



Centro de Investigación en Matemáticas, A.C.

**Medición Robusta de Líneas
Graduadas Utilizando un
Sistema de Visión**

Tesis

que para obtener el grado de

Maestro en Ciencias

con especialidad en

Computación y Matemáticas Industriales

presenta

José Antonio Muñoz Gómez

Director de Tesis

Dr. Mariano José Juan Rivera Meraz

Guanajuato, Gto., Junio del 2001

Carro 9
25508

Agradecimientos

Quiero agradecer al Dr. Mariano José Juan Rivera Meraz por su paciencia y atención para el desarrollo de esta tesis. En gran medida sus conocimientos, consejos y comentarios aportaron el conocimiento para que pudiera concluir el presente trabajo de tesis y mis estudios.

También quisiera agradecer a todos los profesores del área de cómputo del CIMAT, al Dr. Johan Van Horebeek, Dr. Carlos Montes de Oca, Profesora Stephanie Dunbar y especialmente al Dr. J. Luis Marroquín cuyo curso impartido brindo la motivación hacia el estudio de sistemas de visión artificial.

Quisiera dar también un agradecimiento a mis compañeros de estudios dentro de esta institución, Edmar Mota, Pedro Elizarraraz, Jesús Navarro, Justino Ramírez y en especial a Flavio Viguera, que con sus consejos y ayuda fue posible discutir y entender las materias impartidas durante los estudios de la maestría.

En cuanto a la motivación para ingresar a la maestría, quiero agradecer al Dr. Domingo Vera Mendoza por la constante motivación y apoyo para tomar la decisión de estudiar un posgrado.

Agradezco también al Consejo Nacional de Ciencia y Tecnología CONACyT, por el apoyo económico otorgado a lo largo de estos dos años. Gracias a la beca recibida fue posible dedicarme de tiempo completo al estudio.

Finalmente quisiera dar mi gratitud a todo el personal del CIMAT, doctores, profesores, personal administrativo y demás trabajadores que permitieron con su amabilidad y disponibilidad ayudar, en lo que fue posible, a que el transcurrir estos años fuera agradable y propicio para el estudio.

RESUMEN

La medición de reglas patrón es una actividad de gran importancia a nivel mundial. En base a estas reglas se calibran otros sistemas longitudinales de menor precisión. En los laboratorios primarios de los diversos países se cuenta con departamentos especializados cuya función es medir con alta exactitud (del orden de micras) la distancia entre líneas de las reglas graduadas, en el caso de México esta tarea es asignada al Centro Nacional de Metrología (*CENAM*). La calibración de dichas reglas es muy requerida por la industria nacional, debido a que deben de estar certificados bajo una norma internacional para que sus productos manufacturados puedan competir con los estándares internacionales. A nivel mundial se tienen varios métodos automáticos para la medición, sin embargo dichos métodos consideran la regla como un arreglo unidimensional 1-D ocasionando que las mediciones se apoyen en pocos puntos. Actualmente el *CENAM* realiza la medición de manera semi-automática y sobre una muestra de la regla, teniendo con ello que la precisión detectada se vea influenciada negativamente por el error humano y no se logra el objetivo principal que es la medición de cada línea graduada. Las reglas varían en material y tamaño, las más frecuentes son de acero y acero inoxidable cuyas longitud máxima es de 3m, también se cuentan con reglas de cristal cuyas longitud máxima es de 300mm.

En el presente trabajo se presenta un nuevo método fuera de línea para determinar la distancia entre líneas graduadas. El proceso de medición consiste en determinar con alta precisión, del orden de $10^{-6}m$, la distancia entre líneas graduadas de una regla patrón. Para tal efecto se utiliza un sistema de visión (*SV*) que consiste en un microscopio y una cámara CCD con una resolución de 640×480 pixels, la posición en la que se encuentra el *SV* es determinada por un interferómetro HP con una resolución de $5 \times 10^{-9}m$. El *SV* es desplazado encima del objeto a medir obteniendo un vídeo y un archivo con los datos del sensor (*Z*) que indica la distancia a la que se encontraba el *SV* al momento de capturar la imagen. El trabajo se concentra en dos partes: La primera consiste en determinar de forma robusta la ecuación de la recta que mejor se ajusta al centro de cada marca presente en la imagen; para ello se utiliza un filtro de Gabor para detectar el centro de la marca por renglón con precisión subpixel y con base a dichos puntos se realiza un ajuste de regresión lineal robusto empleando un estimador-*M*. La segunda parte corresponde a mapear los datos locales obtenidos a coordenadas del interferómetro y determinar con ello la distancia a la que se detectó cada línea procesada; esto se realiza encontrando el factor de calibración β que indica cuántas micras son equivalentes a un pixel-columna de la imagen. Con base al factor β y la coordenada del sensor *Z*, el centro de la marca es mapeado a coordenadas globales obteniendo así la distancia nominal a la que se encuentran las líneas procesadas.

El método desarrollado arroja nueva información del estado de la regla, ejemplo de ello son las pendientes de las líneas. Al final se muestra el caso de estudio que consiste en medir 20mm de una regla de cristal; el resultado obtenido es comparado con el proceso de medición que realiza el *CENAM* mejorando la precisión de la medición obtenida.

ABSTRACT

The measurement of graduation lines in a ruler is an activity of high importance in the world. With these rulers other systems with low accuracy are calibrated. Inside the primary labs the countries have a special department, the function is to measure with high accuracy ($10^{-6}m$) the distance between graduated lines. In the case of México, this work is assigned to the Centro Nacional de Metrología (CENAM). The calibration of rulers is required by the national factories, because their products must satisfy international standards. Around the world exist few automatic methods for calibration rulers, however these methods consider the rulers as a one-dimensional (1-D) vector. As a result, the information about the quality of the lines is lost. Actually the CENAM makes the measurement with a semi-automatic method, and over a sample of the graduated lines. Thus, the accuracy of the measurement is affected by human uncertainty and the principal objective is not fulfilled. This objective is measurement of all graduated lines. The rulers vary in material and size, the most common are made in steel and stainless steel with a maximum length of 3m, also there are glass rulers with a maximum length of 300mm.

In this thesis we propose a novel method that works off-line to measure the distance between the graduated lines. This process consists of determining with high accuracy ($10^{-6}m$) the distance between graduated lines. For this purpose we used a vision system (VS) that consists of one microscope and a CCD camera with a resolution of 640×480 pixels, the position of the VS with respect to the source is determined by a HP interferometer with a resolution of $5 \times 10^{-9}m$. The VS is displaced over the rule, storing the captured images on video and filing the data of the sensor (Z). The distance Z corresponds to the position of the VS when the image was captured. The work is concentrated in two stages: The first is to determine in a robust way the equation of the line which is better fit to the center of the mark present in the image. To detect the center of the mark in each column with a sub-pixel precision, we used a Gabor filter. With these points we make the robust line regression employing a M-estimator. The second stage corresponds to mapping the computed position obtaining the interferometer coordinates; then we proceed to determine the distance for all the processed lines. This is done using the calibration factor β , to indicate the relationship between a pixel-column and the micras of the sensor. With the factor β and the sensor coordinate Z, the center of the mark is mapped to global coordinates obtaining the nominal distance for all graduated lines.

The proposed method offers new information about the physical state of the ruler, for example the slope of the lines. Finally the study case is shown. The experiment consists of measure 20mm of a glass ruler; the obtained results are compared with the actual process of the CENAM, improving the accuracy obtained.

Índice General

1	Introducción	3
1.1	Estado del Arte	4
1.2	Justificación	8
1.3	Desarrollo de la tesis	9
2	Componentes del Sistema	11
2.1	Arquitectura del Sistema	11
2.1.1	Captura de Datos	12
2.2	Interferometría	13
2.2.1	Movimiento Ondulatorio	14
2.2.2	Representación Compleja	15
2.2.3	Interferencia	16
2.2.4	Interferómetro	18
2.2.5	Interferómetro Michelson con Frecuencia-Dual	20
3	Detección de Líneas	23
3.1	Planteamiento	23
3.2	Preprocesamiento	25
3.2.1	Operación Puntual	25
3.3	Detección de Centros de Marca	28
3.3.1	Número de Marcas y Ubicación Local	28

3.3.2	Detección Centro por Marca	30
3.4	Ajuste de Rectas	35
3.4.1	Mínimos Cuadrados	36
3.4.2	Estadística Robusta	37
3.4.3	Estimadores Robustos	39
3.4.4	Algoritmo General de Estimación	40
3.4.5	Regresión Lineal Usando Estimadores-M	42
4	Ajuste de Datos	45
4.1	Calibración	46
4.1.1	Modelo Matemático	47
4.2	Calibración Robusta	49
4.3	Ajuste Final de Datos	51
4.3.1	Planteamiento	51
4.3.2	Detección de Clusters	52
4.3.3	Post-Procesamiento	55
5	Experimentos	57
5.1	Experimento con Datos Sintéticos	57
5.1.1	Modelo de la Regla	58
5.1.2	Modelo del Sensor	60
5.1.3	Resultados de la Simulación	61
5.2	Datos Reales	65
5.2.1	Ajuste Robusto	66
5.2.2	Calibración de una Regla de Cristal	67
5.2.3	Calibración de una Regla de Acero Inoxidable	75
6	Conclusiones	81

Capítulo 1

Introducción

La motivación del presente trabajo es determinar la distancia entre las líneas graduadas de reglas y cintas de alta exactitud. Para ello se propone un nuevo método automático basado en un sistema de visión. El objetivo principal es determinar la distancia existente entre las líneas graduadas, con la más alta exactitud posible.

La medición de reglas y cintas de alta exactitud es una actividad de gran importancia de ámbito mundial. En base a estas reglas se calibran otros sistemas longitudinales de menor precisión. Entendiéndose por calibración, el proceso de determinar el error entre la distancia nominal (posición ideal) y la posición detectada para cada línea graduada, con respecto a un origen común. Por lo regular la distancia entre cada línea graduada de una regla patrón se encuentra equiespaciada $1mm$. Así el objetivo es determinar el error o desviación en micras entre la posición ideal y la obtenida por el proceso de calibración. Sin pérdida de generalidad en el resto de la tesis nos referiremos a las reglas y cintas de alta exactitud por el acrónimo regla. Por citar algunos ejemplos de aplicación de reglas se tiene: calibración de flexómetros, medición del perímetros de oleoductos, medición reglas de menor exactitud, comparadores ópticos, proyectores de perfiles, etc.

La tarea de calibración de reglas de alta exactitud es asignada a los laboratorios primarios de metrología en los diversos países. En el caso de México, dicha tarea recae en el Departamento de Metrología Dimensional del Centro Nacional de Metrología CENAM

que se encuentra ubicado en la ciudad de Querétaro, Qro.

1.1 Estado del Arte

Los métodos existentes para medir reglas se pueden clasificar en dos categorías: los que usan un sistema de visión y los que usan un sistema sensor láser. A su vez estos se pueden clasificar en sistemas de tiempo real y fuera de línea. En estos métodos se comparte el mismo objetivo; "Determinar con la mayor exactitud posible y de forma automática la desviación¹ existente para la distancia marcada para cada línea". Estos métodos comparten un arquitectura de hardware similar a la mostrada en la sección 2.1. De forma general la arquitectura básica consta de un carro móvil que se desplaza por encima del objeto a medir, en el carro se encuentra montado el sistema de visión o un sistema sensor láser cuya función es determinar el centro de la línea de la regla. La posición del carro es detectada por un interferómetro que detecta desplazamientos con una resolución de $5 \times 10^{-7}m$.

En esta sección se muestran tres métodos distintos de medición, el último método corresponde a la forma en que se miden las reglas actualmente en el CENAM. Cabe mencionar que no existe comercialmente un sistema (hardware-software) con la capacidad de medir reglas, los sistemas son desarrollados en los laboratorios primarios de los diversos países. Como por ejemplo el "National Research Laboratory of Metrology" en Japon, "National Research Council" en Ottawa Canada, "Metrology Research Institute" en Helsinki Finlandia, por nombrar solo algunos.

En la referencia [1] se propone el primer método de visión que opera en tiempo real, la idea básica es la siguiente: capturar una imagen f de la regla (amplificada por un microscopio), luego para determinar la presencia de la marca se realiza una suma sobre

¹Normalmente en micras o centenas de nanómetro.

los renglones obteniendo un arreglo unidimensional (1-D):

$$s(j) = \sum_{i=0}^{n-1} f(i, j) \quad (1.1)$$

donde $j = 0, \dots, m - 1$. Si el vector s pasa un umbral θ entonces existe la presencia de la marca, ver figura 1-1.

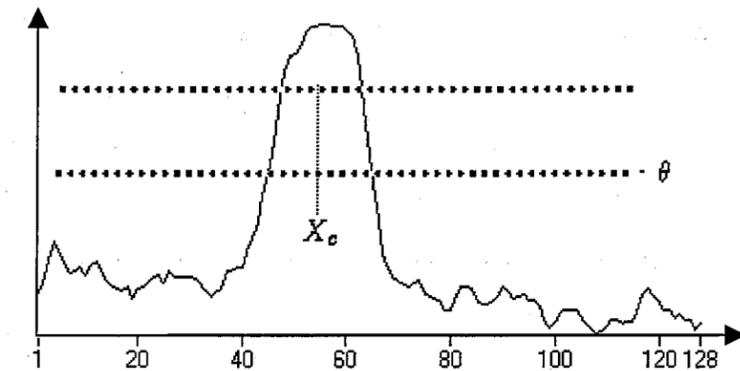


Figura 1-1: Suma por renglones de la imagen

Si se cumple la condición de umbral el centro de la marca se obtiene sacando el centroide geométrico X_c del área comprendida entre las líneas punteadas. Este procedimiento se realiza para cada imagen capturada. Con base a los centros obtenidos y usando información sobre el estado dinámico del sistema de visión, se determina la distancia a la que se encuentra cada línea procesada.

En este método la determinación del centro de la marca no es robusto al ruido y se pierde información al reducir la imagen bidimensional (2-D) a un arreglo unidimensional (1-D). El análisis de los datos 2-D permitiría obtener otras medidas de la calidad de la regla. Esto será analizado con más detalle cuando presentemos nuestra propuesta en la sección 4.3.1.

En la referencia [2] se utiliza un emisor láser (mediante lentes ópticos se obtiene una iluminación en forma de cuña) que incide sobre la regla a calibrar; la luz reflejada

es captada por un fotosensor que la traduce a variaciones de voltaje. El sistema está colocado encima del carro móvil. Los demás componentes del sistema de hardware son similares a las descritos anteriormente. Este método trabaja fuera de línea y se basa en lo siguiente: dado un tipo de material se entona el láser de tal manera que la luz reflejada sobre la marca dé una respuesta alta de voltaje sensada por el fotosensor. Una vez calibrado el sistema se adquieren los datos que corresponden a desplazar el carro por encima de la regla obteniendo así un vector de datos d . En la figura 1-2 se muestra un ejemplo de la variación de voltaje en la presencia de una marca de la regla.

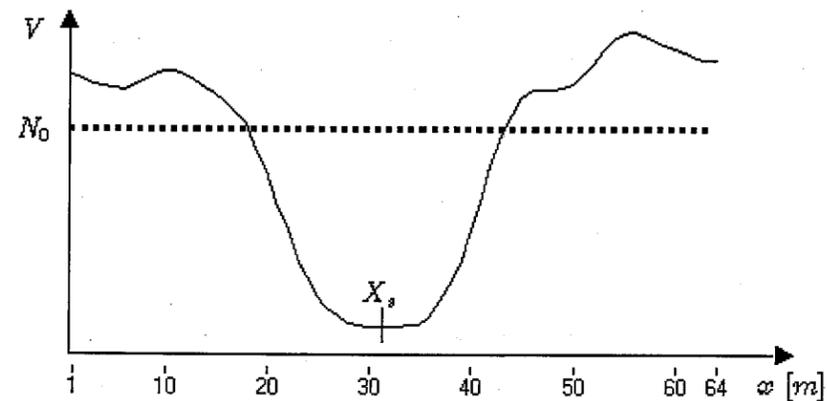


Figura 1-2: Variaciones de voltaje en la presencia de una marca.

Si se pasa un umbral de referencia N_0 el centro de la marca es detectado por:

$$x_s = \frac{\sum_{i=1}^n (x_i y_i (x_i - x_{i-1}))}{\sum_{i=1}^n (y_i (x_i - x_{i-1}))} \quad (1.2)$$

donde y_i indica los valores de voltaje (V) y x_i corresponde a la distancia a la que fue sensado dicho voltaje. La determinación del centro se puede ver como un promedio pesado. Dentro de los inconvenientes se tiene: requiere un proceso de calibración para entonar la frecuencia del láser tal que se exista un contraste alto entre la marca y el fondo

de la regla. Al considerar la regla como un arreglo 1-D, de nuevo se pierde información que se traduce en menos puntos de apoyo para la detección del centro de la marca.

El proceso de medición de reglas en el CENAM utiliza un sistema de visión que procesa los datos en línea, el proceso se realiza de la siguiente forma: después de calibrar el interferómetro y definir un punto inicial se proyecta la imagen de la regla en la pantalla de la computadora, encima de la pantalla se cuenta con una mica que tiene un sistema de coordenadas cartesianas, donde el eje vertical sirve como el sistema de referencia local. Mediante controles electrónicos se ajusta el desplazamiento horizontal del sistema de visión. El objetivo es desplazar la posición del sistema de visión hasta lograr que uno de los cantos de la marca en la imagen se encuentre lo más pegado posible al eje y de la mica, esto se describe gráficamente en la figura 1-3.

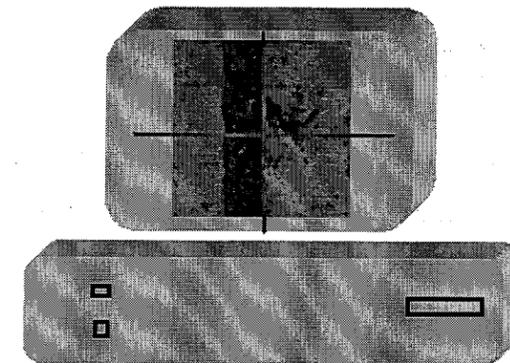


Figura 1-3: Búsqueda del canto de la marca

Una vez ajustado la posición del sistema de visión de tal forma que el canto de la marca coincida con el eje y , se guarda el valor de la posición Z del sensor de posición y se continua con la siguiente marca. Dentro de las desventajas se tienen: el proceso de medición es lento y ello ocasiona que las variables ambientales influyan negativamente, se requiere de una persona de tiempo completo dedicada a dicha labor y al hacer la detección de la marca se incurre en errores debido a que la buena detección de la marca

depende del criterio humano y este criterio se degrada conforme avanzan las horas de trabajo. Además, la medición se realiza sobre una muestra de la regla, es decir no se miden todas las marcas.

En los tres métodos descritos anteriormente el proceso de medición se realiza p -veces, donde $p > 3$. Con estos datos se obtienen valores promedios de la distancia a la que se detectó cada línea procesada, con ello se puede determinar el error entre el valor ideal y el obtenido.

Existen otras formas de medir la distancia entre líneas graduadas, los dos primeros métodos expuestos corresponden a los dos últimos artículos publicados al respecto de este tema y en gran medida muestran el estado del arte. En las referencias [5] y [6] se muestran dos métodos en tiempo real que utilizan un sistema láser, como el descrito anteriormente, y un sistema electrónico para la detección del centro de la marca.

1.2 Justificación

En el presente trabajo se desarrollará un nuevo algoritmo, utilizando un sistema de visión, fuera de línea más exacto para detectar de forma robusta el centro de la marca de la línea logrando una mayor exactitud en la medición a la vez que arroja nueva información para evaluar la calidad de las líneas graduadas. El método propuesto es invariante a los materiales más comunes de fabricación de reglas. La detección del centro de la marca se apoya en todos los renglones de la imagen bidimensional (2-D) logrando un mejor ajuste. Debido a que este trabajo es el resultado de una colaboración CIMAT-CENAM, y su cometido es el de resolver un problema real, se tiene una serie de restricciones. Entre ellas, el sistema de software desarrollado debe de satisfacer:

- Programación en Lab-Windows.
- El proceso de medición debe de ser repetible.
- Los resultados obtenidos deben de ser mejores comparado con el proceso actual.

- Invariante al tipo de material de la regla.
- La medición debe efectuarse sobre todas las líneas.
- La detección de la marca de la regla debe de ser robusta.
- Software documentado.
- Manual de usuario del sistema de software desarrollado.

La ventaja de utilizar un sistema de visión es la generalidad que se logra debido a que el trabajo recae sobre el diseño de algoritmos y no sobre nuevos sistemas de hardware. Uno de los aspectos más importantes en la automatización del proceso de medición es la disminución del tiempo invertido, debido a que las variables ambientales influyen negativamente. Así mismo se minimiza la intervención humana que es una fuente más de incertidumbre, al disminuir el tiempo y la intervención humana se logra una mayor calidad en la medición.

1.3 Desarrollo de la tesis

El contenido del trabajo de tesis es el siguiente: en el segundo capítulo se describen los componentes que conforman al sistema de hardware, la forma de realizar el proceso de captura de los datos y se muestra el principio de interferencia, el cual es requerido para comprender como el interferómetro detecta la posición a la que se encuentra el sistema de visión.

El tercer capítulo es el más extenso, se divide en tres partes: se define el proceso de mejoramiento de las imágenes que consiste en la aplicación de operaciones puntuales que ayuden a resaltar la marca de la regla del resto de la imagen, posteriormente se proponen una serie de algoritmos cuya función principal es detectar los puntos que corresponde al centro de cada marca por renglón con precisión sub-píxel utilizando los filtros de Gabor. En base a estos puntos y utilizando la estadística robusta se obtiene la ecuación de la

recta, obteniendo así los parámetros m, b , donde b indica la posición local (imagen) del centro de la marca.

El cuarto capítulo se compone de dos partes: la calibración que consiste en determinar cuantas unidades del interferómetro son equivalentes a un pixel columna de la imagen, con este factor de calibración β se hace un mapeo entre las coordenadas locales (pixeles) hacia el sistema global (interferómetro.). La segunda parte consiste en determinar la distancia en unidades del interferómetro a la que se encuentra cada línea graduada procesada. En la parte final se explican las nuevas características de calidad que con el sistema propuesto pueden ser obtenidas al calibrar una regla.

El capítulo cinco corresponde a la parte experimental. Se desarrolla un modelo de una regla que sirve para validar el sistema de software en su totalidad, se muestran los resultados obtenidos de la simulación. En base a los datos reales se realiza el ajuste de regresión lineal robusto y no robusto sobre 3 imágenes que corresponden a 3 materiales distintos de fabricación. Por último se muestran dos casos de estudio: el primer caso consiste en medir $20mm$ de una regla de cristal y validar el resultado obtenido con el proceso de medición del CENAM. Para el segundo caso se utiliza una regla de acero inoxidable y se miden 7 líneas, este tipo de reglas representa el caso común de calibración en el CENAM.

Finalmente en el último capítulo del trabajo se determinan las conclusiones generales del trabajo, que de manera objetiva permitan describir las ventajas y desventajas del trabajo desarrollado y se da un panorama del trabajo futuro.

Se incluyen cuatro apéndices, en el primer apéndice se muestran las funciones de estimadores- M más comunes en el área de visión computacional, en el segundo apéndice se tiene el código fuente de los algoritmos propuestos a lo largo de la tesis, en el tercer apéndice se muestra el certificado de medición sobre la regla de cristal de $200mm$ expedido por el CENAM.

Capítulo 2

Componentes del Sistema

En el presente capítulo se describe a detalle las componentes de hardware que conforman al sistema de medición, el proceso de captura de datos y la forma de operación del interferómetro. Para hacer más clara la forma en que el interferómetro es utilizado para detectar la distancia, se exponen los conceptos básicos de interferometría.

