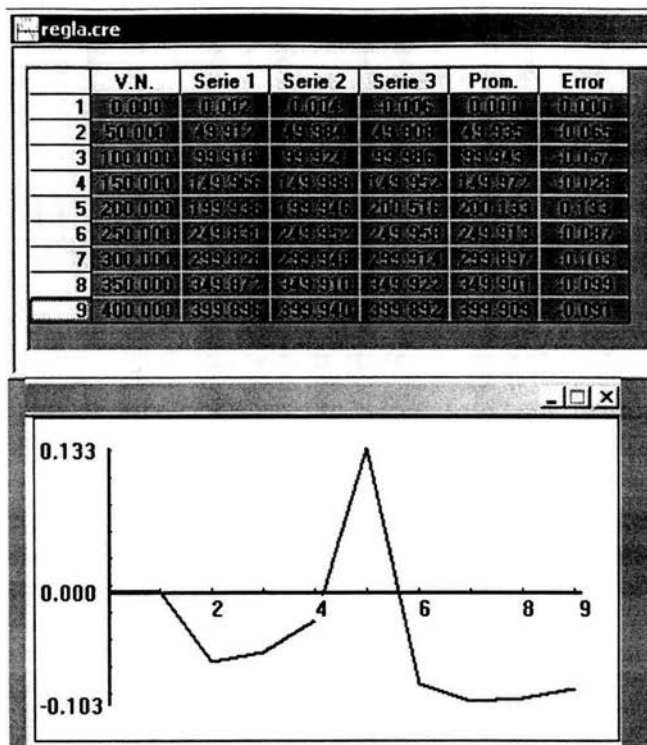


Figura 5.9. Resultados de la calibración de una regla graduada

5.2. Conclusiones

Se presentó un sistema de visión y posicionamiento automático para la calibración de escalas graduadas, figura 5.10. El principal problema enfrentado fue la reducción de la incertidumbre que pueden introducir los operadores humanos. La generación de un conjunto de características sin distorsión se consiguió utilizando técnicas de calibración de cámara. Entonces el algoritmo de medición genera imágenes casi ideales mediante un método para la compensación de la distorsión. La completa automatización del proceso se alcanza mediante las técnicas de extracción de la medición y el control automático de posición. Se desarrolló una serie de experimentos que nos indican la confiabilidad del enfoque adoptado. Para propósitos metrológicos, la exactitud del sistema fue verificada cuantitativa y cualitativamente.

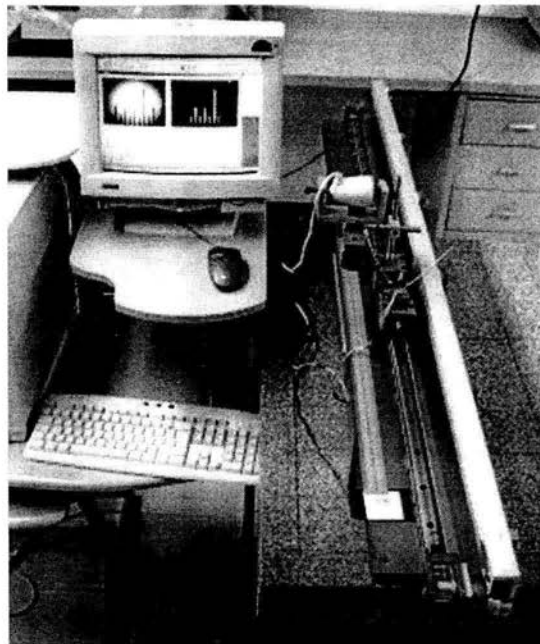


Figura 5.10. Máquina para la calibración de escalas graduadas utilizando visión por computadora y posicionamiento automático.