2.1 Arquitectura del Sistema

En la figura 2-1 se muestra el diagrama esquemático del sistema de medición. Este sistema se encuentra ubicado en una planta bajo tierra, con ello se reduce interferencia ambiental y aislar en lo posible los ruidos (fuentes de vibraciones).

Las componentes de la arquitectura del sistema son: Interferómetro Hewlet-Packard (I) con cabeza láser 5519A, cuya función es determinar la distancia¹ a la que se encuentra el sistema de visión, un sistema de visión (V) que consta de una cámara CCD con una resolución de 640×480 pixeles montada sobre el microscopio que amplifica la zona de interés. Este sistema de visión se desplaza sobre un tornillo sin fin (T) de alta precisión, el objeto a medir (R) que por lo general es una regla patrón de longitud máxima $3m$, sensores de temperatura (S) colocados encima del material a medir, una computadora

¹con una resolución $5 \times 10^{-9}m$

(C) que realiza el control de sistema y el procesamiento de los datos. Donde (Z) indica la distancia a la que se encuentra posicionado el sistema de visión. Adicionalmente se tienen tarjetas electrónicas para la adquisición de datos.

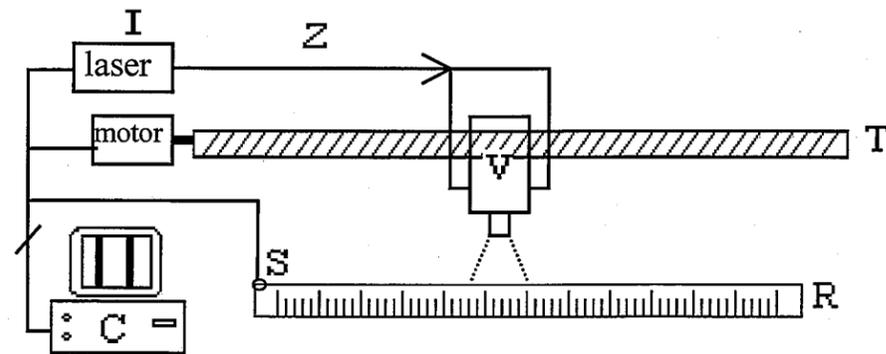


Figura 2-1: Arquitectura del sistema de medición

Las interfaces de comunicación están programadas en Lab-Windows, que está sustentado en el ANSI-C. Antes de definir el procedimiento para la captura de datos se definen tres conceptos que serán utilizados a lo largo de la tesis.

Definición 1 *Línea* : marca física de la regla o la línea graduada.

Definición 2 *Marca* : proyección de la línea en la imagen.

Definición 3 *Recta* : la ecuación de la recta $y = mx + b$ que mejor se ajusta al centro de la marca.

2.1.1 Captura de Datos

La captura de datos se hace de manera semi-continua, se desplaza una cierta distancia Δz el sistema de visión y se espera un tiempo Δt para que el sistema de medición se estabilice, después se graba la imagen f y el valor del interferómetro z , esto se realiza sobre las líneas a medir generando dos archivos de datos representados por la ecuación

(2.1).

$$V = \{f_0, \dots, f_n\} \quad (2.1)$$

$$Z = \{z_0, \dots, z_n\}$$

Este proceso se repite p -veces generando p -parejas de archivos de datos, donde $p > 3$. Para fines de claridad estos archivos de datos se pueden considerar como un solo par de archivos descritos por la ecuación (2.1) donde $n = pN$, siendo N el número de imagen-sensor tomados.

Actualmente no se tiene automatizado el proceso de captura de datos, este proceso se realiza de manera semi-automática. Utilizando controles electrónicos se ajusta el desplazamiento del sistema de visión. La captura de la imagen y los datos del sensor se realiza de manera manual.

El proceso de medición se ve afectado por variables ambientales como son: temperatura, humedad, presión atmosférica, índice de refracción del aire y coeficiente de expansión térmico del material a medir. Estas variables son almacenadas durante el proceso de adquisición de datos y son requeridas para el ajuste final de los datos. El objetivo de este trabajo es realizar únicamente la medición sin tomar en cuenta el ajuste por condiciones ambientales, en [9] se muestra la forma de ajustar el índice de dilatación del material y se encuentra un resumen de la influencia de las variables ambientales al proceso de medición.

2.2 Interferometría

En esta sección se muestran los conceptos teóricos sobre los cuales se basa el funcionamiento del interferómetro. Al final se muestra el interferómetro utilizado y se describe brevemente su modo de operación.

2.2.1 Movimiento Ondulatorio

En la figura 2-2 se muestra el movimiento armónico de una onda 1-D que se propaga en la dirección z . El modelo matemático que la representa se describe por la ecuación (2.2):

$$\psi(z, t) = U \cos \left[2\pi \left(\frac{z}{\lambda} - \nu t + \delta \right) \right] \quad (2.2)$$

donde

U : amplitud

λ : longitud de onda

ν : frecuencia (el número de ondas por unidad de tiempo)

$k = \frac{2\pi}{\lambda}$: número de onda

δ : desfazamiento expresado en radianes

El argumento de la función coseno se denomina la fase de la onda.

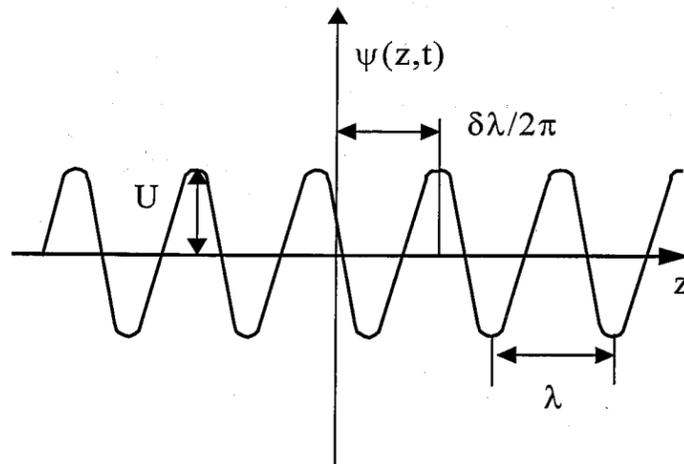


Figura 2-2: Movimiento armónico de la onda

2.2.2 Representación Compleja

La representación de ondas con números complejos ofrece una descripción alternativa que es matemáticamente más simple de trabajar. El número complejo z tiene la forma:

$$z = x + iy \quad (2.3)$$

donde $i = \sqrt{-1}$. Las partes real e imaginaria de z son respectivamente x e y , con $x, y \in \mathbb{R}$. En términos de coordenadas polares (r, θ) , se tiene

$$x = r \cos \theta, \quad y = r \sin \theta \quad (2.4)$$

y

$$z = r (\cos \theta + i \sin \theta). \quad (2.5)$$

La fórmula de Euler establece que

$$e^{i\theta} = \cos \theta + i \sin \theta \quad (2.6)$$

esto nos permite escribir

$$z = r e^{i\theta} \quad (2.7)$$

donde r es la magnitud² de z , y ϕ es el ángulo de fase³ de z dado en radianes. Usando la representación compleja la ecuación (2.2) puede ser escrita

$$\psi(z, t) = \text{Re} \{ U e^{i 2\pi \left(\frac{z}{\lambda} - \nu t + \delta \right)} \} \quad (2.8)$$

²denotada $|z| = \sqrt{x^2 + y^2}$

³donde: $\tan(\theta) = y/x$

Por convención se omite el símbolo de la representación real. Haciendo $\phi = 2\pi \left(\frac{z}{\lambda} - \nu t + \delta\right)$ se tiene:

$$\psi(z, t) = Ue^{i\phi} \quad (2.9)$$

donde U es la magnitud y ϕ la fase. Esta ecuación es la representación en notación compleja del movimiento ondulatorio de una onda.

2.2.3 Interferencia

La interferencia puede ocurrir cuando dos o más ondas son superpuestas unas sobre otras en el espacio. Sean dos ondas descritas por la ecuación (2.10).

$$\begin{aligned} u_1 &= U_1 e^{i\phi_1} \\ u_2 &= U_2 e^{i\phi_2} \end{aligned} \quad (2.10)$$

Si las ondas se traslapan entonces el principio de superposición de la teoría de ondas electromagnéticas establece que el campo resultante es la suma algebraica de dichas ondas. Para este caso se tiene:

$$u = u_1 + u_2 \quad (2.11)$$

donde u se conoce como la perturbación óptica o campo luminoso. Este campo luminoso varía en un tiempo sumamente rápido, haciendo que el campo real sea una cantidad prácticamente indetectable [10] (pág. 293). Por otro lado, la irradiancia I puede ser medida directamente usando una gran variedad de sensores (por ejemplo, fotoceldas, bolómetros, emulsiones fotográficas o los ojos). La ecuación de la irradiancia o intensidad se expresa como:

$$I = |u|^2 = U^2 \quad (2.12)$$

Obteniendo la intensidad de la ecuación (2.11) se establece:

$$\begin{aligned} I &= |u|^2 \\ &= |U_1 e^{i\phi_1} + U_2 e^{i\phi_2}|^2 \\ &= (U_1 e^{i\phi_1} + U_2 e^{i\phi_2})(U_1 e^{i\phi_1} + U_2 e^{i\phi_2})^* \\ &= (U_1 e^{i\phi_1} + U_2 e^{i\phi_2})(U_1 e^{-i\phi_1} + U_2 e^{-i\phi_2}) \\ &= U_1^2 e^{i(\phi_1 - \phi_1)} + U_2^2 e^{i(\phi_2 - \phi_2)} + U_1 U_2 (e^{i(\phi_1 - \phi_2)} + e^{-i(\phi_1 - \phi_2)}) \\ &= U_1^2 + U_2^2 + 2U_1 U_2 \cos(\phi_1 - \phi_2) \end{aligned}$$

de donde se obtiene finalmente

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos(\Delta\phi) \quad (2.13)$$

donde $\Delta\phi = \phi_1 - \phi_2$ es la diferencia de fase. Como se nota en la ecuación (2.13) el campo resultante es la suma de las intensidades $I_1 + I_2$ más el término $2\sqrt{I_1 I_2} \cos(\Delta\phi)$ que se conoce como término de interferencia. Por ello la irradiancia resultante puede ser menor, mayor o igual a $I_1 + I_2$ dependiendo del valor del término de interferencia, es decir, dependiendo del valor de $\Delta\phi$. Un máximo de la irradiancia se obtiene cuando $\cos(\Delta\phi) = 1$ y esto ocurre cuando $\Delta\phi = 0, \pm 2\pi, \pm 4\pi, \dots$. En este caso la diferencia de fase entre las dos ondas es un múltiplo entero de 2π , y las perturbaciones están en fase o se genera una interferencia total (I_{\max}). Cuando $0 < \cos(\Delta\phi) < 1$ las ondas están fuera de fase y esto se conoce como interferencia constructiva. La interferencia destructiva total (I_{\min}) sucede si $\Delta\phi = \pm\pi, \pm 3\pi, \pm 5\pi, \dots$. Cuando $0 > \cos(\Delta\phi) > -1$ se tiene la condición de interferencia destructiva $I_1 + I_2 > I > I_{\min}$.

Para que dos ondas de luz polarizadas se interfieran se deben de satisfacer las leyes de Fresnel-Arago [10] (pág. 299). El punto principal es que las ondas de luz se interfieren si viene de la misma fuente. Si dos haces de luz deben de interferir para formar un patrón estable, éstos deben de tener casi la misma frecuencia. Una diferencia de frecuencia

significante resultaría en una diferencia de fase dependiente del tiempo, variando rápidamente lo cual ocasionaría que el promedio en el tiempo del término de interferencia resultara cero durante el intervalo de detección.

Los patrones más claros existirán cuando las ondas que interfirieron tengan amplitudes iguales o casi iguales. Las regiones centrales de las franjas oscuras y claras corresponden entonces a interferencia completamente destructiva o constructiva, respectivamente.

2.2.4 Interferómetro

El fenómeno de interferencia puede ser observado en interferómetros, uno de los más conocidos es el interferómetro de Michelson. Actualmente existe una gran variedad de interferómetros, ejemplo de ello son el interferómetro de Fresnel, Mach-Zehnder, Sagnac, Pohl, etc., [10] (págs. 299-311). En ellos y en los lasers comerciales se pueden detectar los siguientes componentes generales mostrados en la figura 2-3, [8] (pág 6-4).

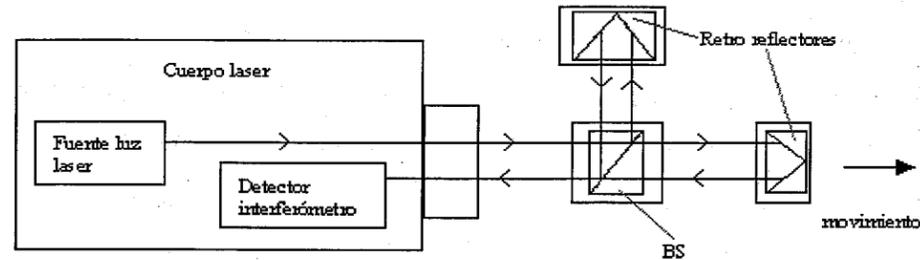


Figura 2-3: Arquitectura de interferómetro comercial

Fuente de luz láser

La fuente de luz láser es una cavidad cerrada llena generalmente con gas Helio-Neon (HeNe), cuando un alto voltaje es aplicado, emite una radiación coherente que produce una banda de luz de frecuencia muy angosta.

Divisor de haz

Un divisor de haz (*beam splitter*) es un instrumento óptico que divide la luz en dos o más haces por separado. Un divisor de haz polarizado tiene la propiedad adicional de transmitir solo una polarización de la luz y reflejar la otra.

Retroreflector

Son espejos especiales que están empotrados en la parte interna de un cubo (*cube - corner*), tienen la propiedad que reflejan la luz incidente de forma paralela no importando el ángulo de incidencia.

Detector de interferencias

Un detector de interferencias determina el cambio en la distancia entre dos componentes ópticos usando un detector óptico de franjas y un circuito de conteo en la cabeza del láser. Dependiendo de como la luz es dividida se pueden clasificar básicamente interferómetros de división de frente de onda y división de amplitud, aunque existen otros modelos que no caen en esta categoría.

Un interferómetro comercial mide los cambios en la distancia por el conteo del número de ondas de luz que son detectadas por el interferómetro. La longitud de la onda es conocida con gran precisión y con ello la distancia puede ser calculada de manera confiable. La longitud de la onda de luz de un láser HeNe es aproximadamente $632nm$ y pudiendo subdividir la onda se puede tener una resolución de $1.25nm$. Los sistemas comerciales más comunes operan usando: Frecuencia Simple (DC) o Dos Frecuencias AC. El interferómetro utilizado emplea dos frecuencias para medir el desplazamiento en función de la diferencia de frecuencias como se indica en la subsección sección 2.2.5.

Efecto Zeeman

Para obtener dos ondas de frecuencias ligeramente diferentes se utiliza el efecto Zeeman. Cuando los átomos de una lámpara de descarga, en este caso haz láser, se sujetan a un campo magnético, sus niveles de energía se dividen. Esto ocasiona que la luz emitida

contenga dos componentes de frecuencias v_1 y v_2 que difieren en proporción a la magnitud del campo aplicado.

2.2.5 Interferómetro Michelson con Frecuencia-Dual

Este tipo de interferómetros llamados interferómetros *a.c* (corriente alterna), mide los desplazamientos en términos de variaciones en la frecuencia como es la medición de variaciones en la intensidad en el interferómetro Michelson. Este tipo de interferómetros es menos susceptible a perturbaciones, como es la turbulencia del aire.

Utilizando dos ondas planas descritas por la ecuación (2.14), [7]:

$$\begin{aligned}\Psi_1 &= e^{i\phi_1} \\ \Psi_2 &= e^{i\phi_2}\end{aligned}\quad (2.14)$$

sumando ambas ondas y utilizando el resultado obtenido en la sección 2.2.3, se obtiene la ecuación (2.15):

$$I = 2 \left[1 + \cos 2\pi \left(\left(\frac{1}{\lambda_1} - \frac{1}{\lambda_2} \right) z - (v_1 - v_2) t \right) \right] \quad (2.15)$$

Si la diferencia de frecuencia $v_1 - v_2$ es muy pequeña y constante, esta variación en I en el tiempo puede ser detectada. Esto es el principio de funcionamiento del interferómetro Michelson con frecuencia-dual para medir distancias.

El diagrama esquemático del interferómetro de Hewlet-Packard se muestra en la figura 2-4. La fuente de luz es un láser donde la frecuencia tiene dos componentes f_1 y f_2 ($f = v/2\pi$) con una frecuencia de diferencia de 2Mhz. Estas dos componentes son generadas utilizando el efecto Zeeman, la luz resultante está dividida en dos ondas ortogonalmente polarizadas. Un divisor de haz polarizador transmite la componente de frecuencia f_1 al espejo movible o externo y la componente de frecuencia f_2 al espejo fijo. Estos espejos reflejan el rayo de luz láser de forma paralela independientemente del ángulo de incidencia. Si el espejo movible se mueve con una velocidad v_1 , la frecuencia del rayo de luz reflejado

es modelado por el desplazamiento Doppler con incrementos $\Delta f = 2v/\lambda_1$.

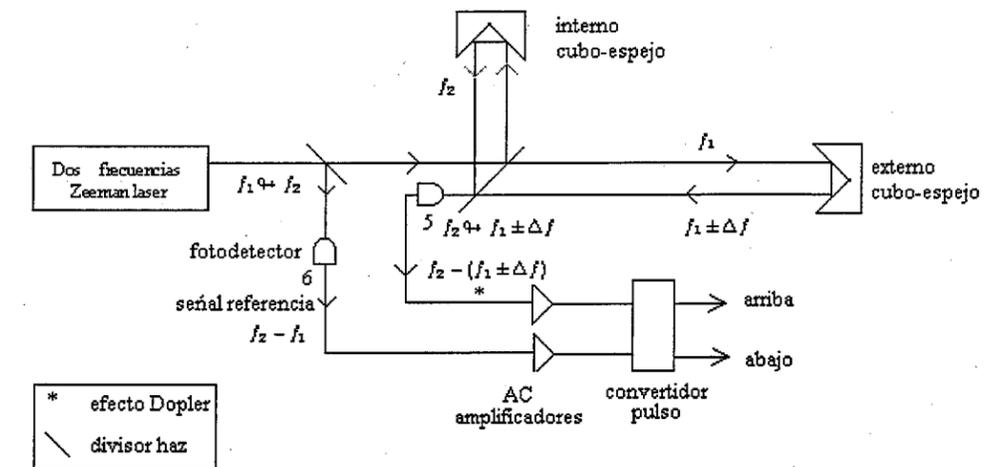


Figura 2-4: Arquitectura del interferómetro utilizado.

Las dos ondas de luz se interfieren en un fotodetector 5 que tiene una señal eléctrica de frecuencia $f_2 - (f_1 \pm \Delta f)$ como se expresa en la ecuación (2.15). Una fracción de la frecuencias del láser f_1 y f_2 son enviadas al detector 6 produciendo una frecuencia de referencia $f_2 - f_1$, que después son amplificadas y enviadas a contadores de frecuencia. Las frecuencias almacenadas en el contador de pulsos son substraídas obteniendo Δf que determina el desplazamiento:

$$s = vt = \Delta f \lambda t / 2 \quad (2.16)$$

Con esto se obtiene finalmente la distancia z a la que se encuentra el sistema de visión descrito en la sección 2.1. El interferómetro detecta desplazamientos de $10^{-7}m.$, es decir, de centenas de nanómetro o diez-milésimas de milímetro. La mejora de esta medición se logra controlando con mayor exactitud las condiciones ambientales, sin embargo esto resulta incosteable.

Capítulo 3

Detección de Líneas

En el presente capítulo se muestra la forma robusta de obtener la recta que mejor se ajusta al centro de cada marca. El capítulo se divide en cuatro secciones: en la primera sección se plantea el problema y se muestra el diagrama esquemático de la solución deseada, en la segunda sección se definen los operadores puntuales necesarios para el mejoramiento de la imagen, en la tercera sección se muestra la forma de detectar el centro por renglón de cada marca con precisión subpixel y por último se realiza el ajuste de regresión lineal robusto sobre los puntos de los centros de la marca, obteniendo así la ecuación de la recta.

3.1 Planteamiento

Como dato de entrada se tienen un archivo con las imágenes descrito por la ecuación (2.1), cada imagen f contiene varias marcas: $\{l_0, \dots, l_k\}$ donde k denota el número de marcas presentes en la imagen, por lo general $k \in [1, 4]$. El número de marcas depende del zoom empleado en la captura del video. Dependiendo del tipo de material del objeto a calibrar se obtienen distintas imágenes que tienen en común un contraste entre el fondo de la imagen y la marca. Los materiales de fabricación de las reglas patrón más comunes son: cristal, acero y acero inoxidable. En la figura 3-1 se muestran las imágenes de los

materiales más comunes.

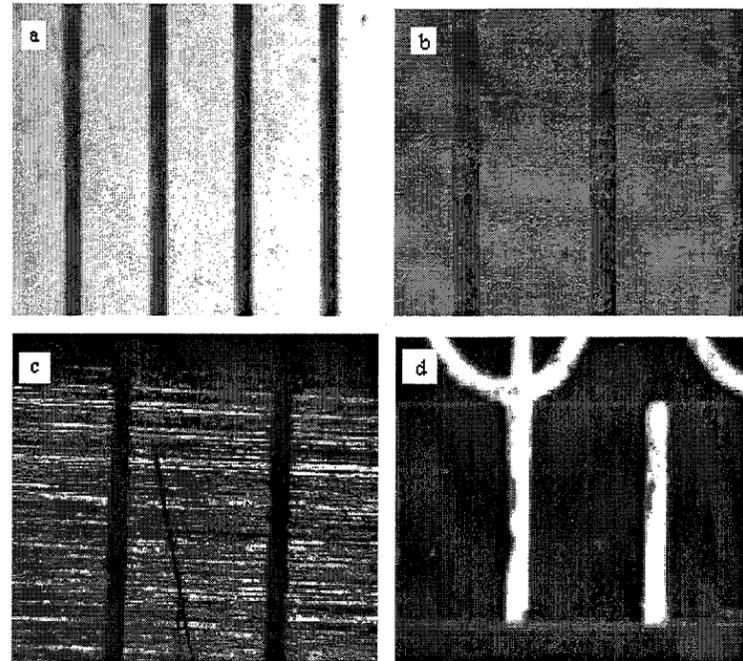


Figura 3-1: Imágenes de 4 materiales distintos de fabricación. a) Regla de cristal. b) Regla de acero. c) Regla de acero inoxidable. d) Cinta flexible.

Para la detección del centro de la marca se requiere que el color de la marca sea claro y exista un contraste con el fondo de la imagen. Para ello se requiere preprocesar a las imágenes. A manera de bloques se muestra en la figura 3-2 el diagrama para detectar el centro de cada marca, para cada imagen.



Figura 3-2: Diagrama del algoritmo de detección de líneas

Cada elemento (m_i, b_i) son los parámetros de la recta que mejor se ajusta a cada marca. El diagrama a bloques se interpreta de la siguiente manera; como dato de entrada

se tiene una imagen $f \in V$, el primer paso es el preprocesamiento o mejora de la imagen que básicamente consiste en resaltar la marca de la imagen con respecto del fondo. El segundo bloque representa la detección del centro X_c^* con precisión subpixel para cada renglón y para cada marca, esto se logra utilizando un filtro de Gabor. El último paso es hacer una ajuste de regresión lineal robusto sobre los puntos X_c^* obteniendo finalmente el conjunto solución descrito por la ecuación (3.1).

$$R = \{(m_0, b_0), \dots, (m_k, b_k)\} \quad (3.1)$$

El detalle de cada parte del diagrama se explica en las siguientes secciones.

3.2 Preprocesamiento

Básicamente existen dos clases de preprocesamiento: operación puntual y geométrica [11]. En este caso el interés es sobre la operación puntual.

3.2.1 Operación Puntual

Una operación puntual modifica el valor de los tonos de gris a cada pixel de manera individual dependiendo solo del nivel de gris y posiblemente de la posición espacial del pixel, dicha operación queda expresada por:

$$I_2(x, y) = P_{xy}(I_1(x, y)) \quad (3.2)$$

los índices de la función P denotan la posible dependencia de la operación puntual sobre la posición del pixel. Cuando la operación puntual es independiente de la posición del pixel se llama *operación puntual homogénea* que es el caso de interés y es representado por:

$$I_2(x, y) = F(I_1(x, y)) \quad (3.3)$$

Estas funciones F también son conocidas como funciones de transferencia de tonos, donde cada elemento del dominio I_1 es mapeado a I_2 empleando la función F . En general las funciones tonales distribuyen el rango dinámico de tal manera que se resalten o atenúen ciertos intervalos del dominio.

Dependiendo del tipo de material de la regla se eligen las funciones F para aumentar el contraste entre la marca de la regla y el fondo. Los operadores puntuales requeridos para el preprocesamiento utilizados son: invertir tonos, elevar a una potencia $\alpha > 1$ y re-escalamiento lineal.

Invertir Tonos

La operación de invertir tonos es una re-escalamiento lineal con pendiente negativa. Para ello se requiere saber los valores máximos y mínimos de la imagen de entrada f_{max} y f_{min} respectivamente. Esta operación queda definida por la ecuación (3.4), en la figura 3-3 se muestra la gráfica de la función de invertir tonos.

$$g(x, y) = f_{max} - (f(x, y) - f_{min}) \quad (3.4)$$

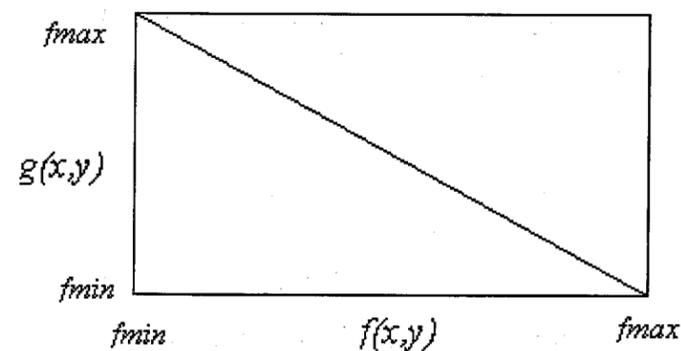


Figura 3-3: Inversor de tonos

Elevar a una Potencia

El elevar a una potencia $\alpha > 1$ los tonos de gris de la imagen ocasiona que los valores altos ocupen un rango dinámico mayor. Es decir, un cierto intervalo Δf es mapeado a un intervalo Δg donde $\Delta g > \Delta f$ ocasionando que los tonos de gris en estos intervalos ocupen un rango mayor, en particular se quiere resaltar la marca de regla cuyos tonos de gris son altos. La ecuación (3.5) define tal operador y su gráfica se muestra en la figura 3-4.

$$g(x, y) = pow(f(x, y), \alpha) \quad (3.5)$$

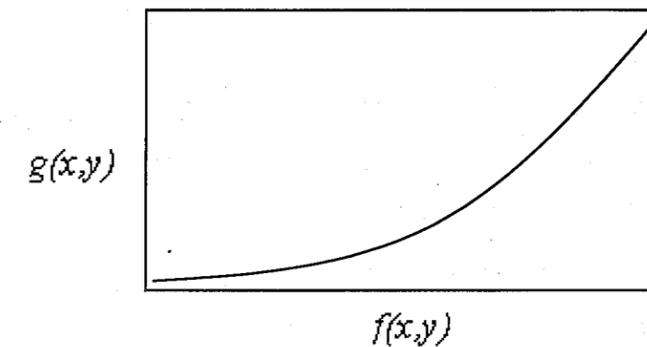


Figura 3-4: Elevar a una potencia

Re-escalamiento Lineal

Al aplicar algún procesamiento sobre la imagen f se afecta el rango dinámico de los tonos de gris y para fines de visualización es conveniente remapear los tonos de gris al intervalo $[0, 255]$. Para ello se define la ecuación (3.6) de re-escalamiento lineal que mapea el intervalo Δf al intervalo Δg , en este caso $g_{min} = 0$ y $g_{max} = 255$.

$$g(x, y) = g_{min} + (g_{max} - g_{min}) \left(\frac{f(x, y) - f_{min}}{f_{max} - f_{min}} \right) \quad (3.6)$$

En la figura 3-5 se muestra la gráfica del re-escalamiento lineal, cuando el valor de entrada es f_{min} como salida se tiene el valor mínimo de g_{min} y si $f(x,y) = f_{max}$ este valor es mapeado a g_{max} , como se indica en la ecuación (3.6).

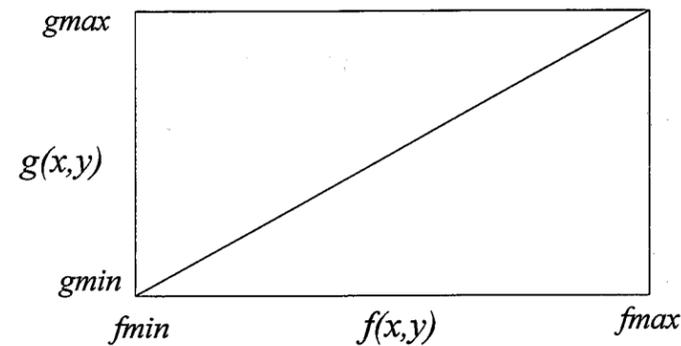


Figura 3-5: Re-escalamiento lineal

En la literatura existe una gran cantidad de operadores puntuales, uno de los más comunes es la ecualización del histograma. Una mayor referencia de operadores puntuales se puede encontrar en [11] y [12]. La elección del operador puntual y el orden de aplicación depende del tipo de imagen analizada, la selección de los operadores queda como un parámetro programable dentro del sistema de software.

3.3 Detección de Centros de Marca

La detección del centro por renglón para cada marca es la tarea de procesamiento de imágenes más importante del presente trabajo. Para tal tarea se divide la sección en dos partes: determinación del número de marcas en la imagen y la detección del centro con precisión sub-píxel.

3.3.1 Número de Marcas y Ubicación Local

En la figura 3-6 se muestra una imagen con dos marcas de una regla de acero inoxidable, el objetivo es seleccionar sub-regiones de la imagen que contengan una sola marca, para

tal tarea se requiere determinar el número de marcas y su ubicación. Para ello se propone el algoritmo-1, en base a dicho algoritmo se determina el número de marcas y la ubicación local de los centros de las marcas, estos datos son requeridos en la siguiente sección.

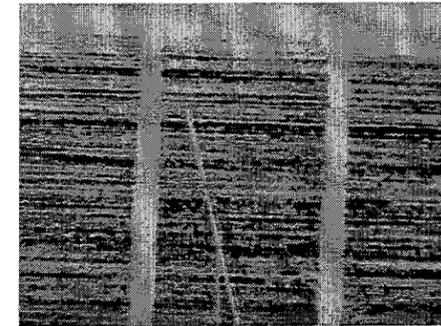


Figura 3-6: Imagen de una regla de acero inoxidable con dos marcas.

Como dato de entrada se tiene una imagen f que contiene marcas en color claro sobre fondo oscuro, donde $f(i,j)$ es el valor en tono de gris del i -ésimo renglón y j -ésima columna. Como dato de salida se tiene el vector Max_Mag que contiene las posiciones en donde se encuentran los centros generales de las marcas y en $\#N_m$ se guarda el número de marcas detectado.

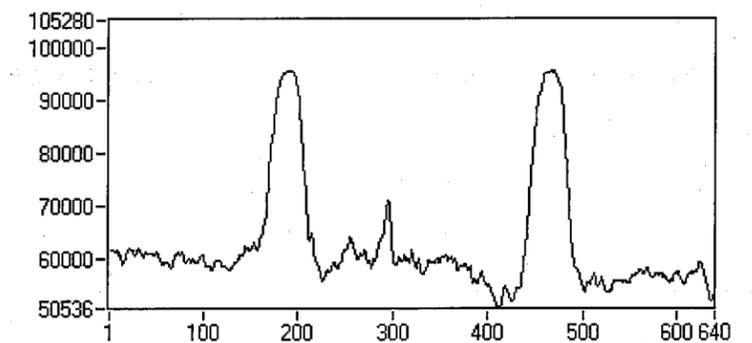


Figura 3-7: Integral por columnas de una imagen con dos marcas de la figura 3-6.

Algoritmo 1 Número de marcas y ubicación

1. Realizar la suma por columnas : $integral(i) = \sum_{j=0}^m f(j, i)$, ver figura 3-7.

2. Obtener el valor máximo de la integral: $x_{max} = \arg \max_x (integral(x))$

3. Calcular la cota inferior en función de un porcentaje φ :

$$cota_inf = x_{max} - (x_{max} * \varphi) / 100$$

4. Seleccionar los puntos de la integral que estén por encima de la cota_inf.

$$arr(x) = \begin{cases} 1 & \text{si } integral(x) > cota_inf \\ 0 & \text{otro caso} \end{cases}$$

5. Recorrer $arr(x)$ y determinar el número de marcas $\#N_m$, donde cada marca corresponde a detectar el par de cambios de (0 a 1) y (1 a 0).

6. Determinar los centro de las marcas, esto se logra detectando el centro entre el par de cambios (0 a 1) y (1 a 0). Se almacenan en el arreglo Max_Mag.

El código fuente del algoritmo se muestra en la sección 1 del apéndice B. En la práctica se elige $\varphi = 25\%$, que queda como un parámetro programable. Para el caso de la figura 3-7 la salida obtenida fue: $\#N_m = 2$ y $Max_Mag = \{194, 468\}$,

3.3.2 Detección Centro por Marca

En la sección anterior se determinó el número de marcas y su posición aproximada, sin pérdida de generalidad y para enfocarse en la idea del algoritmo, en el resto de la sección se asume que la imagen contiene una sola marca, la determinación de los centros por renglón para las marcas restantes se realiza de forma similar.

En la figura 3-8 se muestra el renglón 64 de una imagen de una regla de cristal, el objetivo es determinar para cada renglón de la imagen el centro con precisión subpixel de la marca, la marca corresponde al montículo observado.

Para determinar el centro de la marca se eligió el uso de un filtro de Gabor, antes de describir el algoritmo para la detección de los centros se exponen algunos conceptos que son necesarios.

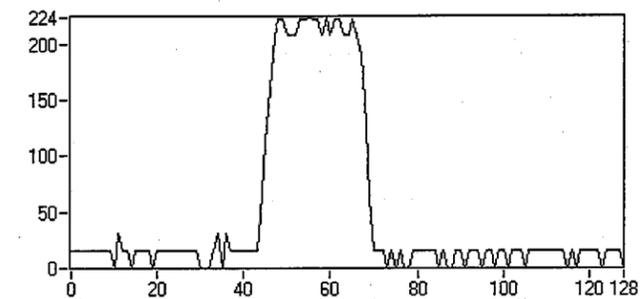


Figura 3-8: Niveles de gris del renglón 64 de una imagen de la regla de cristal.

Fase Local

La señal de entrada corresponde a una señal 1-D como se muestra en la figura 3-8, esta gráfica puede considerarse que se aproxima a la cresta de una onda cosenoidal. El punto donde la cresta de la onda es máxima corresponde al centro de la marca que se quiere detectar. Esta posición esta en función del argumento w_0 de la función coseno, donde la fase tiene el valor de cero. Para determinar la fase de la onda cosenoidal $g(x) = \cos(w_0x)$ asumase que existe un operador H que retrasa la fase en 90° , obteniendo una señal que está en cuadratura con la primera:

$$g'(x) = -\sin(w_0x) \quad (3.7)$$

Entonces la fase ϕ se puede obtener como:

$$\phi = \arctan\left(\frac{-g'(x)}{g(x)}\right) \quad (3.8)$$

Este operador corresponde al operador de Hilbert [18] (pág 261), cuya función de transferencia está definida por:

$$H_i(u) = \begin{cases} -iH_r(u) & \text{si } u < 0 \\ iH_r(u) & \text{si } u \geq 0 \end{cases} \quad (3.9)$$

donde $H_i(u)$ es la función de transferencia de $g(x)$. Las funciones H_r y H_i son reales, esto hace posible que se puedan combinar en un filtro complejo donde H_r es la parte real y el filtro H_i es la parte imaginaria:

$$h(x) = h_r(x) - ih_i(x) \quad (3.10)$$

La fase es representada como un valor complejo donde la magnitud representa la energía de la señal y el argumento refleja la relación entre la parte par e impar de la señal. Dentro de las características que tiene la fase local se tienen:

- La fase local estimada es invariante a la energía de la señal.
- La fase local estimada y su posición espacial son equivalentes.
- La derivada espacial de la fase local estimada es equivalente con la frecuencia espacial.

Un filtro que cumple con las condiciones de las ecuaciones (3.9,3.10) se denomina filtro de cuadratura, una demostración de ello se encuentra en [19] (págs. 15-17).

Filtros de Cuadratura

Los filtros de cuadratura [11] (pág. 369) son una alternativa para obtener una par de señales al convolucionar un renglón de la imagen, se llama en cuadratura debido a que las señales del filtro difieren solamente por un corrimiento de $\pi/2$. La salida de estos filtros se parece a una función analítica, sin embargo, la magnitud de la función de transferencia no es unitaria y puede tomar cualquier valor real. El más conocido filtro de cuadratura es el filtro pasa banda de Gabor, que corresponde al caso adoptado en el presente trabajo.

La detección del centro de la marca, ver figura 3-8, se logra haciendo que la mitad del período de la frecuencia w_0 se asemeje al ancho de la línea. La ecuación (3.11) define el filtro de Gabor.

$$h(x) = g_\sigma(x) [\cos(\omega_0 x) + i \sin(\omega_0 x)] \quad (3.11)$$

donde $g_\sigma(x)$ es una gaussiana con $\sigma = w_0/4$. En la figura 3-9 se muestra la parte real del filtro y en la figura 3-10 su parte imaginaria.

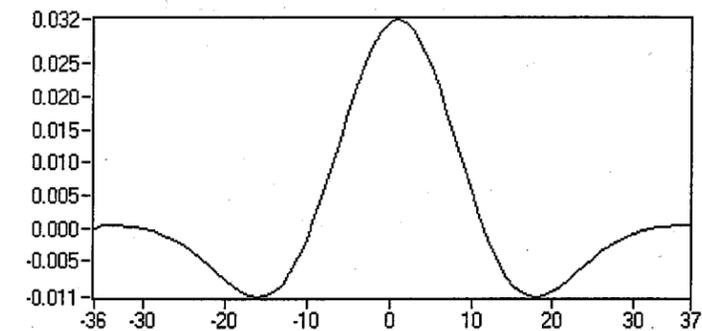


Figura 3-9: Parte real del filtro de Gabor

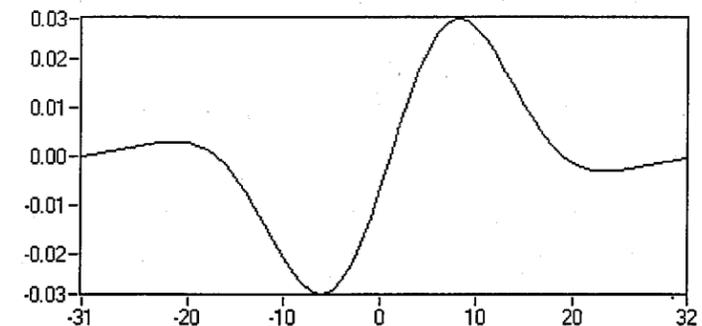


Figura 3-10: Parte imaginaria del filtro de Gabor

El resultado de aplicar el filtro de cuadratura a un renglón de la imagen se muestra en la figura 3-11, el panel (a) corresponde a la señal de entrada, paneles (b) y (c) son gráficas de la magnitud y la fase de la señal de salida del filtro, respectivamente.

El valor de la fase en la figura 3-11 toma el valor de cero cuando el centro del filtro

se encuentra en el centro de la marca, con el valor de máxima magnitud se determina la posición en donde se encontrará el cruce por cero de la fase, obteniendo con ello la precisión sub-píxel.

Utilizando el filtro de Gabor como detector del centro de la marca se propone el siguiente algoritmo que detecta el centro de la marca con precisión sub-píxel, cuando la imagen contiene más de una marca se aplica de igual forma considerando el número de marcas y la región local donde se buscará el valor de máxima magnitud, obtenidos por el algoritmo-1.

Algoritmo 2 Centro de marcas por renglón

1. Para cada renglón $f(x)$ y siendo $h(x)$ el filtro de Gabor correspondiente realizar los pasos 2-4.

2. Calcular la magnitud M y de fase ϕ , expresadas por:

$$M(x) = \|f(x) \otimes h(x)\| \quad \text{y} \quad \phi(x) = \text{atan2}(\text{Im}(f(x) \otimes h(x)), \text{Re}(f(x) \otimes h(x)))$$

donde \otimes denota la convolución no circular.

3. Encontrar el valor de máxima magnitud

$$x_{\max} = \arg \max\{M(x)\}$$

4. Seleccionar un conjunto de puntos $D = \{(x_0, y_0), \dots, (x_n, y_n)\}$, de la fase $\phi(x)$ en una vecindad de x_{\max} .

5. Usando regresión lineal obtener los parámetros de la recta (m, b) .

6. Obtener el cruce por cero de la fase. Igualando a cero la ecuación $y = mx + b$ se obtiene: $x = -b/m$.

La implementación de este algoritmo y las funciones auxiliares que utiliza se muestran en las secciones 2 a la 4 del apéndice B.

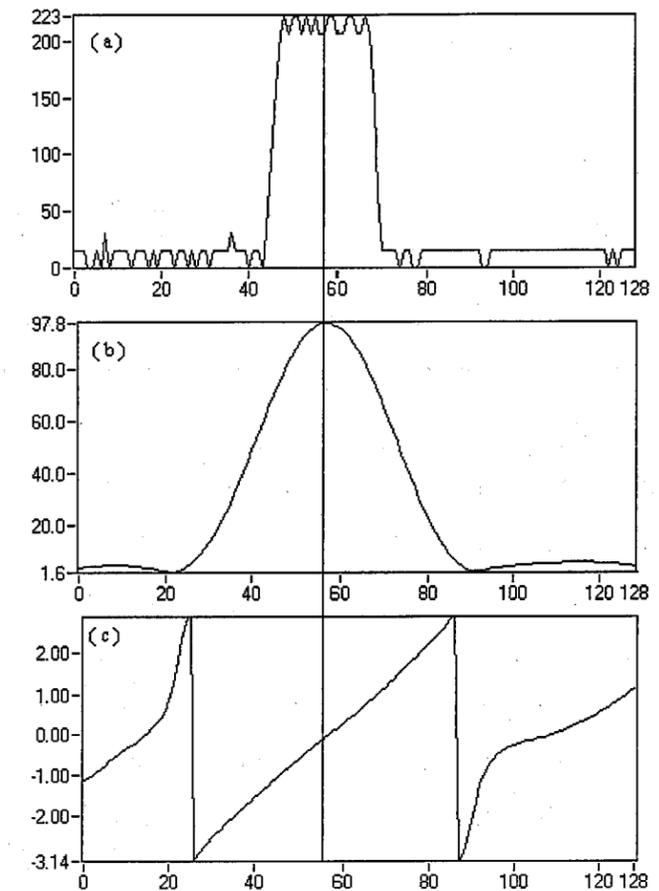


Figura 3-11: Respuesta del filtro de Cuadratura: (a) datos, (b) magnitud y (c) fase.

3.4 Ajuste de Rectas

En la sección anterior se describió la forma de encontrar el centro de cada marca por renglón con precisión sub-píxel. Así para k -marcas que se muestran en la imagen sus centros son almacenados en un arreglo matricial C

$$C = \begin{pmatrix} C_{11} & \cdots & C_{1k} \\ \vdots & & \vdots \\ C_{m1} & \cdots & C_{mk} \end{pmatrix} \quad (3.12)$$

donde

m : número de renglones de la imagen.

k : número de marcas en la imagen.

Los elementos de una columna corresponden a los cruces por cero de la fase ϕ de una marca en particular. El objetivo ahora es ajustar una recta a cada columna de datos de C . Sin pérdida de generalidad en el resto de la sección se considerará $k = 1$. Es decir, la imagen contiene una sola marca, cuando $k > 1$ el ajuste de la recta se realiza sobre cada columna de C de forma similar.

Para poder hacer el ajuste de regresión lineal a cada columna de la matriz C , es necesario cambiar el orden de los ejes. Es decir el eje x pasa a ser el eje y y eje y es el eje x . Este cambio se requiere debido a la pendiente de una recta esta definida como: $m = \Delta y / \Delta x$ y como la marca de la regla es casi vertical, entonces $\Delta x \rightarrow \infty$ y no es posible calcular la pendiente. Por ello en lo subsecuente cuando se hable de la coordenada x realmente se está refiriendo a la coordenada y , y la coordenada y pasa a ser la coordenada x . Este cambio no afecta los resultados obtenidos.

3.4.1 Mínimos Cuadrados

Sea un conjunto de puntos $C = \{(x_1, y_1), \dots, (x_n, y_n)\}$ sobre los cuales se requiere ajustar una recta, suponiendo que los datos están contaminados con ruido blanco se tiene

$$y_i = mx_i + b + \eta \quad (3.13)$$

donde $\eta \sim N(0, \sigma^2)$. El enfoque de mínimos cuadrados consiste en penalizar con norma L_2 la suma de las diferencias entre los valores de la recta aproximante y los valores dados. Esto queda expresado por la ecuación (3.14).

$$U(m, b) = \sum_{i=1}^n (y_i - (mx_i + b))^2 \quad (3.14)$$

que corresponde al estimador de máxima-verosimilitud. Los valores óptimos de m, b se obtienen minimizando la ecuación anterior.

$$m = \frac{n \left(\sum_{i=1}^n x_i y_i \right) - \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n y_i \right)}{n \left(\sum_{i=1}^n x_i^2 \right) - \left(\sum_{i=1}^n x_i \right)^2} \quad (3.15)$$

$$b = \frac{n \left(\sum_{i=1}^n x_i^2 \right) \left(\sum_{i=1}^n y_i \right) - \left(\sum_{i=1}^n x_i y_i \right) \left(\sum_{i=1}^n x_i \right)}{n \left(\sum_{i=1}^n x_i^2 \right) - \left(\sum_{i=1}^n x_i \right)^2} \quad (3.16)$$

Obteniéndose así formulas cerradas para m y b . Este ajuste funciona bien cuando la estadística del ruido de los datos es ruido gaussiano, la presencia de outliers ocasiona que la estimación de los parámetros se sesgue. Esto se debe a como son pesados los residuos al utilizar la norma L_2 , para evitar en lo posible que se sesgue la estimación de los parámetros se utiliza la estadística robusta como se describe en la siguiente sección.

3.4.2 Estadística Robusta

Los métodos clásicos de estadística están basados en dos supuestos: las observaciones de los datos son independientes e idénticamente distribuidas (*iid*) y la distribución del ruido es normal. El primer supuesto puede ser realista en algunos casos, pero el segundo punto es frecuentemente irrealista desde un punto de vista práctico, en efecto, las distribuciones no normales prevalecen en la práctica. Al asumir que la estadística del ruido es normal puede llevar a errores en la inferencia estadística.

Existen varias definiciones de la estadística robusta, en un sentido amplio se define [13] (pág. 7):

Definición 4 "Estadística robusta es una colección de teorías relacionadas, es la estadística de la aproximación a modelos paramétricos, en presencia de valores atípicos.