Según los resultados obtenidos podemos concluir lo siguiente:

- En nuestro caso, la corrección de la cámara reduce la incertidumbre de la medición. Si bien la distorsión no es tan notoria para el caso de la cámara acoplada al microscopio, las distorsiones mayores pueden ser corregidas notablemente como la presentada en la figura 5.4.
- La calibración de la cámara se realiza sólo una vez y se utiliza durante todo el proceso de medición, siempre y cuando la cámara o el sistema de visualización en general no sean modificados o sustituidos.
- En el caso de la extracción de la medición, la iluminación resultó parte fundamental para el reconocimiento, ya que si existieran variaciones de esta durante el proceso, las líneas podrían ser clasificadas erróneamente o simplemente no ser reconocidas.
- Las condiciones de la regla o escala a calibrar también son parte fundamental para el correcto reconocimiento, ya que de estar sucias, oxidadas o maltratadas en las partes donde se realizará la medición, podría incluirse información falsa y por ende arrojar resultados ilógicos e irreales.

- Debido al enfoque adoptado para la extracción de la medición presentado en la figura 3.12, se concluye que es posible obtener una calibración de cualquier regla o escala graduada siempre y cuando el error existente no sea mayor de $\frac{1}{2}$ de su resolución.
- En el caso del control PID, aunque existe un pequeño error de posición, para nuestro caso no resulta ser un problema considerablemente grave, ya que este error es absorbido por el algoritmo de extracción de características según lo explicado en la sección 3.3.
- El sistema trabajando en conjunto logra mediciones completas como la mostrada en la figura 5.9. Puede observarse que los errores máximo y mínimo son relativamente pequeños, este error podría ser aceptado en ciertos tipos de regla, sin embargo, según la tabla 1.1, el error sería aun muy grande para reglas de tipo II.

Se presentó un avance de este proyecto en el Congreso Iberoamericano de Reconocimiento de Patrones en La Habana, Cuba, en el mes de Noviembre de 2003, integrando las memorias del Congreso y la publicación en Lecture Notes in Computer Science 2905 [23].

Finalmente, el proyecto utilizó el financiamiento por parte del proyecto PAPIT IN11100 "Utilización de la visión estéreo para mediciones geométricas precisas sin contacto".

5.3. Trabajo a futuro

Una vez obtenidos los resultados anteriores, proponemos los siguientes trabajos a futuro que contribuyen en la mejora del sistema de medición.

En el caso de la calibración:

- Diseñar un sistema mecánico que permita resoluciones de posicionamiento cercanas a $1 \mu\text{m}$ o menores.

En el caso de la extracción de la medición:

- Eliminar el posible ruido que pudiera afectar a las imágenes utilizadas en la medición, como son, manchas en la regla, números, oxidación, etc.
- Aunque en una medición controlada en las condiciones de un laboratorio de metrología no es necesario, sería conveniente compensar por posibles variaciones en la iluminación que afecten la calidad de las imágenes.

- Implementar el reconocimiento con cámaras de mayor resolución para obtener más detalle en la imagen a analizar.

En el caso del control de posición:

- Mejorar el controlador del sistema, hasta lograr un posicionamiento micrométrico del dispositivo de visualización.

Para la operación en tiempo real:

- Mejorar el software desarrollado, haciendo más sencilla la instalación y configuración de los periféricos.

Bibliografía

- [1]. R. Amaya Méndez, *Interfaz electrónica para una máquina usada en la calibración de reglas y escalas graduadas de alta exactitud*, Ingeniería Eléctrica Electrónica, Facultad de Ingeniería, UNAM, Noviembre de 2000, Tesis de licenciatura, México D.F.
- [2]. Automation and Metrology Incorporated, ENC 250 linear encoder, http://www.auto-met.com/acurite/enc_250.htm, feb-2004
- [3]. A. Azar y P. Musé, Calibración de Cámaras digitales Proyecto de Fin De Carrera "ÑACURUTÚ", 1998-1999, <http://iie.fing.edu.uy/investigacion/grupos/gti/proyectos/calibracion/>, oct-2003
- [4]. J. Barroso, E. L. Dagless, A. Rafael and J. Bulas-Cruz, "Number plate reading using computer vision", *Proceedings of the IEEE International Symposium on Industrial Electronics*, 1997, pp 761-766.
- [5]. C. Cañero, F. Vilariño, J. Mauri and P. Radeva "Predictive (un)distortion model and 3D Reconstruction by Biplane Snakes", *IEEE Transactions on Medical Imaging*, vol. 21, num. 9, 2002, pp 1188-1201.
- [6]. F. J. Ceballos, *Microsoft Visual C++: Aplicaciones para Win32*, 2ª edición, Ra-Ma, Marzo 2003.
- [7]. D. Chapman, *Aprendiendo Visual C++ en 21 días*, Prentice-Hall, 1999.
- [8]. F. J. Cortijo Bon , *Patrones y Reconocimiento de Patrones*, Octubre de 2001,