Esta definición considera a la estadística robusta como una extensión de la estadística

clásica paramétrica, tomando en cuenta que los modelos paramétricos son una aproximación a la realidad. Cualquier procedimiento de estadística robusta se espera que posea las siguientes características [13]:

- Describir la estructura del mejor ajuste, aplicado a la mayoría de los datos.
- Identificar los outliers o estructura de datos desviados, para un tratamiento futuro si es deseado.
- Algunas grandes desviaciones del modelo no deberían causar una catástrofe en la estimación.
- Debería tener una razonable o buena eficiencia del modelo asumido (óptimo o cercano al óptimo).
- Debería ser robusto en el sentido que pequeñas desviaciones de los datos no sesguen la estimación.

Para el propósito del trabajo el término robusto significa *insensibilidad a pequeñas desviaciones de los supuestos*. De manera intuitiva se puede pensar en aplicar los siguientes dos pasos para estimar de manera robusta los parámetros (sin usar métodos robustos):

1. Limpiar los datos mediante la aplicación de alguna regla para eliminar valores atípicos ("outliers").
2. Después usar la estimación clásica sobre el conjunto de datos restantes.

Desafortunadamente esto no es aplicable debido a que es muy raro hacer posible la separación de los dos pasos de manera limpia, en regresión multiparamétrica la detección de outliers es una tarea muy difícil de realizar. Este tipo de procedimientos no es aplicable en general, por ello es conveniente utilizar la estimación robusta. En el artículo "Robust estimation of a location parameter", Huber (1964) se forma la primera base de la teoría de la estimación robusta [13] (pág. 36).

3.4.3 Estimadores Robustos

Uno de los estimadores robustos más usados es el estimador- M . El problema en general es encontrar los parámetros $a = [a_0, \dots, a_n]$ tales que provean el mejor ajuste al modelo $u(s; a)$, sobre un conjunto de mediciones $d = [d_0, \dots, d_s]$, en casos cuando los datos pueden estar contaminados por grandes errores. Así, el objetivo es encontrar los parámetros a , tal que se minimice el valor de los errores residuales $(d_s - u(s; a))$ [16], esto queda expresado por la ecuación (3.17).

$$\min_a \sum_{s \in S} \rho(d_s - u(s; a), \sigma_s) \quad (3.17)$$

donde σ_s es un parámetro de escala y ρ es la norma del error. Equivalentemente se tiene la ecuación implícita.

$$\sum_{s \in S} \Psi(d_s - u(s; a), \sigma_s) = 0 \quad (3.18)$$

donde $\Psi(s; a) = \frac{\delta \rho(s; a)}{\delta a}$, es denominada la función de influencia del estimador- M .

Cuando los errores en las observaciones están normalmente distribuidos la función ρ óptima es la cuadrática

$$\rho(d_s - u(s; a), \sigma_s) = \frac{(d_s - u(s; a))^2}{2\sigma_s^2} \quad (3.19)$$

que corresponde al problema de mínimos cuadrados descrito en la sección anterior.

La selección de diferentes funciones ρ genera diferentes estimadores robustos, la robustez de un estimador estriba en la sensibilidad a los outliers o desviaciones del modelo estadístico asumido [16] (pág. 62). La función ρ debe de satisfacer: $\rho(x) = \rho(-x)$, $\rho(x) \geq 0$ y $\rho(x) \in C^2$. Dependiendo de la monotocidad de la función Ψ los estimadores- M se pueden clasificar como [17]:

Caso 1: monótona, ejemplo el estimador de Huber.

Caso 2: descenso suave, $\Psi(\infty) = 0$.

Caso 3: descenso duro, $\Psi(x) = 0 \forall |x| \geq a$, ejemplo el estimador biweight Tukey's.

En [13] el análisis de la función ρ se hace basándose en el comportamiento de la función de influencia Ψ que mide la influencia de los residuos, este es el enfoque adoptado en el presente trabajo.

La elección del método de mínimos cuadrados es notoriamente sensible a los outliers, esto se debe a que la influencia de los outliers es una línea recta no acotada, esto se muestra en las figura 3-12.

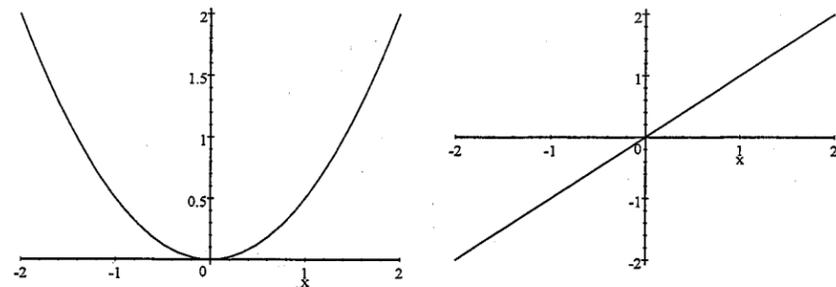


Figura 3-12: a) Función $\rho = x^2$

b) Función de influencia $\Psi = x$

En general se buscan funciones Ψ que pesen menos los datos en la presencia de outliers. Es decir, debe de incrementarse más lentamente que la cuadrática, en el apéndice A se muestran los estimadores M más utilizados para el caso de regresión lineal y en visión computacional.

3.4.4 Algoritmo General de Estimación

A manera de ejemplo se muestran los pasos para el ajuste de regresión lineal usando una función ρ general. Sea $r_i = y_i - (mx_i + b)$ el residuo de los datos y empleando la notación de la ecuación (3.17) se obtiene:

$$\min_{m,b} \sum_{i \in \Omega} \rho(r_i) \quad (3.20)$$

minimizando con respecto m, b se obtiene un sistema con dos ecuaciones

$$\sum_{i \in \Omega} \Psi(r_i) \left(\frac{\partial r_i}{\partial m} \right) = 0, \quad \sum_{i \in \Omega} \Psi(r_i) \left(\frac{\partial r_i}{\partial b} \right) = 0 \quad (3.21)$$

haciendo $w(r_i) = \Psi(r_i)/r_i$ y multiplicando por r_i

$$\sum_{i \in \Omega} w(r_i) (x_i) r_i = 0, \quad \sum_{i \in \Omega} w(r_i) (1) r_i = 0 \quad (3.22)$$

expandiendo el último término, r_i

$$\sum_{i \in \Omega} w(r_i) (x_i) (y_i - mx_i - b) = 0, \quad \sum_{i \in \Omega} w(r_i) (y_i - mx_i - b) = 0 \quad (3.23)$$

despejando los parámetros m, b

$$m = \frac{\sum_{i \in \Omega} w(r_i) (x_i) [y_i - b]}{\sum_{i \in \Omega} w(r_i) x_i^2}, \quad b = \frac{\sum_{i \in \Omega} w(r_i) [y_i - mx_i]}{\sum_{i \in \Omega} w(r_i)} \quad (3.24)$$

Obteniéndose finalmente la expresión iterativa para el cálculo de los parámetros.

$$b^{t+1} = \frac{\sum_{i \in \Omega} w(y_i - m^t x_i - b^t) [y_i - mx_i]}{\sum_{i \in \Omega} w(y_i - m^t x_i - b^t)} \quad (3.25)$$

$$m^{t+1} = \frac{\sum_{i \in \Omega} w(y_i - m^t x_i - b^t) (x_i) [y_i - b^t]}{\sum_{i \in \Omega} w(y_i - m^t x_i - b^t) (x_i^2)} \quad (3.26)$$

La condición de paro se establece $|b^{t+1} - b^t| < \epsilon$ y $|m^{t+1} - m^t| < \epsilon$. El algoritmo requiere de los parámetros de inicialización m^0, b^0 . En el planteamiento anterior se supuso que el factor de escala $\sigma = 1$, este no sucede en general, para ello se requiere determinar de forma robusta la estimación de dicho parámetro, en las secciones 7.7 y 7.8 de [15] se encuentran algunos algoritmos iterativos para el cálculo del factor de escala.

3.4.5 Regresión Lineal Usando Estimadores-M

El estimador-M seleccionado es la función cuadrática truncada, en la parte izquierda de la figura 3-13 se muestra la función ρ y su función de peso Ψ en la parte derecha. La interpretación de la función cuadrática corresponde a considerar que dentro de un cierto intervalo $|x| < a$ los residuos son pesados cuadráticamente y fuera del intervalo son pesados de manera constante, sus funciones ρ y Ψ están definidas por:

$$\rho(x) = \begin{cases} x^2 & |x| < a \\ k & \text{otro caso} \end{cases} \quad \Psi(x) = \begin{cases} x & |x| < a \\ 0 & \text{otro caso} \end{cases} \quad (3.27)$$

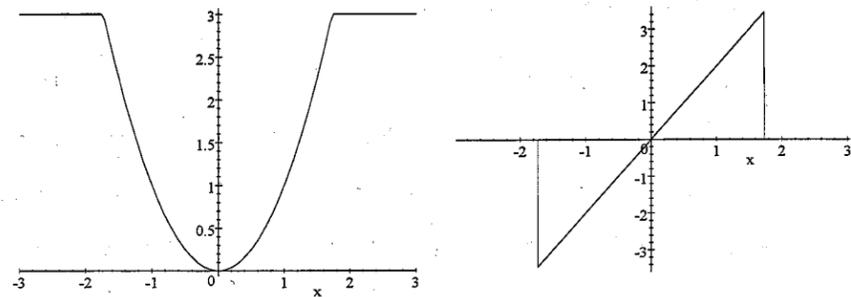


Figura 3-13: a) Cuadrática truncada b) Función de influencia ψ de la cuadrática truncada

El modelo de regresión lineal robusta usando la función cuadrática truncada se describe por:

$$U(m, b) = \sum_{i \in \Omega} (y_i - mx_i - b_i)^2 \quad (3.28)$$

donde la zona Ω corresponde a los puntos que satisfacen $|x| < a$. Minimizando $U(m, b)$ se obtiene los valores óptimos para m, b .

$$m = \frac{n \left(\sum_{i \in \Omega} x_i y_i \right) - \left(\sum_{i \in \Omega} x_i \right) \left(\sum_{i \in \Omega} y_i \right)}{n \left(\sum_{i \in \Omega} x_i^2 \right) - \left(\sum_{i \in \Omega} x_i \right)^2} \quad (3.29)$$

$$b = \frac{n \left(\sum_{i \in \Omega} x_i^2 \right) \left(\sum_{i \in \Omega} y_i \right) - \left(\sum_{i \in \Omega} x_i y_i \right) \left(\sum_{i \in \Omega} x_i \right)}{n \left(\sum_{i \in \Omega} x_i^2 \right) - \left(\sum_{i \in \Omega} x_i \right)^2} \quad (3.30)$$

donde n es el número de puntos que satisfacen la condición de umbral. Con base en estas dos ecuaciones anteriores se formula el algoritmo iterativo para la estimación de los parámetros. A continuación se describe este procedimiento.

Descripción del Algoritmo Iterativo La idea general es ir reduciendo continuamente la cota a logrando con ello la exclusión paulatina de los outliers, para ello se determina a en función de un porcentaje α del ancho de la marca, por definición se inicializa a de tal manera que todos los puntos caen en la región cuadrática. Esto es $\Omega = \text{ancho marca}^1$. Con base en la zona definida se hace el ajuste de regresión robusto usando las ecuaciones (3.29), (3.30) y se obtiene el porcentaje φ de datos que contribuyeron en la estimación. Si φ es menor que un porcentaje ζ se indica que el algoritmo converge y se para el algoritmo. En caso contrario $\varphi > \zeta$ se decrementa la zona a en s unidades y se itera hasta convergencia. Formalizando lo descrito se obtiene el algoritmo de regresión lineal robusta.

¹parámetro que es dado por el usuario del software.

Algoritmo 3 Regresión Lineal Robusta

1. Realizar pasos 2-4 hasta convergencia
2. Estimar m, b usando las ecuaciones (3.29), (3.30) y determinar el porcentaje de datos φ que contribuyeron a la estimación.
3. Si el porcentaje obtenido $\varphi < \zeta$ salir.
4. Decrementar la zona de datos válida $a = a - s$.

Este algoritmo converge rápido y se garantiza la exclusión paulatina de los outliers, el código fuente se muestra en la sección 5 del apéndice B.

El algoritmo se aplica a cada columna de C , ver ecuación (3.12), obteniendo el conjunto solución R descrito por la ecuación (3.1). Así para cada marca se obtienen los parámetros m, b que la caracterizan. En el resto de la tesis solo se hará referencia al parámetro b que indica la posición en coordenadas de la imagen donde se encuentra el centro de la marca.

Capítulo 4

Ajuste de Datos

En el capítulo anterior se describió la forma robusta de obtener la recta que se ajusta el centro de cada marca. Con base en esta información se plantea el último paso del proceso que consta de dos partes:

- Mapear los puntos b 's a coordenadas globales z del interferómetro.
- Obtener la distancia z de cada línea procesada.

El primer punto se denomina calibración, que consiste en encontrar la relación entre las unidades del sensor z y un pixel-columna x de la imagen. Es decir, por cada pixel cuantas unidades del sensor interferométrico equivalen.

$$z \approx \beta x \quad (4.1)$$

donde

z : coordenadas del sensor

x : pixel columna

β : parámetro a estimar

El segundo punto corresponde al ajuste de datos y consiste en utilizar el factor de escala β que mapea el centro de cada marca cuyas coordenadas locales se expresan en

pixeles, a coordenadas globales (μm). Al hacer esto se tiene información redundante para cada línea debido a que dicha línea es referenciada en distintas imágenes. Es decir, aparece en varias imágenes y por lo tanto se tienen varias mediciones de la misma. Por ello, el objetivo es determinar los componentes que hacen referencia a una misma línea, es decir, determinar los "clusters" o grupos de datos. Esto se representa por la ecuación (4.2) que representan a n cluster detectados y corresponden a las n líneas procesadas.

$$K = \{k_0, \dots, k_n\} \quad (4.2)$$

donde $k_i = \{\mu_i, \sigma_i\}$, μ_i es la media muestral y σ_i la desviación de los datos.

4.1 Calibración

Como se muestra en la figura 4-1 una línea cuya posición global es z^w aparece en las imágenes f_0, f_1, \dots, f_n referenciada en las distintas posiciones b_0, b_1, \dots, b_n , respectivamente. Tomando como referencia esa línea, se establece el Sistema de Ecuaciones Sensor Imagen (*SESI*) descrito por la ecuación (4.3), donde la parte derecha corresponde a la posición global z^w (desconocida) que se obtiene de sumar la posición z_i en donde fue tomada la imagen (posición del marco de la imagen) más la posición local en pixeles b_i donde fue detectado el centro de la marca, por el factor de calibración β que es el parámetro a determinar. Esto se expresa como:

$$SESI \begin{cases} z_0 + \beta b_0 = z^w \\ z_1 + \beta b_1 = z^w \\ \dots\dots\dots \\ z_n + \beta b_n = z^w \end{cases} \quad (4.3)$$

donde

z_i : posición de la imagen f_i respecto del marco de referencia global (μm).

b_i : posición del centro de la marca en referencia a la imagen f_i (pixeles).

z^w : unidades del sensor que corresponden a la misma marca física (μm).

β : parámetro a estimar ($\mu m/pixeles$).

La incógnita del sistema de ecuaciones es el parámetro β .

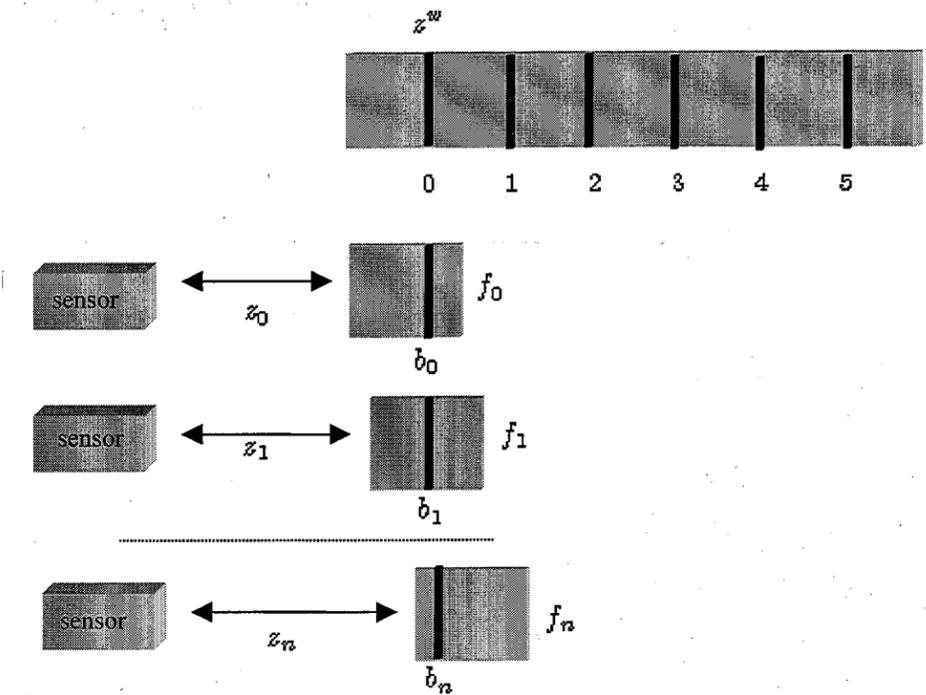


Figura 4-1: Imágenes con referencia a una misma línea z^w

4.1.1 Modelo Matemático

La ecuación (4.3) tiene como parámetros conocidos los vectores z y b , y como incógnita el factor de escala β . Idealmente debería de cumplirse la igualdad en un par de elementos de la ecuación (4.3), esto es: $z_i + \beta b_i = z_k + \beta b_k$, o equivalentemente la diferencia r_j entre un par de elementos debe ser igual a cero.

$$r_j = 0 = (z_k + \beta b_k) - (z_i + \beta b_i) \quad (4.4)$$

sin embargo esto no sucede en la práctica, para ello se plantea la función de costo de mínimos cuadrados descrita por la ecuación (4.5) que corresponde a utilizar la norma L_2 para pesar los residuos de los datos y cuya incógnita es el parámetro β .

$$U(\beta) = \sum_{\langle i,k \rangle \in C} (z_k + \beta b_k - (z_i + \beta b_i))^2 \quad (4.5)$$

El valor óptimo de β se encuentra minimizando la ecuación anterior

$$\min_{\beta} \left\{ U(\beta) = \sum_{\langle i,k \rangle \in C} (z_k + \beta b_k - (z_i + \beta b_i))^2 \right\} \quad (4.6)$$

donde C : es el conjunto de todas las combinaciones posibles de parejas de datos.

Derivando e igualando a cero la ecuación (4.6):

$$\begin{aligned} \frac{\delta U(\beta)}{\delta \beta} &= \frac{1}{2} \sum_{\langle i,k \rangle \in C} (z_k - z_i + \beta(b_k - b_i))(b_k - b_i) = 0 \\ &= \sum_{\langle i,k \rangle \in C} [(z_k - z_i) + \beta(b_k - b_i)](b_k - b_i) = 0 \\ &= \sum_{\langle i,k \rangle \in C} [-(z_i - z_k) + \beta(b_i - b_k)](b_i - b_k) = 0 \\ &= \sum_{\langle i,k \rangle \in C} [(z_i - z_k)(b_i - b_k)] = \sum_{\langle i,k \rangle \in C} \beta(b_i - b_k)^2 \end{aligned}$$

despejando β se obtiene finalmente la ecuación (4.7) que corresponde al factor de escala buscado.

$$\beta = \frac{\sum_{\langle i,k \rangle \in C} [(z_i - z_k)(b_i - b_k)]}{\sum_{\langle i,k \rangle \in C} (b_i - b_k)^2} \quad (4.7)$$

El proceso de calibración se realiza antes de la captura del video de la regla, en la práctica se elige una marca de la regla y sobre la misma marca se toma un conjunto de imágenes junto con sus respectivos valores del sensor. Con esta información se obtiene el parámetro de calibración β que posteriormente se requiere para hacer el ajuste de los datos como se describe en la siguiente sección. Para el cálculo de β se toman los valores absolutos de las diferencias. La implementación de la ecuación (4.7) se muestra en la sección 6 del

apéndice B.

4.2 Calibración Robusta

En la sección anterior se mostro la forma de estimar el factor de calibración β utilizando la norma L_2 . Como se describió en las secciones 3.4.2 y 3.4.3, el uso de la norma L_2 ocasiona que la estimación de los parámetros se sesgue cuando los datos están contaminados con "outliers". Para evitar en lo posible que la estimación de los parámetros se sesguen se utiliza estadística robusta.

La ecuación (4.5) corresponde a utilizar la función $\rho(x) = x^2$ para pesar los residuos de los datos, es decir se puede ver como un estimador- M . Eligiendo la función $\rho(x) = (x^2/2)/(1+x^2)$ como norma de error (ver apéndice A), se plantea la siguiente función de energía para la estimación robusta del parámetro β :

$$U(\beta) = \sum_{\langle i,k \rangle \in C} \rho(z_k + \beta b_k - (z_i + \beta b_i)). \quad (4.8)$$

El valor óptimo de β se encuentra minimizando la ecuación anterior

$$\min_{\beta} \left\{ U(\beta) = \sum_{\langle i,k \rangle \in C} \rho(z_k + \beta b_k - (z_i + \beta b_i)) \right\}, \quad (4.9)$$

donde C es el conjunto de todas las combinaciones posibles de parejas de datos. Minimizando la ecuación anterior se tiene:

Sea $\Delta z = z_i - z_k$ y $\Delta b = b_i - b_k$

$$\frac{\delta U(\beta)}{\delta \beta} = \sum_{\langle i,k \rangle \in C} \frac{\delta}{\delta \beta} \rho(\Delta z + \beta \Delta b) = 0 \quad (4.10)$$

donde $\frac{\delta}{\delta \beta} \rho(\Delta z + \beta \Delta b) = \frac{(\Delta z + \beta \Delta b) \Delta b}{[1 + (\Delta z + \beta \Delta b)^2]^2}$. Haciendo $w(\Delta z, \beta, \Delta b) = \frac{\frac{\delta}{\delta \beta} \rho(\Delta z + \beta \Delta b)}{(\Delta z + \beta \Delta b)}$ y multiplicando por $(\Delta z + \beta \Delta b)$ la ecuación anterior

$$\sum_{\langle i,k \rangle \in C} w(\Delta z, \beta, \Delta b)(\Delta z - \beta \Delta b) = 0 \quad (4.11)$$

despejando el parámetro β del último término

$$\beta = \frac{\sum_{\langle i,k \rangle \in C} w(\Delta z, \beta, \Delta b) \Delta z}{\sum_{\langle i,k \rangle \in C} w(\Delta z, \beta, \Delta b) \Delta b} \quad (4.12)$$

Obteniéndose finalmente la expresión iterativa para la estimación robusta del parámetro β .

$$\beta^{t+1} = \frac{\sum_{\langle i,k \rangle \in C} \frac{[\Delta b]}{[1+(\Delta z + \beta^t \Delta b)^2]^2} \Delta z}{\sum_{\langle i,k \rangle \in C} \frac{[\Delta b]}{[1+(\Delta z + \beta^t \Delta b)^2]^2} \Delta b} \quad (4.13)$$

La condición de paro se establece como $|\beta^{t+1} - \beta^t| < \varepsilon$. Como es usual se requiere una buena inicialización de β^0 . De manera práctica se elige $\beta = 5\mu m$ ó $\beta = 2\mu m$; el primer caso corresponde cuando la imagen contiene 4 marcas, el segundo caso se aplica cuando la imagen contiene una sola marca. En ambos casos corresponde a decir que una columna de la imagen tiene como ancho β micras del sensor. Para el cálculo de β se toman los valores absolutos de las diferencias. En la sección 7 del apéndice B se muestra el código fuente de la implementación de la ecuación (4.13). Una forma práctica de calcular el valor inicial β^0 es elegir una marca de la regla y adquirir la imagen y su respectivo valor del sensor z , después se desplaza una cierta distancia Δz y se adquiere otra imagen y su valor de sensor. Ambas imágenes deben hacer referencia a la misma marca de la regla. Con este pareja de datos se calcula: $\beta^0 = \Delta z / \Delta b$, que es una muy buena inicialización para detectar el valor óptimo de β .

4.3 Ajuste Final de Datos

En la presente sección se describe la parte final del proyecto que consiste en obtener la distancia a la que se encuentra cada línea procesada. La exposición se divide en tres partes: se plantea el problema de que una línea aparezca en distintas tomas de la imagen, después se describe de manera informal la forma de resolver dicho problema y por último se expone el algoritmo de *Clasificación de Puntos Finales* cuya función es determinar la posición global de las líneas procesadas.

Durante la exposición se utilizará un ejemplo sintético de datos que ayuda establecer conceptos y hace mas clara la explicación. Como parte final se propone la interpretación del resultado obtenidos.

4.3.1 Planteamiento

Dependiendo de los desplazamientos del carro sobre el que se encuentra montado el sistema de visión¹, un conjunto de líneas pueden aparecer en distintas imágenes. A menor velocidad de desplazamiento dicho conjunto aparecerá una cantidad mayor de veces en las imágenes adquiridas. En la figura 4-2 se muestra el caso donde tres líneas son referenciadas en tres imágenes. Cabe notar que la línea z_2^w aparece en las imágenes f_0, f_1, f_2 en distintas posiciones b_0, b_4 y b_8 respectivamente.

Usando la notación de la ecuación (4.3) se tiene el sistema de igualdades descrito en forma de tabla:

$$\begin{array}{lll} z_0 + \beta b_0 = z_0^w & z_1 + \beta b_3 = z_3^w & z_2 + \beta b_6 = z_6^w \\ z_0 + \beta b_1 = z_1^w & z_1 + \beta b_4 = z_4^w & z_2 + \beta b_7 = z_7^w \\ z_0 + \beta b_2 = z_2^w & z_1 + \beta b_5 = z_5^w & z_2 + \beta b_8 = z_8^w \end{array}$$

Aplicando el factor β se obtiene el vector Z^w que contiene toda la información sobre

¹Ver capítulo dos para mas detalle.

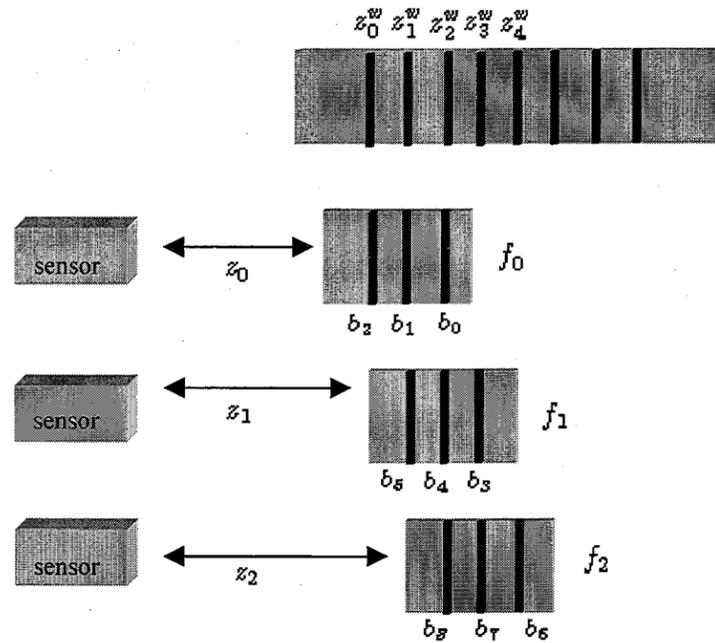


Figura 4-2: Correspondencia entre marcas

la posición global de las líneas.

$$Z^w = \{z_0^w, z_2^w, \dots, z_q^w\} \quad (4.14)$$

donde q indica el número total de marcas detectadas en todo el video procesado, para este caso $q = 8$. El objetivo ahora es determinar que componentes del vector Z^w corresponden a la misma línea, formando un cluster de datos.

4.3.2 Detección de Clusters

Antes de definir el algoritmo que detecta los clusters del vector Z^w descrito por la ecuación (4.14), se muestra de manera informal los pasos del algoritmo y posteriormente es definido.

Suponiendo que el vector de datos Z^w corresponde a las posiciones entre cada milímetro,

la gráfica del vector Z^w es un arreglo casi-ordenado de puntos como se describe en la figura 4-3, en ella se muestra el efecto que una línea de la regla aparezca en distintas tomas del video. La línea z_2^w es referenciada en las imágenes f_0, f_1 y f_2 , y los centros de las marcas son detectados en las posiciones b_0, b_4 y b_8 respectivamente. En la figura 4-3 se muestran dichos puntos marcados con negro.

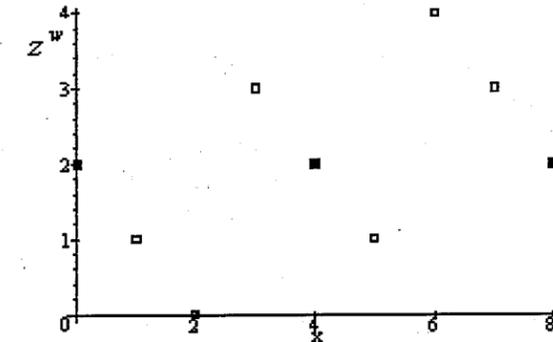


Figura 4-3: Vector Z^w sin ordenar

Para determinar los cluster se requiere realizar los siguientes pasos:

- Ordenar el vector Z^w de menor a mayor valor, en la figura 4-4 se muestra el resultado de ordenar el vector Z^w .
- Determinar los integrantes de cada cluster, donde el cluster k representa a una línea.
- Por cada cluster obtener su media μ y su desviación σ , el valor medio representa la distancia real en unidades del interferómetro de la marca de la regla y σ mide la dispersión de los datos.

Para la determinación de los cluster se aprovecha su estructura, como se nota en la figura 4-4 existe una clara división entre los cluster.

El siguiente algoritmo realiza la agrupación de los puntos y determina los dos primeros momentos estadísticos por cada cluster detectado. Como dato de entrada se tiene el

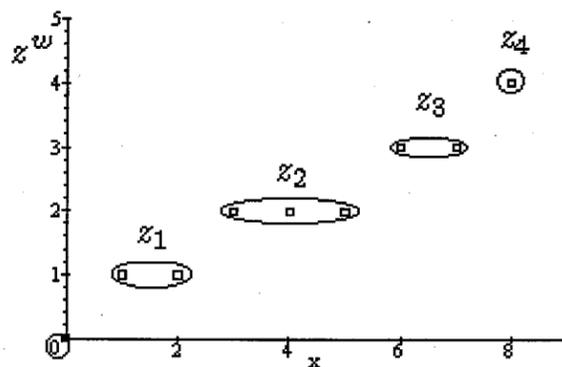


Figura 4-4: Vector Z^w ordenado

vector Z^w y como dato de salida se obtiene el número de clusters y para cada cluster se obtiene sus dos primeros momentos estadísticos, estos datos son representados por la ecuación (4.2).

Algoritmo 4 *Clasificación de Puntos Finales*

1. Ordenar de menor a mayor el vector Z^w .
2. Inicializar el cluster $C_0^0 = \{z_0^w\}$, $k = 0$.
3. Agregar elementos al cluster C_k mientras se satisfaga $z_j^w \in C_k \Leftrightarrow |C_k^0 - z_j^w| < \varepsilon$, $j = 1, \dots, |z^w|$ donde: C_k^0 es el primer elemento del cluster k .
4. Incrementar el número de clusters $k = k + 1$.
5. Inicializar el cluster C_k con el elemento z_j que no cumplió la condición del paso 3.
6. Realizar los pasos 3-5 mientras existan elementos por agregar.
7. Por cada cluster se obtiene su media μ_i y desviación σ_i formando $K = \{k_0, \dots, k_n\}$ donde $k_i = (\mu_i, \sigma_i)$.

La determinación del parámetro ε indica cual es la variabilidad que puede existir entre los componentes de un mismo cluster, dicho parámetro debe ser mucho menor que

la distancia entre cada marca. En la práctica se elige $\varepsilon = 30\mu m$. En la sección 8 del apéndice B se muestra el código fuente del algoritmo.

Este algoritmo tiene la propiedad de permitir identificar los "clusters" aun si las marcas medidas no están igualmente espaciadas. Esto permite ser aplicado en calibraciones donde por norma se analiza solo una muestra de las marcas de las reglas.

4.3.3 Post-Procesamiento

El resultado obtenido de aplicar el algoritmo *Clasificación de Puntos Finales* es un conjunto de parejas de datos que determinan la posición global de cada marca y la dispersión de los datos, representado por la ecuación (4.2). A partir de este conjunto de datos se puede realizar un análisis que determinará la calidad de la regla, por ejemplo:

- La diferencia entre el valor de la distancia nominal y la detectada.
- En base a los cruces por cero de la fase (ver sección 3.3.2), obtener la dispersión σ e identificar las líneas cuya desviación σ sea mayor que un umbral η , las líneas que no cumplan esta condición significa no estan bien definidas y eso puede corresponder a un problema de fabricación de la regla.
- Determinar las líneas cuya pendiente sea mayor que un umbral.
- Obtener la distancia entre líneas utilizando la información de la pendiente. Es decir, la distancia obtenida es multiplicada por el coseno del ángulo que forma con respecto a la horizontal.

La determinación del análisis final del conjunto de datos descrito por la ecuación (4.2) depende de las necesidades del usuario, en particular solo se requiere obtener la diferencia entre el valor nominal y el detectado por el procedimiento descrito en el presente trabajo.

Capítulo 5

Experimentos

En este capítulo se describe a detalle la aplicación de la técnica propuesta en los capítulos anteriores a algunos problemas en particular. El capítulo se divide en dos partes: la primera parte corresponde a un ejemplo sintético de una regla de $10mm$ de largo, para ello se desarrolla un modelo de la regla y sirve como banco de pruebas para validar el sistema de software en su totalidad. La segunda parte contiene tres ejemplos sobre datos reales; en el primer ejemplo se muestra una comparación del resultado obtenido por el ajuste lineal robusto y no robusto aplicado a 3 imágenes de distintos materiales. Como segundo ejemplo se tiene un caso de estudio que consiste en medir $20mm$ de una regla de cristal, dicha regla es medida por dos métodos; el método desarrollado en el presente trabajo y como lo realiza el CENAM. Como último experimento sobre datos reales se muestran las desviaciones detectadas sobre una muestra de 7 líneas de una regla de acero inoxidable, la calibración se realiza utilizando ambos métodos.

5.1 Experimento con Datos Sintéticos

Como se mencionó en el primer capítulo el proyecto desarrollado surge de un problema real que tiene el CENAM. Por diversas razones durante el desarrollo de los algoritmos no se contaba con un video de imágenes y sus respectivos valores del sensor que sirvieran para

validar el comportamiento del sistema en su totalidad, esto originó que se desarrollara un simulador de reglas patrón que sirviera como datos de prueba para validar los algoritmos desarrollados. El lenguaje utilizado fue *Builder C++*, ver. 4.0.

El resto de la sección consta de dos partes: se desarrolla el modelo de la regla y del sensor, con estos modelos se validan los algoritmos propuestos y se muestra el resultado obtenido.

5.1.1 Modelo de la Regla

Como primer paso se propone el modelo ideal de una regla de longitud n donde la distancia entre cada marca es constante 1.

$$X_{k+1} = X_k + 1 \quad k = 0, \dots, n-1 \quad (5.1)$$

Para aumentar la distancia entre cada marca de la ecuación anterior se introduce un factor de escala Δp que multiplica a cada miembro de la ecuación (5.1).

$$Y_k = X_k * \Delta p \quad k = 0, \dots, n-1 \quad (5.2)$$

El resultado obtenido es un vector de datos de longitud $\Delta p * n$ donde cada marca distinta de cero está separa por múltiplo de Δp . Esto se muestra en la figura 5-1.

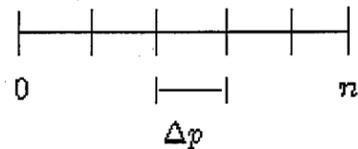


Figura 5-1: Regla ideal con n líneas.

El siguiente paso es proporcionar a cada línea de la regla un cierto ancho o contorno, esto equivale al zoom del microscopio que permite ver la marca de la regla mas ancha o

delgada. Esto se logra agregando a cada marca de la ecuación (5.2) una caja o región C que está centrada sobre Y_k , esto queda definido como:

$$Y_k = Y_k + C \quad k = 0, \dots, n-1 \quad (5.3)$$

La región C es simétrica, agrega a cada lado de Y_k el mismos ancho, esto se aprecia en la figura 5-2.

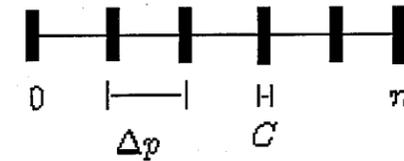


Figura 5-2: Diagrama de distancias entre marcas

El siguiente paso es convertir el vector de datos descrito por la ecuación (5.3) a una matriz R_{rp} donde $p = \Delta p * n$ el número de columnas y r el número de renglones. Cada renglón de la matriz es construido utilizando las ecuaciones (5.1-5.3), una vez construida la matriz se agrega ruido gaussiano con distribución $\eta \sim N(0, \sigma^2)$ sobre la matriz R . De esta forma se obtiene una matriz R que queda en función de cinco parámetros como se describe en la ecuación (5.4):

$$R(n, \Delta p, C, r, \sigma_m^2) \quad (5.4)$$

donde

- n : número de líneas
- Δp : distancia entre el centro de cada marca
- C : ancho de cada marca
- r : número de renglones
- σ_m^2 : varianza del ruido gaussiano

5.1.2 Modelo del Sensor

El modelo del sensor es sencillo, a cada columna de la matriz R descrita por la ecuación (5.4) le corresponde un valor del sensor z . Este valor z corresponde a múltiplos del factor β descrito en la sección 4.1. El modelo del sensor se describe por la ecuación (5.5) con la condición inicial $z_0 = 0$.

$$z_{k+1} = z_k + \beta + \eta \quad k = 1, \dots, n - 1 \quad (5.5)$$

donde

$\eta \sim U(0, 9)$ y β indica el ancho de cada columna de la imagen en micras. El ruido solo se agrega en las centenas de nanómetro, esto puede corresponder a la resolución del instrumento empleado.

Captura del Video Simulado

En base a la regla simulada R descrita por la ecuación (5.4) se requiere obtener una secuencia de video, descrito por la ecuación (2.1) donde f_i es una sub-matriz de $R_{r,q}$ con la restricción $q \ll \Delta p * n$, y se asocia a cada imagen una lectura del sensor z .

La simulación de la captura del video se basa en una idea sencilla, se extrae una sub-matriz f_i de la regla R y se le asocia la lectura del sensor z_i , donde el valor del sensor es tomado de la ecuación (5.5) que hace referencia a la primera columna de la sub-matriz, después se desplaza una cierta distancia Δs y se repite el proceso hasta recorrer toda la regla R . En la figura 5-3 se muestra esquemáticamente el proceso de captura de video.

De esta manera se obtiene una secuencia de video V con sus correspondientes lecturas del sensor Z que corresponden a los datos de entrada para el sistema de software como se describe en la sección 2.1.1.

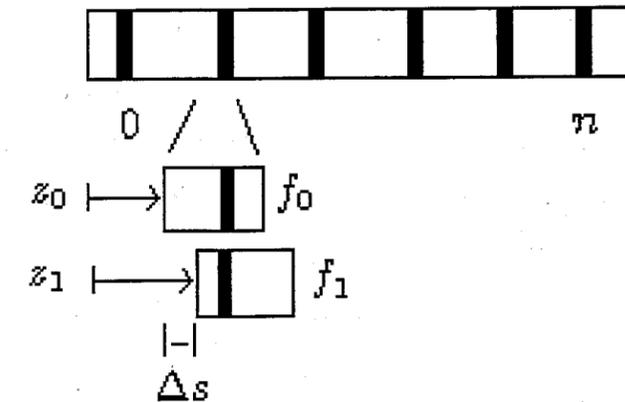


Figura 5-3: Proceso de captura imágenes

5.1.3 Resultados de la Simulación

Utilizando los modelos de la regla y del sensor con los parámetros mostrados abajo, se construyeron los datos de prueba..

- n : diez marcas σ_m^2 : 30
- Δp : 200 pixeles. β : $5\mu m$
- C : 10 pixeles. Δs : 15 pixeles
- r : 30 renglones

En la figura 5-4 se muestra el resultado de la simulación de la regla obtenida usando los parámetros anteriores. De esta regla se obtiene un video V donde cada imagen f_i es una sub-región de la figura 5-4.

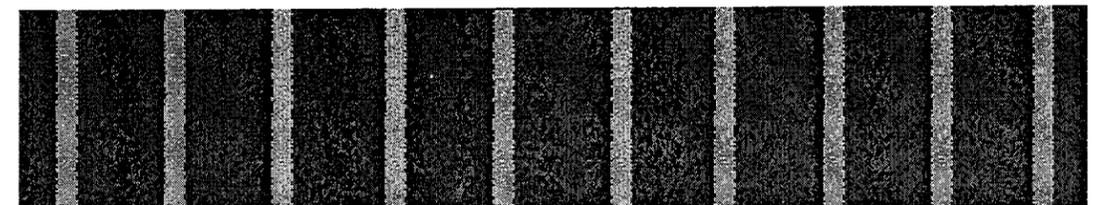


Figura 5-4: Regla simulada

Calibración

El video obtenido contiene 54 imágenes con sus respectivos valores del sensor, antes de procesar el video se requiere obtener el factor de calibración β . Para ello se seleccionó la primera línea de la imagen y en base de ella se obtuvieron ocho imágenes, como se describe en la sección 4.1.1. Los valores obtenidos del sensor y los centros detectados se muestran en la tabla siguiente:

Datos simulados para la calibración

$Z(\mu m)$	$b's(pixeles)$
605.5	78.7108
635.5	72.3134
675.4	64.7104
720.3	55.0190
755.2	48.9188
785.6	42.7103
815.2	36.4171
865.7	26.2103

Con base a estos datos y utilizando las ecuaciones (4.7) y (4.13) se obtuvo el factor de calibración por el método no robusto (NR) y robusto (R), respectivamente. Los valores obtenidos son:

$$\beta_{nr} = 5.00489\mu m \quad \beta_r = 5.00482\mu m$$

Como era de esperar ambos resultados coinciden con el resultado esperado que es de $5\mu m$. Agregando un outlier en la detección del centro de la marca al cuarto renglón de la tabla anterior, se tiene $b_4 = 55.7100 + 20$. Con base a esta modificación se calcula el factor β por el método robusto y no robusto obteniéndose:

$$\beta_{nr} = 4.3393\mu m \quad R : 4.9968\mu m$$

Tomando la diferencia entre el valor obtenido y el ideal $E = |\beta_{ideal} - \beta_{obtenida}|$ se tiene:

$$E_{nr} = 0.6606\mu m \quad E_r = 0.0031\mu m$$

Como se nota el error detectado por el método robusto tiene un mejor desempeño que el método no robusto. En lo subsecuente se utiliza el método robusto para la estimación del factor de calibración β .

Una manera gráfica de ver cuál es el dato que esta ocasionando que se sesgue la estimación, es guardar los valores de los pesos $w(\Delta z, \beta, \Delta b)$ en una matriz $W_{n,n}$. Como se describe en las ecuaciones (4.13) y (4.12), estos valores w son calculados para cada par de elementos i, k . Con base a estos datos y almacenandolos en la matriz W , se obtiene una matriz diagonal superior, con ceros en la diagonal principal. Para ver como está estructurada la matriz W supongase que se tienen un conjunto con cuatro elementos $\{x_1, x_2, x_3, x_4\}$ sobre los cuales se quieren formar todas las combinaciones posibles de dos elementos, para este caso se tiene 6 combinaciones posibles que son: $(x_1, x_2), (x_1, x_3), (x_1, x_4), (x_2, x_3), (x_2, x_4), (x_3, x_4)$. Estos datos se pueden representar en forma matricial como:

$$W = \begin{pmatrix} 0 & (x_1, x_2) & (x_1, x_3) & (x_1, x_4) \\ 0 & 0 & (x_2, x_3) & (x_2, x_4) \\ 0 & 0 & 0 & (x_3, x_4) \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.6)$$

donde $w_{i,j} \geq 0$. Obteniendo la suma por renglones de W se obtiene un arreglo de datos v_w , que en caso de que los datos no estén contaminados con outliers la gráfica obtenida corresponde a un función decreciente en forma de rampa. Para fines de visualización es conveniente obtener la transpuesta de la matriz W , y sumando por renglones se obtiene una arreglo de datos cuya gráfica es una función creciente.

En la figura 5-5 se muestra la gráfica de los valores obtenidos de la función de peso w , utilizando los datos de la tabla anterior.

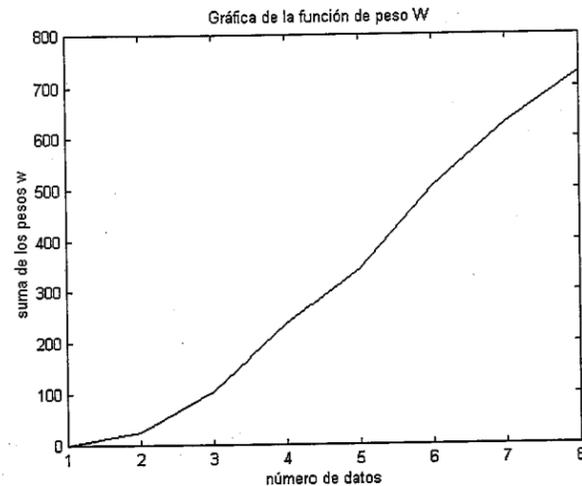


Figura 5-5: Gráfica de la función de peso W

Esta gráfica se interpreta como que los datos utilizados no estaban contaminados con outliers. Agregando un outlier al renglón 4 de la tabla anterior se tiene $b_4 = 55.0190 + 20$, entonces se obtiene la gráfica de la figura 5-6. Notese como en la gráfica la posición 4 la gráfica se va hacia cero, esto indica que el dato asociado a dicha posición corresponde a un outlier.

Procesamiento del Video

Procesando el video obtenido como se describe en la sección 4.2 se obtiene el vector de datos Z^w descrito por la ecuación (4.14) y utilizando el algoritmo de *Clasificación de Puntos Finales* se obtiene el resultado final que es un vector K de datos descrito por la ecuación (4.2). El error o la desviación entre el valor nominal y el obtenido se describe por la ecuación (5.7).

$$E = P_{\text{nominal}} - P_{\text{obtenida}} \quad (5.7)$$

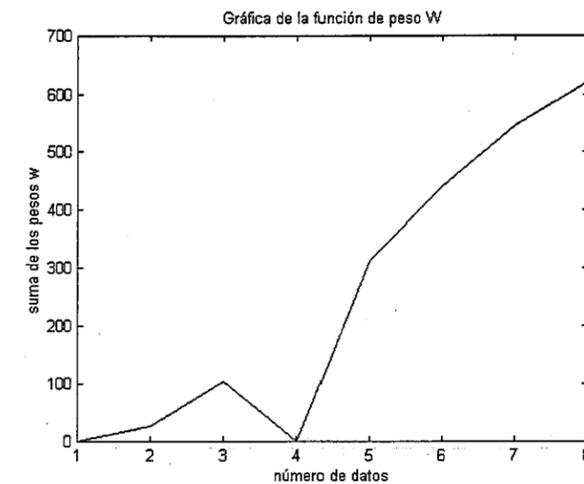


Figura 5-6: Gráfica de la función de peso W

Usando la ecuación anterior con los valores promedios de la distancias donde fueron detectadas las líneas, se obtiene un arreglo de datos que se muestra en la figura 5-7. El error detectado es del orden de decimas de micra.