- http://www.etsi2.ugr.es/depar/ccia/rf/www/tema1_00-01_www/node2.html, feb-2004
- [9]. C. Delgado, R. Nava, G. Ruiz, *Metrología Dimensional*, Centro de Instrumentos, UNAM.
- [10]. R. Díaz de Santos C. Gonzales, Richard E. Woods, *Tratamiento digital de imágenes*, Addison-Wesley, 1996.
- [11]. K. Harris, S. N. Efstratiadis, N. Maglaveras, J. Gourassas, C. Pappas and G. Louridas, "Coronary Arterial Tree Extraction based on Artery Tracking and Mathematical Morphology", *Computers in Cardiology*, 1988, pp 769-772.
- [12]. International Recommendation No. 35: *Material Measures of length for General Use*, OIML, 1985, pp 22.
- [13]. D. J. Kruglinski, *Progrese con Visual C++*, McGrawHill, 2000.
- [14]. MinEco, *La Metrología y el Centro Nacional de Metrología en Guatemala*, 06-Feb-2004, <http://www.mineco.gob.gt/mineco/calidad/suplemento.pdf>, 1-Mar-2004.
- [15]. K. Ogata, *Sistemas de Control en Tiempo Discreto*, Pearson Educación, 1996.
- [16]. C. H. Pappas, William H. Murray, *Visual C++ 6.0 Manual de Referencia*, McGrawHill, 1999.
- [17]. D. F. Rogers, J. Alan Adams, *Mathematical Elements for Computer Graphics*, McGraw-Hill, 1990.
- [18]. J. Tang, "PID Controller Using the TMS320C31 DSK with On-line Parameter Adjustment for Real-time DC Motor speed and Position Control", *Proceedings of the IEEE International Symposium on Industrial Electronics*, 2001, pp 786-791.
- [19]. R. Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses", *IEEE Journal of Robotics and Automation*, vol. 3, num. 4, 1987, pp 323-344.
- [20]. US Digital Co., PC7266 PC to Incremental Encoder Interface Card, Tuesday June 24 2003 , [http://www.usdigital.com/products/pc7266/.](http://www.usdigital.com/products/pc7266/), feb-2004
- [21]. B. Valera, O. S. Padilla, "Implementación de un controlador PID digital", *Serie: Implementación de un controlador PID digital: Informe técnico*, Octubre de 2002, CCADET UNAM, pp 50.

- [22]. B. Valera, O. S. Padilla, B. G. Ruiz, "Implementación electrónica para una máquina usada en la calibración de escalas graduadas de alta exactitud", *Serie: MET 0003: Informe técnico*, Noviembre de 2001, CCADET UNAM, pp 12.
- [23]. B. Valera, S. Padilla, G. Chávez y V. García, 2003, "Graduated Scale Inspection Using Computer Vision", *8th Iberoamerican Congress on Pattern Recognition CIARP 2003*, Lecture Notes in Computer Science 2905, International Association for Pattern Recognition IARP, La Habana Cuba, 26-29 de Noviembre de 2003, pp 29-36.
- [24]. Vision Technology Group, Microsoft Vision SDK, Version 1.2, MSDN, May 2000,
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvisssdk/html/visssdk.asp>, Agosto 2003.
- [25]. J. Weng, P. Cohen and M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, num. 10, 1992, pp 965-980.
- [26]. G. Woldberg, *Digital Image Warping*, Wiley, NJ; 2002.