Como se mostró en esta sección los resultados obtenidos concuerdan con lo esperado y con ello se valida el funcionamiento del software en su conjunto. En la práctica se realizaron pruebas de reglas simuladas de hasta $3m$ de longitud obteniendo resultados favorables. En la siguiente sección se muestra el caso de estudio y los resultados obtenidos.

5.2 Datos Reales

En esta sección se muestran los resultados obtenidos para datos reales. Los primeros resultados corresponden al ajuste robusto y no robusto de regresión lineal, el segundo corresponde al caso de estudio que consiste en la medición de las líneas 26 a la 46 de una regla de cristal de $200mm$.

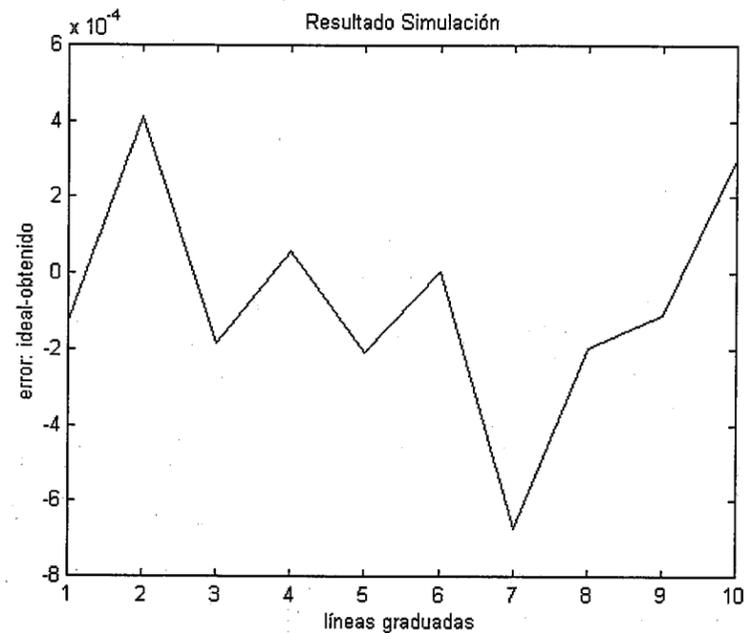


Figura 5-7: Gráfica de las desviaciones de la simulación.

5.2.1 Ajuste Robusto

En la figura 5-8 se muestran 3 parejas de imágenes. En la parte izquierda se muestran los puntos que corresponden a los cruces por cero de la fase que fueron obtenidos por el algoritmo-2 descrito en la sección 3.2. Con base a este conjunto de puntos se requiere ajustar de forma robusta la ecuación de una recta. Utilizando el algoritmo-3 (ver sección 3.4.5) se obtuvo de forma robusta las rectas mostradas en la parte derecha de las imágenes. En forma de tabla se muestran los parámetros obtenidos del método robusto (R) y no robusto (C).

Tradicionalmente solo interesa el parámetro b . Considerando que el factor de escala $\beta = 5\mu m$, la diferencia entre el método robusto y no robusto para la regla de acero es: $7.9095\mu m$, lo cual es significativo para los propósitos de medición de reglas.

	m	b
R	0.006221	116.83979
C	0.006221	116.83979

	m	b
R	-0.00876	51.5750
C	-0.00025	53.1569

	m	b
R	-0.012007	157.98168
C	0.070257	173.64783

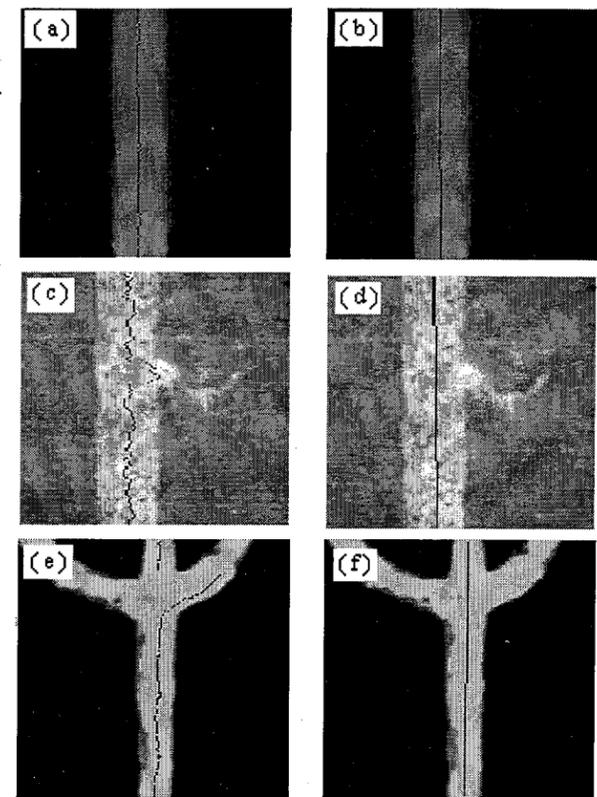


Figura 5-8: (a) y (b) Regla de cristal. (c) y (d) Regla de Acero. (e) y (f) Cinta de acero flexible.

5.2.2 Calibración de una Regla de Cristal

La prueba sobre datos reales se realizó sobre una regla de cristal. Este es el tipo de reglas de mejor calidad (ver figura 5-9). El experimento consistió en tomar una muestra de datos sobre las líneas 26 a la 46, la distancia entre cada línea es de $1mm$. El zoom del microscopio permitía ver solo una línea por imagen, como se muestra en la figura 5-9. El proceso de captura de datos se realizó cuatro veces obteniendo un total de 84 imágenes con sus respectivos valores del sensor. El procesamiento utilizado para la mejora de la imagen consistió en este caso en una simple inversión de tonos, como se describe en la sección 3.2.1.

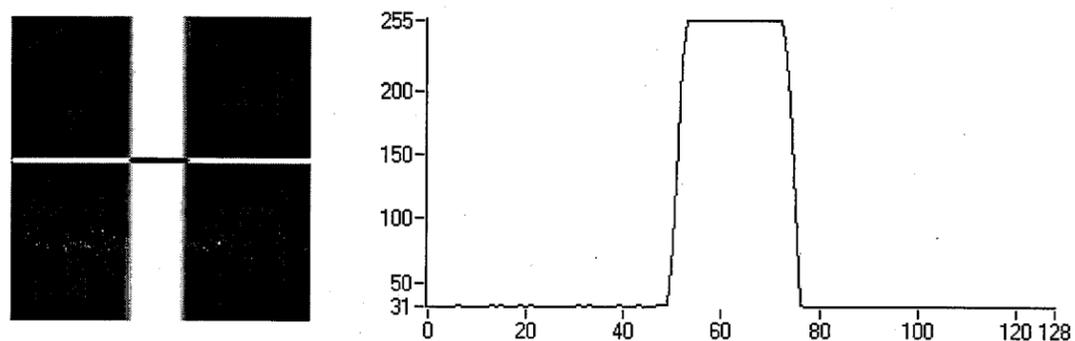


Figura 5-9: Imagen de una regla de cristal.

Calibración

Para calcular el factor de calibración β se tomó una muestra de 8 imágenes. Dada la posición en que se colocó la regla a calibrar, los datos del sensor van en sentido descendente, como se muestra en las siguientes tablas. Los valores del sensor Z mostrados corresponden a los datos originales. Dichos valores Z los llamaremos como la posición global relativa, debido a que para fines de visualización deben de estar escalados hacia un origen común.

En la siguiente tabla se muestran los valores obtenidos del sensor (mm) y la ubicación local ($pixels$) de los centros $b's$ detectados por el algoritmo-3 descrito al final del capítulo 3.

Datos reales para la calibración			
Z	$b's$	Z	$b's$
151.9354	65.5582	151.6418	319.5246
151.8442	121.5114	151.5863	279.0688
151.7509	179.0971	151.5216	377.8163
151.7133	201.8626	151.4252	245.7945

Con base a los datos de la tabla de arriba, se calculó el factor de calibración robusto β_r y no robusto β_{nr} como se describe en las secciones 4.1 y 4.2. Obteniendo como

resultado:

$$\beta_r = 1.6340\mu m \quad \beta_{nr} = 1.6340\mu m$$

Los resultados obtenidos prácticamente son iguales. En la figura 5-10 se muestra gráficamente el vector de pesos w que indica como fueron pesados los datos.

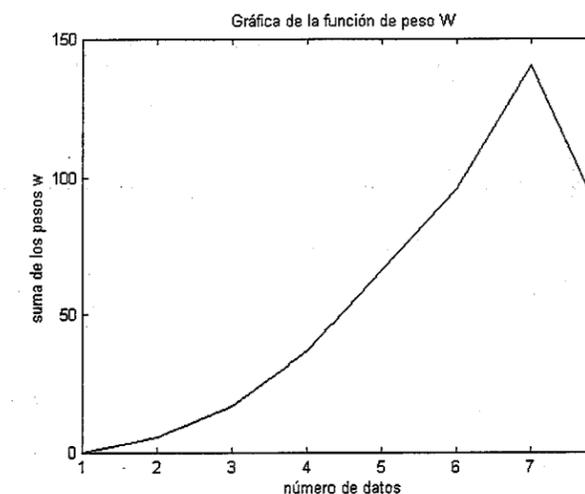


Figura 5-10: Gráfica de los pesos W de la calibración para datos reales.

Notese como el dato correspondiente a la octava posición de la tabla anterior fue pesado menos. Eliminando este dato, se calcula el factor β por el método robusto y no robusto obteniéndose la estimación: $\beta=1.6338\mu m$ para ambos casos. Este resultado difiere de la estimación anterior a partir de la tercera cifra decimal.

Para este caso los resultados obtenidos por el método robusto y no robusto prácticamente son iguales, esto se debe en gran medida a que el diseño del experimento se realizó con gran cuidado y la definición de la imagen es de buena calidad.

Detección de centro y resultado final

Procesando las 84 imágenes y los datos del sensor como se describió en los capítulos anteriores se obtuvo finalmente el vector de datos K descrito por la ecuación (4.2). Como

ejemplo del ajuste obtenido se muestran los datos obtenidos para las líneas 26-29 de la regla de cristal. Cada tabla contiene los datos correspondientes a cada una de los cuatro mediciones realizadas sobre un línea en particular. Los valores del sensor Z corresponden a la posición global relativa.

Valores de la línea 26			Valores de la línea 27		
$z + \beta b = z^w$			$z + \beta b = z^w$		
Exp1	$151.6318 + \beta 251.716$	$=152.04330$	Exp1	$150.6193 + \beta 259.177$	$=151.04305$
Exp2	$151.6236 + \beta 256.648$	$=152.04288$	Exp2	$150.6054 + \beta 267.324$	$=151.04247$
Exp3	$151.4184 + \beta 381.854$	$=152.04263$	Exp3	$150.9466 + \beta 58.244$	$=151.04184$
Exp4	$151.6908 + \beta 215.805$	$=152.04360$	Exp4	$150.3766 + \beta 407.165$	$=151.04226$

Valores de la línea 28			Valores de la línea 29		
$z + \beta b = z^w$			$z + \beta b = z^w$		
Exp1	$149.6023 + \beta 269.446$	$=150.0427$	Exp1	$148.6135 + \beta 262.521$	$=149.0426$
Exp2	$149.5998 + \beta 271.253$	$=150.0432$	Exp2	$148.6386 + \beta 247.093$	$=149.0425$
Exp3	$149.8176 + \beta 137.356$	$=150.0422$	Exp3	$148.5115 + \beta 325.199$	$=149.0431$
Exp4	$149.4243 + \beta 378.136$	$=150.0424$	Exp4	$148.4713 + \beta 349.640$	$=149.0428$

El resultado obtenido es un vector de datos Z^w descrito por la ecuación (4.14) que corresponde a la columna de la derecha de cada tabla. Con base al vector Z^w y utilizando el algoritmo-4 (ver sección 4.3) se obtiene el valor medio μ y la dispersión σ para cada "cluster" detectado. Los valores promedios de las 26-29 se muestran en la segunda columna de la tabla siguiente, estos valores corresponden a la posición global relativa. Tomando como sistema de referencia a la línea 26 los datos son mapeados al origen obteniendo la tercera columna de datos.

Posición	Valor promedio	Valor detectado	Valor nominal	Error μm
26	152.0431	26.0000	26.0	0
27	151.0424	27.0007	27.0	-0.7002
28	150.0426	28.0004	28.0	-0.4244
29	149.0428	29.0002	29.0	-0.2887

La cuarta columna de datos corresponde a la posición nominal. Usando la ecuación (5.7) se obtienen los errores entre la posición nominal y la obtenida, esto se muestra en la columna más a la derecha. El error obtenido es del orden de décimas de micras.

Dado que la regla medida es de cristal y las marcas son de la mejor calidad, la detección del centro de cada marca está bien definida y ello contribuyó al decremento de la incertidumbre en la medición.

Uno de los objetivos de las tesis fue el mejorar el proceso de medición de reglas en el CENAM, para validar tal requisito la regla en cuestión fue medida en el CENAM expidiendo un certificado de medición que muestra las desviaciones detectadas. Este certificado se muestra en el apéndice C. La medición se realizó sobre los 200mm y la gráfica de dichas desviaciones se muestra en la figura 5-11.

Resulta interesante ver que al principio de las mediciones las desviaciones detectadas eran del orden de décimas de micras, de igual magnitud que las obtenida por el método propuesto. Conforme se avanzó en la medición las desviaciones crecen hasta 10 veces. El experimento se realizó 3 veces.

Para comparar los resultados de la calibración del CENAM y el método propuesto, se tomaron las desviaciones detectadas por el método del CENAM de la línea 26 a la 46 y se re-escalaron tomando la línea 26 como origen, esto se muestra en la figura 5-12 con una raya semi-continua. Las línea continua corresponde al resultado obtenido por el método propuesto. Como se nota en la figura 5-12 con el método propuesto se detectó una menor variabilidad en los datos y no se detecto un aumento en las desviaciones detectadas como lo presenta el CENAM. En la tabla siguiente se muestra la varianza de los datos y el valor pico para las líneas 26-46, a ambos resultados previamente se les quito su valor medio.

Es decir $E = P_n - P_o - \left(\frac{P_n - P_o}{k} \right)$ y en base a E se calculó la varianza y valor pico.

	Varianza μm	Valor Pico μm
CENAM	0.21522	+0.93809
Tesis	0.41573	-0.44518

En la figura 5-13 se muestra en forma de tabla las desviaciones detectadas entre el valor nominal y el obtenido, del método propuesto. Un factor que influye en la buena ubicación de la posición global de la línea es el factor de escala β , esto se debe a que si el zoom del microscopio es grande entonces solo es visible una marca en la imagen. Con ello la distancia entre el centro de cada marca es grande y por la relación $\beta = \frac{1000}{k}$ donde k indica la distancia en pixeles entre los centros de dos marcas y la constante 1000 es un factor de amplificación para expresar las micras. El valor de β decrece cuando k crece y ello ocasiona que β adquiera valores de nanómetros. Esto permite un margen de error de unos cuantos pixeles en la ubicación local de la marca.

Durante el proceso de calibración del CENAM las imágenes contenían 4 marcas de la regla de cristal, el factor de calibración aproximado es: $\beta = 6.45 \mu m$. Durante el proceso de medición que realizamos el factor de calibración obtenido: $\beta = 1.63 \mu m$. Recordando que el factor β equivale al ancho de un pixel columna de la imagen, con ello una mala detección del centro de la marca o del canto, es equivalente a decir que por cada k columna pixel que se equivoque el método de calibración de reglas¹, el error crece β -veces. Esta diferencia de factor de escala hace que los errores cometidos por el método del CENAM sean mayores que por el método propuesto, debido a que el factor de calibración utilizado es aproximadamente 4 veces mayor al empleado por el método propuesto.

Utilizando la detección del centro de la marca como se describe en el primer método del capítulo uno, se obtienen resultados similares al obtenido usando el método propuesto.

¹El método del CENAM o el propuesto.

Esto se debe en gran medida a que los datos de prueba están bien definidos. Sin embargo dichos métodos no ofrecen información adicional como la descrita en la sección 4.3.1.

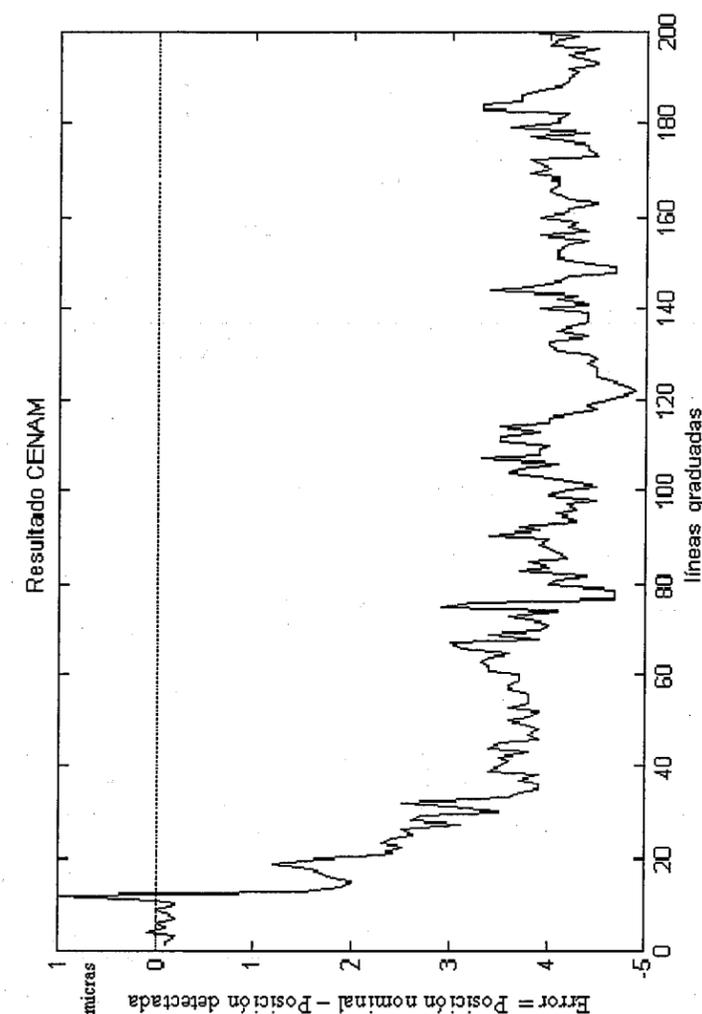


Figura 5-11: Resultado de Calibración del CENAM

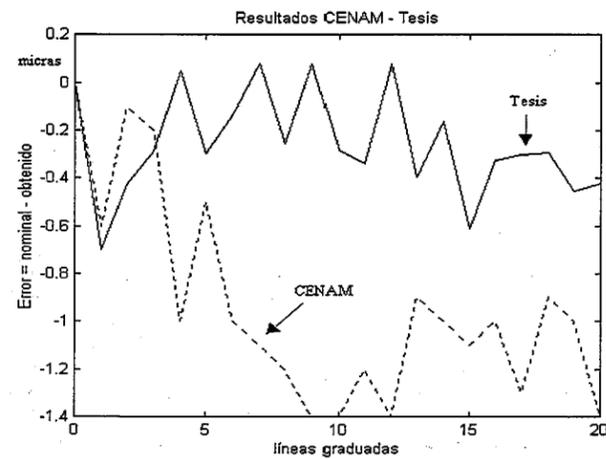


Figura 5-12: Resultados de las desviaciones detectadas en la regla de cristal

Desviaciones medidas de la línea 26 a la 46

Valor nominal mm	Valor detectado mm	Desviación micras
26	26.000000	0
27	27.000700	-0.700
28	28.000424	-0.424
29	29.000288	-0.288
30	29.999949	0.050
31	31.000295	-0.295
32	32.000134	-0.134
33	32.999916	0.083
34	34.000254	-0.254
35	34.999921	0.078
36	36.000281	-0.281
37	37.000336	-0.336
38	37.999918	0.081
39	39.000395	-0.395
40	40.000162	-0.162
41	41.000613	-0.613
42	42.000324	-0.324
43	43.000301	-0.301
44	44.000290	-0.290
45	45.000453	-0.453
46	46.000423	-0.423

Figura 5-13: Desviaciones detectadas por el método propuesto

5.2.3 Calibración de una Regla de Acero Inoxidable

Uno de los inconvenientes del resultado anterior es que no se tomó un origen en común, el CENAM tomó como origen el milímetro 1 y nosotros el milímetro 23. Para satisfacer tal requisito se midieron 7 líneas de una regla de acero inoxidable en los milímetros: 1, 2, 5, 10, 50, 100, 200, tomando como origen común el milímetro 1. Sobre dichos puntos se adquirieron 3 videos con sus respectivos valores de sensor z, teniendo en total una muestra de 35 datos. El CENAM realizó la medición 3 veces sobre el mismo conjunto de puntos. En la figura 5-14 se muestra una imagen de la regla de acero inoxidable a calibrar, la línea en blanco corresponde al centro de la marca detectado por el método propuesto.

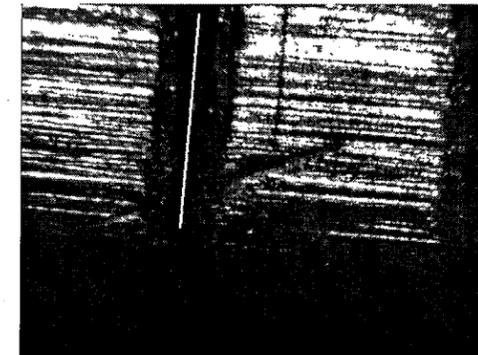


Figura 5-14: Imagen de una regla de acero inoxidable

Cabe mencionar que este tipo de reglas son las más comunes y como se nota en la figura anterior la determinación del centro de la marca de forma visual no es problema trivial. El procesamiento utilizado para la mejora de la imagen consistió en este caso en invertir los tonos de gris y elevar a un potencia cuadrada, como se describe en la sección 3.2.1.

Calibración

Para calcular el factor de calibración β se tomó una muestra de 8 imágenes sobre el milímetro 7. En la tabla siguiente se muestran los valores obtenidos del sensor (mm) y la ubicación local ($pixeles$) de los centros $b's$ detectados por el algoritmo-3 descrito al final del capítulo 3.

Datos reales para la calibración

Z	$b's$	Z	$b's$
7.7274	446.8637	8.2624	231.9116
7.7971	419.3237	8.3669	189.8658
7.9195	370.9302	8.4932	138.1004
8.1620	273.3442	5.5946	99.3496

Con base en los datos de la tabla de arriba se calculó el factor de calibración robusto β_r y el no robusto β_{nr} (ver secciones 4.1 y 4.2). Obteniéndose como resultado:

$$\beta_r = 2.4799\mu m \quad \beta_{nr} = 2.4812\mu m$$

Los resultados obtenidos prácticamente son iguales. Tomando el valor absoluto de la diferencia entre ambos resultados se tiene: $E = 0.001285\mu m$, esto indica que los datos adquiridos durante el proceso de calibración no contienen outliers.

Resultado Final

Procesando las 35 imágenes y los datos del sensor como se describió anteriormente, se obtuvo la siguiente tabla de datos insertada en la figura 5-15. Dicha tabla contiene las desviaciones detectadas por el procedimiento del CENAM y el propuesto.

Los valores de la tabla mostrada en la figura 5-15 corresponden a utilizar la ecuación (5.7). Graficando los valores de la tabla de las desviaciones, se tiene la figura 5-16.

Milímetro	Desviación micras Tesis	Desviación micras CENAM
1	0	0
2	-33.4	-18.4
5	-17.3	-31.7
10	-15.4	-20.3
50	2.1	-21.5
100	-3.4	-19.4
200	-35.3	-37.9

Figura 5-15: Tabla de desviaciones detectadas CENAM - Tesis

Resulta interesante ver que en el milímetro 50 que corresponde a la quinta posición sobre el eje x , el error detectado por el método del CENAM y el propuesto difieren por $-19.4\mu m$, lo cual es significativo para propósitos de medición en reglas patrón. Con base a esta gráfica se puede concluir que el método propuesto es aplicable para calibrar reglas de acero inoxidable. Un mejor resultado podría obtenerse si se mejora la iluminación durante la adquisición de datos.

Con base en los valores de las desviaciones detectadas por ambos métodos, como se muestra en la figura 5-16, y quitándoles su valor medio respectivamente, se obtiene el siguiente cuadro de datos que muestra la varianza de los datos y el valor pico.

	Varianza μm	Valor Pico μm
CENAM	11.92	+21.36
Tesis	15.33	-20.68

En las figuras 5-19 y 5-18 se muestran las desviaciones detectadas utilizando el método propuesto y como lo realiza el CENAM, respectivamente. En cada imagen se muestran tres líneas, cada línea representa las desviaciones detectadas en cada uno de las 3 mediciones realizadas sobre la muestra de 7 líneas de la regla de acero inoxidable. La línea continua corresponde al primer conjunto de datos, la línea continua a trozos corresponde al segundo conjunto de datos y la línea punteada al tercer conjunto de datos. En la figura 5-17 se muestra en forma de tabla las desviaciones E_{ij} detectadas en cada experimento.

Nótese la repetibilidad de las desviaciones detectadas por el método propuesto como se muestran en la figura 5-19. Para comparar la variabilidad detectada por ambos métodos, se define:

$$\sigma_T = \frac{1}{n} \sum_{i=1}^n (\sigma_i) \quad (5.8)$$

$$\sigma_i = \sqrt{\frac{1}{m-1} \sum_{j=1}^m (E_{ij} - \mu_j)^2}$$

donde

n : número de renglones, $n = 7$.

m : número de columnas, $m = 3$.

$$\mu_j = \frac{1}{m} \sum_{i=1}^m E_{ij}$$

E_{ij} : corresponde al error detectado en el experimento j en la línea graduada i .

Con base a la tabla de datos de la figura 5-19 y utilizando la ecuación (5.8) se obtuvo para el método propuesto $\sigma_T = 1.7369$ y el CENAM $\sigma_T = 5.3077$, lo cual significa que el método del CENAM tiene una mayor variabilidad en las mediciones. El método propuesto reporta una menor variabilidad indicando con ello la repetibilidad en las desviaciones detectadas.

Con los resultados obtenidos utilizando el modelo sintético y con los datos reales, se concluye que el método propuesto mejora la calidad de calibración de las reglas, en comparación con el CENAM.

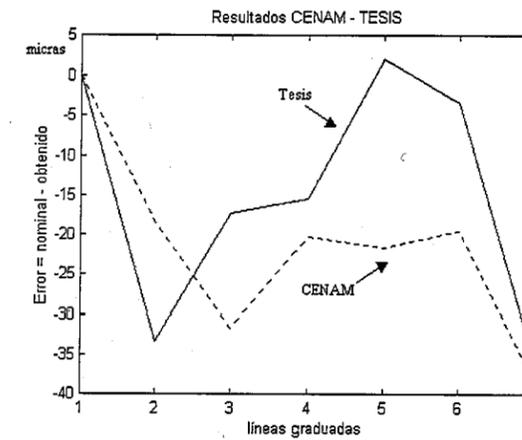


Figura 5-16: Resultado de la medición de la regla de acero inoxidable

Línea	Exp1 Desviación micras	Exp2 Desviación micras	Exp3 Desviación micras
1	0	0	0
2	-9.8	-25.2	-20.2
5	-39.6	-28.4	-27.1
10	-19.0	-23.3	-18.6
50	-30	-21.5	-13.1
100	-30.7	-17.4	-10.2
200	-37	-38.1	-38.8

Línea	Exp1 Desviación Micras	Exp2 Desviación micras	Exp3 Desviación micras
1	0	0	0
2	-33.6	-34.2	-32.4
5	-17.0	-17.2	-17.5
10	-14.9	-18.7	-13.2
50	4.8	3.5	2.8
100	-0.8	-4.4	-3.7
200	-38.4	-39.3	-29.8

Figura 5-17: Desviaciones detectadas en cada experimento

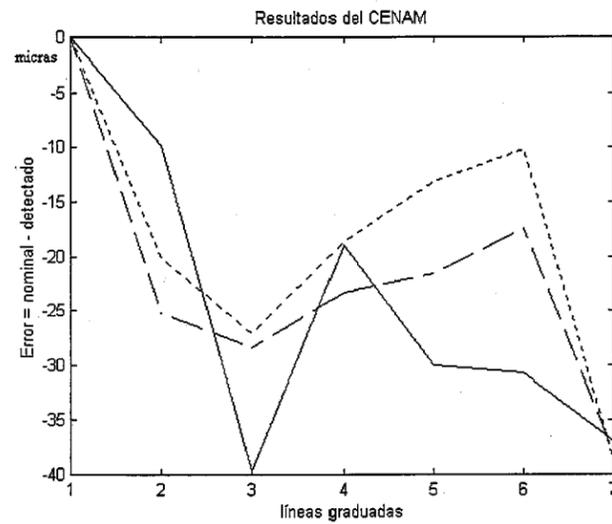


Figura 5-18: Desviaciones detectadas por el CENAM para cada experimento

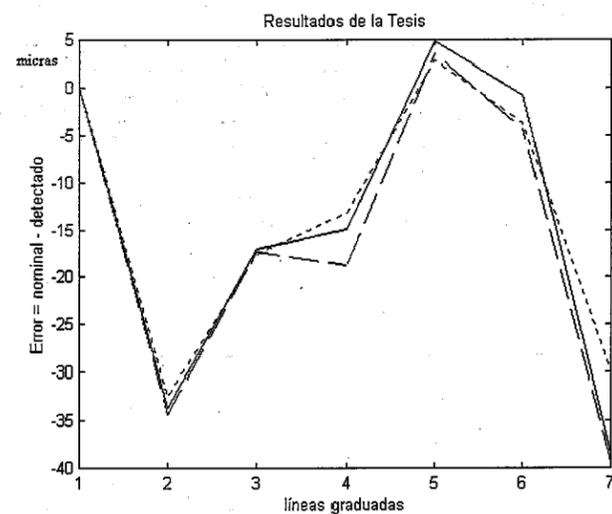


Figura 5-19: Desviaciones detectadas por el método propuesto para cada experimento

Capítulo 6

Conclusiones

En este trabajo se presentó un nuevo método automático fuera de línea basado en un sistema de visión para medir la distancia entre líneas graduadas. Su campo de aplicación básicamente es la medición de reglas y cintas patrón. El método propuesto es invariante al tipo de material de fabricación y la detección del centro de la marca es robusta a la presencia de outliers.

El proceso de calibración de reglas patrón es una actividad de gran importancia a nivel mundial, debido a que con ellas otros sistemas longitudinales de menor precisión son calibrados. La calibración o medición de las reglas es requerida por la industria nacional, debido a que deben de estar certificados bajo una norma internacional para que sus productos manufacturados puedan competir con los estándares internacionales. Esta actividad se realiza en el Centro Nacional de Metrología CENAM.

Los métodos encontrados en la literatura para medir la distancia entre líneas graduadas, se pueden considerar como un caso particular del método propuesto. La generalización consiste en procesar la imagen 2-D obteniendo más información acerca del estado y ubicación local de la marca, en los métodos propuestos se considera que la regla es un arreglo 1-D, ocasionando que se tengan menos puntos de apoyo y por lo tanto se pueda incurrir en errores para la determinación del centro de la marca.

Los algoritmos propuestos en el presente trabajo arrojan nueva información acerca

del estado del objeto a medir, ejemplo de ello es la pendiente de la línea y el estado físico de la línea.

La ventaja de utilizar un sistema de visión es la generalidad que se logra debido a que el trabajo recae sobre el diseño de algoritmos y no sobre el diseño de nuevos sistemas de hardware para realizar la medición, estos resultan costosos y cada vez que se cambia de material se debe de ajustar el sistema electrónico.

En el capítulo 5 se mostró el caso de estudio de donde se puede concluir que el método propuesto obtiene resultados que mejoran los obtenidos por el procedimiento actual de calibración del CENAM.

Como trabajo futuro se tienen dos propuestas: la medición del ancho de las líneas graduadas, esta es una medida más de calidad y un nuevo método de medición que tome en cuenta la inclinación de la línea graduada y su ancho. Así mismo se considera la posibilidad de implementar el sistema desarrollado para que funcione en tiempo real.

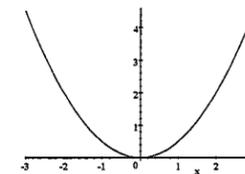
Apéndice A

Estimadores-M

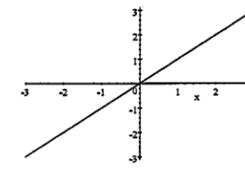
En este apéndice se muestran las funciones de los estimadores-M mas frecuentemente utilizados en regresión lineal robusta y visión computacional [16],[15],[13]. La elección del mejor estimador es un problema abierto. Para el caso de regresión lineal si se conoce la distribución del ruido se puede seleccionar un estimador que minimice la influencia de los outliers, sin embargo este no es el caso general.

Como se mencionó en la sección 3.2, un estimador-M se puede caracterizar por su función de influencia $\Psi(x)$ que indica como son pesados las diferencias entre el modelo y las observaciones, por ello es frecuente seleccionar el estimador partiendo de las función $\Psi(x)$, luego integrando se obtiene $\rho(x)$ que es la función del estimador-M.

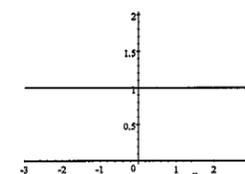
Nombre :Norma L_2



$$\rho(x) = \frac{x^2}{2}$$

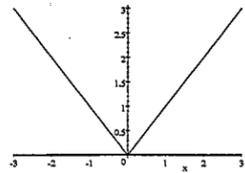


$$\Psi(x) = x$$

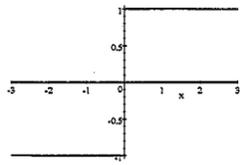


$$w(x) = 1$$

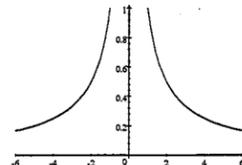
Nombre: Norma L_1



$$\rho(x) = |x|$$

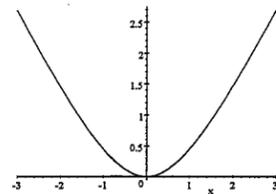


$$\Psi(x) = \text{sgn}(x)$$

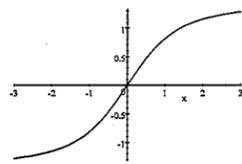


$$w(x) = \frac{1}{|x|}$$

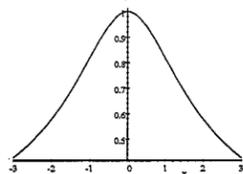
Nombre: Norma $L_1 - L_2$



$$\rho(x) = 2\sqrt{1+x^2/2} - 2$$

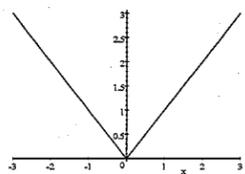


$$\Psi(x) = \frac{x}{\sqrt{1+x^2/2}}$$

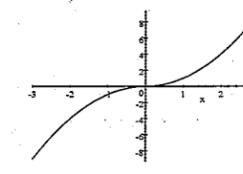


$$w(x) = \frac{1}{\sqrt{1+x^2/2}}$$

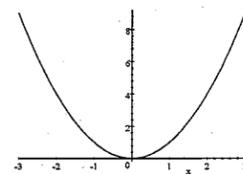
Nombre: Norma L_p



$$\rho(x) = \frac{|x|^p}{p}$$

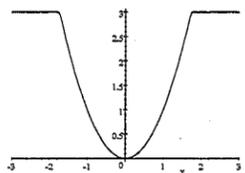


$$\Psi(x) = \text{sgn}(x)|x|^{p-1}$$

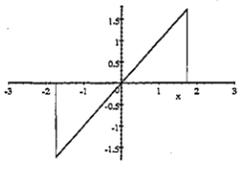


$$w(x) = |x|^{p-1}$$

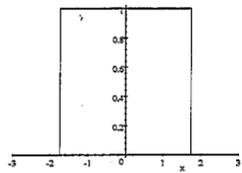
Nombre: Cuadratica Truncada $\begin{cases} \text{if } x^2 \leq k \\ \text{if } x^2 > k \end{cases}$



$$\rho(x) = \begin{cases} \frac{x^2}{2} \\ k \end{cases}$$

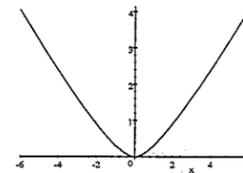


$$\Psi(x) = \begin{cases} x \\ 0 \end{cases}$$

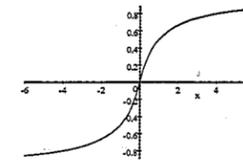


$$w(x) = \begin{cases} 1 \\ 0 \end{cases}$$

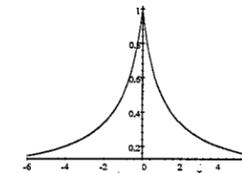
Nombre: Fair



$$\rho(x) = c^2 \left[\frac{|x|}{c} - \log \left(1 + \frac{|x|}{c} \right) \right]$$

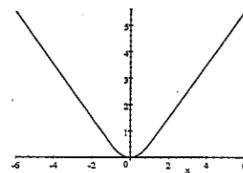


$$\Psi(x) = \frac{x}{1+|x|/c}$$

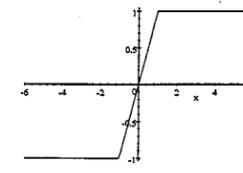


$$w(x) = \frac{1}{1+|x|/c}$$

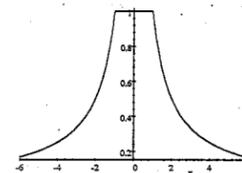
Nombre: Huber $\begin{cases} \text{if } |x| \leq k \\ \text{if } |x| > k \end{cases}$



$$\rho(x) = \begin{cases} \frac{x^2}{2} \\ k(|x| - \frac{k}{2}) \end{cases}$$

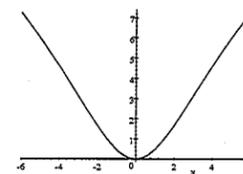


$$\Psi(x) = \begin{cases} x \\ k \text{sgn}(x) \end{cases}$$

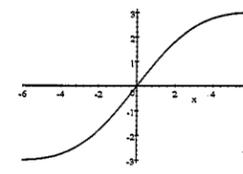


$$w(x) = \begin{cases} 1 \\ \frac{k}{|x|} \end{cases}$$

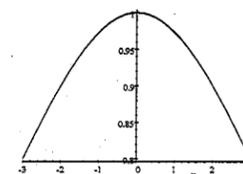
Nombre: Cauchy



$$\rho(x) = \frac{c^2}{2} \log \left(1 + \left(\frac{x}{c} \right)^2 \right)$$

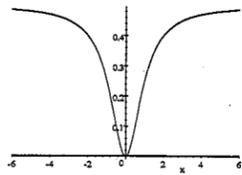


$$\Psi(x) = \frac{x}{1+(\frac{x}{c})^2}$$

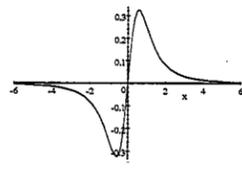


$$w(x) = \frac{1}{1+(\frac{x}{c})^2}$$

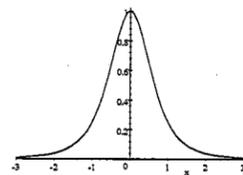
Nombre: Hampel



$$\rho(x) = \frac{x^2}{1+x^2}$$

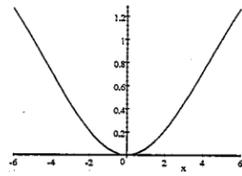


$$\Psi(x) = \frac{x}{(1+x^2)^2}$$

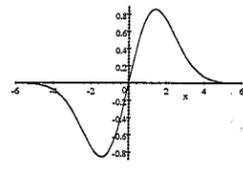


$$w(x) = \frac{1}{(1+x^2)^2}$$

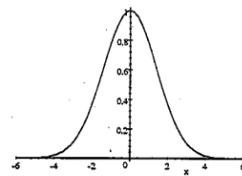
Nombre: *Welsch*



$$\rho(x) = \frac{c^2}{2} [1 - \exp(-(\frac{x}{c})^2)]$$

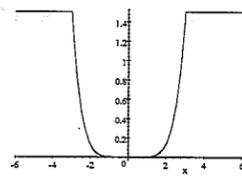


$$\Psi(x) = x \exp(-(\frac{x}{c})^2)$$

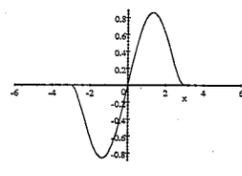


$$w(x) = \exp(-(\frac{x}{c})^2)$$

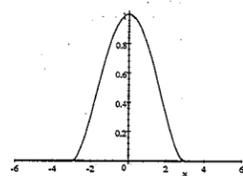
Nombre: *Tukey* $\begin{cases} \text{if } |x| \leq c \\ \text{if } |x| > c \end{cases}$



$$\rho(x) = \begin{cases} \frac{c^2}{6} (1 - [1 - (\frac{x}{c})^2]^3) \\ \frac{c^2}{6} \end{cases}$$



$$\Psi(x) = \begin{cases} x [1 - (\frac{x}{c})^2]^2 \\ 0 \end{cases}$$



$$w(x) = \begin{cases} [1 - (\frac{x}{c})^2]^2 \\ 0 \end{cases}$$

Existen varias programas de software para el caso de la regresión lineal robusta, ejemplo de ellos son LINWDR, ROBETH, TROLL y PROGRESS, por mencionar algunos. Estos cuatro programas fueron desarrollados entre los años 70-80.

Apéndice B

Implementación de los Algoritmos

En este apéndice se muestra el código fuente de las partes del programa que se consideran vitales para el funcionamiento total del sistema de software desarrollado. Los algoritmos descritos son:

1. Determinación del número de marcas existentes en la imagen y su posición.
2. Detección de los puntos de máxima magnitud por renglón para cada marca válida.
3. Detección de los cruces por cero de la fase, con base en los puntos de máxima magnitud.
4. Regresión lineal utilizada para detectar los cruces por cero de la fase.
5. Regresión lineal robusta sobre los puntos de cruce por cero, por cada marca se obtiene la ecuación $y = mx + b$.
6. Calibración no robusta para determinar el factor β , requerido para mapear los datos locales a coordenadas globales.
7. Calibración robusta para determinar el factor β .
8. Clusterización de los datos obtenidos. Con ello se obtiene la medición nominal de la posición en micras de cada línea graduada procesada.

El sistema de software desarrollado utiliza internamente varias estructuras de datos globales, las tres estructuras de datos más importantes : *FImagen*, *Configuración* y *Filtro*. La descripción de las componentes de cada estructura se muestra en detalle en los siguientes párrafos.

Estructura Imagen

La primera acción que requiere el usuario es cargar la imagen a procesar, dicha imagen que está en formato bmp de 256 niveles de gris sin comprimir es almacenada en la estructura:

```
struct FImagen
{
    double **data; // doble apuntador, matriz de dobles
    int nr,nc; // almacena el número de renglones y columnas
} F,Copia; // variables globales
```

De la estructura son declaradas dos variables globales *F* y *Copia*, la primera variable es la interfaz con el resto del programa y la segunda variable sirve como respaldo para recuperar la imagen original, esto se utiliza internamente para fines de visualización.

Estructura Configuración

Antes de procesar el vídeo capturado *V* se requiere definir los datos de configuración del sistema, estos se logra cargando una imagen y con base en ella se ajustan los datos de configuración. Por su grado de complejidad se dividen en dos grupos; el primer grupo abarca la configuración básica que es:

- Definir el ancho de la marca, esto se logra ajustando las variables *pos_ini_marca* y *pos_fin_marca*. Una vez ajustadas se actualiza el ancho de la marca $ancho_marca = pos_fin_marca - pos_ini_marca + 1$

- Establecer la zona rectangular de trabajo dentro de la imagen mediante las cotas *CotaArriba*, *CotaAbajo* y *CotaDer*, *CotaIzq*.

Estas son las variables de configuración iniciales para que el sistema de software trabaje correctamente, el segundo grupos de variables de configuración son para una configuración avanzada, no es frecuente que se ajusten estos parámetros. Las variables que lo conforman son:

- Definir el *porcentaje* que se requiere para seleccionar las marcas que son consideradas válidas, en el primer algoritmo se explica con detalle su funcionamiento.
- El cruce por cero de la fase se realiza en una vecindad *No_Puntos* tomando como centro el valor de máxima magnitud.
- La detección del valor de máxima magnitud se hace en una vecindad *porcentaje_dim*.
- El criterio de paro en el algoritmo de regresión lineal robusta es que la cantidad de datos que influyen en la regresión lineal debe ser mayor que *porcentaje_datos*.

La estructura completa de configuración de datos es:

```
struct Configuracion
{
    int pos_ini_marca; // posición inicial de la marca de la regla
    int pos_fin_marca; // posición final de la marca de la regla
    int pos_fin_marca // posición final de la marca de la regla
    int ancho_marca; // ancho de la marca de la regla
    int CotaDer; // cota derecha donde es válida la imagen
    int CotaIzq; // cota izquierda donde es válida la imagen
    int CotaArriba; // cota arriba donde es válida la imagen
    int CotaAbajo; // cota abajo donde es válida la imagen
```

```

int No_Puntos; // vecindad en donde se encuentra el cruce por cero de la fase
int step; // decremento de la zona válida en el algoritmo de regresión robusta
float porcentaje; // de la integral los valores > porcentaje son válidos
float porcentaje_dim; /// porcentaje del ancho de la marca de la imagen
float porcentaje_datos; // condición de paro en la regresión lineal robusta
}Config;

```

Estructura Filtro

Una vez definidas las variables de configuración se puede procesar el video V y obtener la ecuación de la recta para cada línea graduada presente en la imagen. Para ello se requiere la estructura de datos *Filtro* que contiene trece variables que se dividen en dos grupos; el primer grupo de variables se encarga de procesar la imagen usando el filtro de Gabor obteniendo la matriz de magnitud M y fase ϕ , sus componentes son:

- Se define el kernel complejo en función del ancho de la marca referenciador por *Config.dim* y se almacena en los arreglos *Cos*, *Sin*.
- El resultado de la convolución de la imagen con el filtro complejo se almacena en las matrices *outReal* y *outImg*.
- La magnitud y fase son almacenadas en las matrices *Mag* y *Fasee*.

Una vez obtenida la matriz de magnitud Mag y fase $Fasee$ el proceso siguiente es detectar los cruces por cero de la fase y en base de ellos obtener la ecuación de la recta. Para ello se utilizan las siguientes siete variables del segundo grupo:

- La sumatoria por columnas de la matriz de magnitud *Config.Mag* se almacena en el arreglo *integral*.
- El número de marcas detectadas en una imagen se guarda en *ipvecMaxMag*.
- Para detectar el número de marcas se utiliza el arreglo *aux2*.

- Los puntos de máxima magnitud son almacenados en la matriz *MaxMag*, los elementos de una columna pertenecen a la misma marca.
- Los cruces por cero de fase se guardan en la matriz *Algoritmo4*.
- Los parámetros de la ecuación de una línea recta son guardados en los arreglos M , B .
- Una copia de las dimensiones de la imagen guardadas en nr,nc .

Durante la exposición de los algoritmos se hace uso de estas variables haciendo más comprensible el uso de cada una de ella. La estructura de datos completa es:

```

struct Filtro
{
double *Cos,*Sin; // kerneles del filtro de Gabor
double **outReal; // salida real del filtro de Gabor
double **outImg; // salida img del filtro de Gabor
double **Mag; // almacena la magnitud
double **Fasee; // almacena la fase
double *integral; // almacena la integral de la magnitud
double *aux2; // arr binario, 1=zona de marca y 0=zona sin marca de la regla
int ipvecMaxMag; // cuenta el número de marcas válidas
double **MaxMag; // almacena los puntos de máxima magnitud
double **Algoritmo4; // almacena los cruces por cero
double M[MAX],B[MAX]; // almacena los parámetros m's, b's detectados
int nr,nc // copia de nr y nc de la estrucutra FImage
}Gabor;

```

La exposición de los algoritmos comienza asumiendo que la imagen de entrada ya fue procesada por el filtro de Gabor obteniéndose como resultado la matriz de magnitud y fase almacenadas en *Config.Mag* y *Config.Fasee* respectivamente.

B.1 Número de Líneas y Ubicación

Para obtener el número de marcas y sus respectivas zonas de ubicación se realiza lo siguiente: se integra por renglón la matriz M obteniéndose un arreglo $Gabor.Integral$, del arreglo se obtiene el valor máximo max y dado un porcentaje $Config.porcentaje$ se seleccionan los puntos que satisfacen $Gabor.Integral[i] > max - max * Config.porcentaje / 100$, las posiciones de los puntos que cumplen la condición se almacena en $Gabor.aux2$. Este arreglo contiene 0/1 donde un 1 indica que se cumplió la condición y un 0 que no se cumplió, con ello se sabe cuantas marcas hay en la imagen y la vecindad en donde se encuentran. El arreglo $Gabor.aux2$ se utiliza como guía para determinar si el máximo en la magnitud de una marca en un renglón es considerada como válida, para hecho se realiza lo siguiente: recorro el arreglo $Gabor.aux2$ mientras existan 0 en la transición 0 – 1 guardo la posición ip_ini sigo recorriendo mientras exista 1 y en la transición 1 – 0 guardo la posición ip_fin , con estas dos marcas busco dentro de esa vecindad el índice del valor máximo: $ip_max = \{M(j) | j \in V_{ip_fin, ip_ini}\}$. Para considerar que la marca detectada es válida debe de satisfacer la condición de concavidad hacia abajo alrededor del punto de máxima magnitud, la condición de concavidad hacia abajo es necesaria debido a que dadas las cotas izquierda y derechas de la imagen es posible que la marca si aparezca completa en la imagen pero alguna cota tape dicha marca, si cumple la condición aumento el contador de números de marcas encontradas $Gabor.ipvecMaxMag++$ y guardo la posición del valor de máxima magnitud en una matriz $Gabor.vec_max[Gabor.ipvecMaxMag=ip_max]$. Este proceso se repite mientras existan elementos por procesar. Con esto se determinan el número de marcas en la imagen y las posiciones de máxima magnitud generales. El código fuente necesario es:

```
// In: Matriz de magnitud y fase
// Out: Número líneas graduadas y los puntos de maxima magnitud global del lugar
//      donde estan las marcas en la imagen.
void Get_Num_Lineas_Posicion(void)
```

```
{
int    i,j,k;           // índices para los arreglos
float  s;              // variable auxiliar
int    ip_ini,ip_fin,dim; // cotas de inicio y fin de una marca
int    ip_max;        // guarda la posicion de maxima magnitud
// limpio las variables
for(i=0;i<Gabor.nc;i++)
    Gabor.integral[i]=Gabor.aux[i]=Gabor.aux2[i]=0;
// integro sobre y
for(j=Config.CotaIzquierda;j<Gabor.nc-Config.CotaDerecha;j++)
{
    // inicializo el valor de la suma
    s=0;
    // realizo la suma sobre los valores de la magnitud
    for(i=Config.CotaArriba; i<Gabor.nr-Config.CotaAbajo; i++)
        s+=Gabor.Mag[i][j];
    // guardo el valor de suma
    Gabor.integral[j]=s;
} // for j

// obtengo el valor maximo del vector Gabor.integral
// inicializo el valor maximo
max=0;
// recorro los elementos del arreglo Gabor.integral
for(j=Config.CotaIzq j<Gabor.nc-con;j++)
    if(Gabor.integral[j]>max) max=Gabor.integral[j];

// obtengo el vector de datos de la integral que estan a un porcentaje del valor maximo
for(j=Config.Cota_izq; j<Gabor.nc-Config.CotaDer; j++)
```

```

    if(Gabor.integral[j]>=max-max*epsi/100.0) Gabor.aux[j]=Gabor.integral[j];
// obtengo un vector [0,1] en Gabor.aux2[]
for(j=Config.CotaIzq; j<Gabor.nc-Config.CotaDer; j++)
    Gabor.aux2[j]=((Gabor.aux[j]!=0)?1:0);
//inicializo el contador de marcas de linea
Gabor.ipvecMaxMag=0;
// inicializo la posicion de inicio de la primera marca
ip_ini=0;
for(j=Config.CotaIzq; j<Gabor.nc-Config.CotaDer; j++)
{
    // si se cumple indica la transicion 0-1 indica el inicio de la marca
    if(Gabor.aux2[j]==1 && Gabor.aux2[j-1]==0)
        ip_ini=j;
// si se cumple indica la transion 1-0 marcando el fin de la marca
if( Gabor.aux2[j]==0 && Gabor.aux2[j-1]==1 )
{
    ip_fin=j;
// en dicha vecindad ip_ini..ip_fin obtengo el valor del arreglo Gabor.Integral
// inicializo el valor del maximo y su posicion
max=0, ip_max=0;
for(k=ip_ini-1;k<ip_fin+1;k++)
    if(Gabor.aux[k]>max)
    {
        // guardo el valor maximo y la posicion
        max=Gabor.aux[k];
        ip_max=k;
    }
// checo la condicion de concavidad hacia abajo

```

```

if(Gabor.aux[ip_max-1]<Gabor.aux[ip_max])
    if(Gabor.aux[ip_max+1]<Gabor.aux[ip_max])
        if(Gabor.aux[ip_max-1]!=0)
            if(Gabor.aux[ip_max+1]!=0)
                {
                    // agrego la línea y guardo la posicion del maximo
                    Gabor.vec_max[Gabor.ipvecMaxMag++]=ip_max;
                }
    } // fin del if de la transion 1-0
} // for j recorre los elementos de Gabor.integral
} // fin del procedimiento

```

B.2 Puntos de Máxima Magnitud

Una vez determinado el número de marcas presentes en la imagen y su ubicación general se requiere detectar los puntos de máxima magnitud por renglón para cada una de las marcas válidas. Para cada renglón de la imagen se realiza lo siguiente: se posiciona en el punto de maxima magnitud general dado por *Gabor.vec_max* y dado un porcentaje *Config.porcentaje_dim* que esta en función del ancho de la marca del rectangulo dado en la opción de configuración, se busca en dicha vecindad el valor de máxima magnitud: $ip_max = \max\{M(j) | j \in vecindad\}$, este valor se guarda en la matriz *Gabor.vecMaxMag*. Este proceso se repite para cada marca y para todos los renglones. El código fuente es:

```

// In: Porcentaje del rectangulo Config.porcentaje_dim y posiciones globales de las marcas
Gabor.vec_max.
// Out: Matriz Gabor.vecMaxMag con las posiciones de máxima magnitud.
void Get_Matrix_Max_Magnitud(void)
{

```

```

int i,j,k; // índices para los arreglos
float max; // almacena el valor de max magnitud
int ip_max; // índice del valor de max magnitud
float porcentaje; // guarda el porcentaje del ancho del rectangulo donde se busca
// el valor de máxima magnitud

int cota_ini,cota_fin;
// obtengo el tamaño de la vecindad que esta en función del porcentaje del ancho de la
marca
porcentaje=Config.dim*Config.porcentaje_dim/100.0;
// recorro cada linea encontrada en la integral
for( k=0; k<Gabor.ipvecMaxMag; k++)
{
// dado el centro Gabor.vec_max le agrego la vecindad porcentaje y checo las
// condiciones de frontera
cota_ini=(( Gabor.vec_max[k]-Config.dim*porcentaje)<0)?
0:Gabor.vec_max[k]-Config.dim*porcentaje);
cota_fin=(( Gabor.vec_max[k]+Config.dim*porcentaje)>Gabor.nc)?
Gabor.nc: Gabor.vec_max[k]+Config.dim*porcentaje);
// inicializo variables
max=0, ip_max=0;
// encuentro el maximo por renglon en cada componente de vec_max
for(i=Config.CotaArriba; i<Gabor.nr-Config.CotaAbajo ;i++)
{
// inicializo el valor maximo
max=0;
// obtengo el valor maximo de la magnitud en cierta vecindad
for( j=cota_ini; j<cota_fin; j++)
if(Gabor.Mag[i][j]>max)

```

```

{
// guardo el valor maximo y su posición
max=Gabor.Mag[i][j];
ip_max=j;
}
// agrego el indice del valor máximo
Gabor.vecMaxMag[i][k]=ip_max;
} // for i
} // for k que recorre las marcas por renglón
} // fin del procedimiento

```

B.3 Cruces por Cero

En base a las posiciones de máxima magnitud almacenadas en la matrix *Gabor.vecMaxMag* se determinan los cruces por cero de la fase, obteniendo la precisión subpixel del centro de la marca, esto se realiza para cada renglón. Para ello se requiere una vecindad de puntos *Config.No_Puntos* alrededor del puntos de máxima magnitud , en esta vecindad se seleccionan los puntos de la fase y por medio de regresión lineal se obtienen los parámetros m, b de la recta $y = mx + b$, de donde igualando a cero se obtiene el cruce por cero de la fase: $x = -b/m$. Para ello se requieren dos algoritmos, el primero recorre las marcas de la regla válidas e invoca al segundo algoritmo el cual detecta por renglón el cruce por cero de la marca correspondiente.

```

// In : Puntos de máx mag Gabor.vecMaxMag y número de marcas Gabor.ipvecMaxMag.
// Out : Matriz con los cruces por cero de la fase Gabor.Algoritmo4.
void Cruces_x_Cero()
{
int i; // índice del arreglo
// para cada marca de la regla obtener sus cruces por cero por renglón

```

```

for( i=0; i<Gabor.ipvecMaxMag; i++ )
    Cruces_x_Cero_NPuntos(i,Config.No_Puntos);
}

// In : Indice de la marca col_k y vecindad de puntos N_Puntos
// Out : cruce por cero de la fase para cada renglón en la marca col_k
void Cruces_x_Cero_NPuntos(int col_k, int N_Puntos)
{
int    i,j,k;                // índices de los arreglos
double m,b;                 // var almacena los parametros de la regresion lineal
int    x_max;               // guarda el valor de máxima magnitud
double x_dat[100],y_dat[100]; // arr auxiliares
// recorro los renglones de la matriz de fase
for( i=Config.CotaArriba; i<Gabor.nr-Config.CotaAbajo; i++ )
{
// obtengo el valor de máxima magnitud de la marca col_k
x_max=Gabor.vecMaxMag[i][col_k];
// obtengo los valores de la fase en una vecindad del punto de maxima mag
for(j=0;j<2*N_Puntos+1;j++)
{
// guardo el índice
x_dat[j]=x_max-N_Puntos+j;
// guardo el valor de la fase
y_dat[j]=Gabor.Fasee[i][x_max-N_Puntos+j];
}
// aplico regresión lineal y obtengo los parámetros m,b
Minimos_Cuadrados(x_dat,y_dat,2*N_Puntos,&m,&b);
// guardo el valor del cruce por cero de la fase
Gabor.Algoritmo4[i][i-Config.CotaArriba]=-b/m;
}
}

```

```

} // fin del índice de los renglones
} // fin del procedimiento.

```

B.4 Regresión Lineal

El algoritmo de cruces por cero hace uso de la regresión lineal, ello es necesario para encontrar la estimación de los parámetros de la recta $y = mx + b$ de donde se obtiene el cruce por cero de la fase con precisión sub-pixel. Las fórmulas cerradas para la estimación de los parámetros se muestran en la sección 3.3. El código fuente requerido es:

```

// In : arreglos x[],y[]. Dimensión del arreglo N y las variables m,b de inicialización.
// Out: parámetros estimados m,b
void Minimos_Cuadrados(double *x,double *y,int N,double *m, double *b)
{
int    i;                    // índice de los arreglos x,y
double alfa,beta,gama,li;    // variables auxiliares
// inicializo las variables con cero
alfa=beta=gama=li=0;
// recorro los elementos del arreglo y calculo los parametros m,b
for( i=0; i<N; i++)
{
alfa = alfa+x[i]*y[i];
beta = beta+x[i];
gama = gama+y[i];
li    = li+x[i]*x[i];
} // fin del for
*m = ((N)*alfa - beta*gama)/((N)*li - beta*beta);
*b = ( li*gama - alfa*beta)/((N)*li - beta*beta);
}

```

B.5 Regresión Lineal Robusta

Sobre el conjunto de puntos de cruces por cero de la fase que están almacenados en la matriz *Gabor.Algoritmo4* se aplica la regresión lineal robusta obteniéndose los parámetros m, b donde m indica la pendiente de la recta y b la posición local del centro de cada línea graduada capturada en la imagen. Este procedimiento se divide en tres algoritmos: el primero selecciona las marcas sobre las cuales se hará el ajuste, el segundo algoritmo toma el índice de la marca y obtiene dos arreglos con el índice y valor de los cruces por cero para los renglones de la imagen, el último algoritmo realiza el ajuste de regresión robusta regresando el parámetro m, b que es almacenado en los vectores M y B . La implementación del tercer algoritmo corresponde al *algoritmo-3* descrito en la sección 3.2.2.

```
// In : Matriz con los cruces por cero de la fase y número de marcas
// Out : parámetros m.,b de cada marca
void Get_Rectas(void)
{
    int i; // índice del arreglo
    // para cada marca presente en la imagen
    for( i=0; i<Gabor.ipvecMaxMag; i++)
    {
        // realizo la regresión lineal robusta de la marca i y almaceno el resultado
        Get_Recta_Robusta(i,Gabor.M[i],Gabor.B[i]);
    } // fin del i
} // fin del procedimiento

// In : índice de la marca y las variables m,b
// Out : parámetros m,b del ajuste robusto a la marca referenciado por col_k
void Get_Recta_Robusta(int col_k, double *m_, double *b_)
```

```
{
    int i,j; // índice de los arreglos
    // inicializo el contador de número de renglones
    j=0;
    // almaceno los índices y valores de los cruces por cero de la marca referencia col_k
    for( i=Config.CotaArriba; i<Gabor.nr-Config.CotaAbajo; i++ )
    {
        Gabor.y_dat[j]=Gabor.Algoritmo4[i][col_k];
        Gabor.x_dat[j]=j;
        j++;
    }
    // como primer paso realizo el ajuste sobre todos los puntos, esto es la condicion
    inicial
    // para el algoritmo de regresión lineal robusta
    Minimos_Cuadrados(Gabor.x_dat,Gabor.y_dat,Gabor.nr,m_,b_);
    RegresionCasiRobusta(Gabor.x_dat,Gabor.y_dat,N,m_,b_);
} // fin del procedimiento

// In : Dos arreglos de datos datos_x, datos_y. La dimensión del arreglo N y las
// variables donde se regresará el resultado
// Out : Parámetros m,b
void RegresionCasiRobusta(double *datos_x, double *datos_y, int N, double *m_, double
*b_)
{
    int i,j; // índices de los arreglos
    float cantidad_datos // almacena la cantidad de datos que corresponde al
porcentaje // Config.dim
    // Config.dim
```

```

double alfa,beta,gama,li; // var para ajuste de regresión
double m,b; // auxiliares para el cálculo de regresión

// inicializo los valores
m=*m_, b=*b_, cantidad_datos=round((float)N*Config.porcentaje_datos/100.0);
// Itero mientras la cantidad de datos procesados sea >= cantidad_datos
do
{
// inicializo las variables
cont=0, alfa=beta=gama=li=0;
for( i=0; i<N; i++)
{
// si satisfacen la condición de pertenencia a la zona de umbral omega
// los utilizo para el cálculo de regresión
if ( fabs(datos_y[i]-*m_*datos_x[i]-*b_)<=cota)
{
alfa = alfa+datos_x[i]*datos_y[i];
beta = beta+datos_x[i];
gama = gama+datos_y[i];
li = li+datos_x[i]*datos_x[i];
// incremento el contador de datos
cont++;
} // fin del if
} // fin del índice i

// verifico que si hubo datos procesados y actualizo m,b
if(cont>0)
{
*m_=((cont)*alfa - beta*gama)/((cont)*li - beta*beta );

```

```

*b_=( li*gama - alfa*beta)/((cont)*li - beta*beta );
m=*m_;
b=*b_;
}
// decremento la vecindad
cota=cota-Config.step;
// verifico la condición de paro
}while(cont>cantidad_datos);
} // fin del procedimiento

```

B.6 Calibración

El proceso de calibración consiste en encontrar el factor β que indica cuantas micras equivale un pixel de la imagen, en la sección 4.1.1 se explica la ecuación que lo describe. En base a este factor de escala se obtiene el vector $Z^w = z_i + \beta x_j$ el cual contiene la información global de las líneas graduadas procesadas y de donde se determinarán las componentes de una misma línea. El código fuente para obtener el factor de calibración es:

```

// In : Dos arreglos, Z con las coordenadas del sensor, B con las posiciones
// del centro de las marcas y la dimensión del arreglo N
// Out : regresa el factor de calibración Beta
double Calibración (double Z[], double B[], int N)
{
int i,j; // índice de los arreglos
double num,den; // aux para el cálculo de la calibración
// inicializo las variables
num=0, den=0;
// para cada elemento del arreglo

```

```

for( i=0; i<N; i++ )
{
    // recorriendo las parejas de datos
    for( j=i+1; j<N; j++ )
    {
        num += fabs(Z[i]-Z[j])*fabs(B[i]-B[j]);
        den += fabs(B[i]-B[j])*fabs(B[i]-B[j]);
    }
}
return -num/den;
} // fin del procedimiento

```

B.7 Calibración Robusta

El proceso de calibración consiste en determinar de forma robusta el factor de calibración β que indica cuantas micras equivale un pixel de la imagen, en la sección 4.1.2 se explica la ecuación que lo describe. En base a este factor de escala se obtiene el vector $Z^w = z_i + \beta x_j$ el cual contiene la información global de las líneas graduadas procesadas y de donde se determinarán las componentes de una misma línea. El código fuente para obtener el factor de calibración β robusto es:

```

double Calibracin_Robusta(double Beta1, double epsi)
{
    int i,j,k; // var aux para contadores
    double val1,val2,W; // temporales
    double num,den;
    double Beta2;
    //Itero hasta convergencia
    do{

```

```

// actualizo el parametro beta
Beta2=Beta1;
num=den=0;
for(i=0; i<Calibra.Nz; i++)
    for(j=0;j<Calibra.Nz;j++)
    {
        k=(i+j+1)%Calibra.Nz;
        if (fabs(Calibra.B[i]-Calibra.B[k])!=0)
        {
            val1=fabs(Calibra.B[i]-Calibra.B[k]);
            val2=(fabs(Calibra.Z[i]-Calibra.Z[k])-Beta1*fabs(Calibra.B[i]-Calibra.B[k]));
            val2=val2*val2;
            val2=(1.0+val2)*(1.0+val2);
            W=val1/val2;
            num+=W*fabs(Calibra.Z[i]-Calibra.Z[k]);
            den+=W*fabs(Calibra.B[i]-Calibra.B[k]);
        } // end if
    } // end de los for
Beta1=num/den;
}while(fabs(Beta2-Beta1)>epsi);
return (Beta1);
}

```

B.8 Cluster de Datos Finales

Como se describe en el capítulo 4, despues de aplicar el factor de escala β se obtiene el vector de datos Z^w que contiene la información global de las posiciones de las líneas graduadas procesadas. El ajuste de datos corresponde a encontrar los cluster

$C = \{c_0, c_1, \dots, c_n\}$ del vector Z^w , donde cada cluster contiene información sobre la posición de una misma línea graduada. Para cada cluster se requieren sus dos primeros momentos estadísticos, media y varianza. El siguiente algoritmo es la implementación del algoritmo de clasificación de puntos finales descrito en la sección 4.2.2.

```
// In : Elementos del vector Z y la dimensión del arreglo N
// Out : Número de clusters N y arreglos con la media y varianza de cada cluster
void Clasificación_Final(double media, double delta, double Lista[], int N,
                        int N_cluster, double *Media, double *Varianza)
{
    int i;                // índice del arreglo
    double elemento;     // aux para el primer elemento del cluster
    double Datos;        // aux para ir guardando los datos de un cluster
    int cont_clases;     // contador del número de cluster

    // inicializo las variables
    i=0; cont_clases=0; cont_datos=0;
    // inicializo el primer elemento del primer cluster
    elemento=Lista[0];
    // mientras existan elementos por procesar
    while(i<N)
    {
        if( fabs(elemento-Lista[i])<media )
        {
            // guardo el indice, es decir lo agrego al cluster-k
            Datos[cont_datos]=i;
            // incremento el número de elementos del cluster
            cont_datos++;
            // incremento el índice del arreglo
            i++;
        }
    }
}
```

```
    } // fin del if
else
{
    // si no se cumple indica que quedo definido un cluster
    // se calcula su media y varianza y lo dejo en el arreglo Media y Varianza
    Calcula_Media_Varianza(Datos,cont_datos,
                            &Media[cont_clases],&Varianza[cont_clases]);
    // incremento el numero de clases detectadas
    cont_clases++;
    // reinicio el contador de datos
    cont_datos=0;
} // end del else
} // end while
} // fin del procedimiento
```

Los algoritmos descritos en este apéndice son los más importantes para el funcionamiento del sistema de software, el orden de exposición corresponde a la forma que son utilizados dentro del sistema de software.

Apéndice C

Certificado de Calibración CENAM

El informe de calibración facilitado por el CENAM, corresponde a una regla de cristal de 200mm de longitud con un equiespaciado de 1mm entre cada línea. Esta regla fue utilizada en la sección 5.2.2 para efectos del estudio del comportamiento del sistema propuesto. Los datos del informe que pudieran identificar el origen de la empresa que solicitó el servicio, han sido omitidos por efectos de confiabilidad.

Informe de la calibración

Cliente: Centro Nacional de Metrología, División Dimensional.
Equipo: Escala para Microscopio de 0 - 200 mm.
Serie:
Fecha de calibración: CONFIDENCIAL

DESARROLLO:

La calibración se realizó con un microscopio Leitz modelo Libra 200, con amplificación de 100x, número de serie 418631, trazable al patrón primario de longitud y de acuerdo al procedimiento interno 740-AC-P.142.

RESULTADOS OBTENIDOS DE LA CALIBRACION:

1.- Espesor de líneas grabadas:

Promedio 0,0540 mm
Máximo 0,0543 mm
Mínimo 0,0525 mm

2.- Evaluación de la escala graduada:

TABLA I Desviaciones medidas desde 0 hasta 200 mm.

Valor nominal mm	Desviación μm	Valor nominal mm	Desviación μm	Valor nominal mm	Desviación μm
1	-0,1	21	-2,3	41	-3,6
2	-0,1	22	-2,5	42	-3,5
3	-0,2	23	-2,3	43	-3,8
4	0,1	24	-2,4	44	-3,4
5	-0,1	25	-2,6	45	-3,5
6	0,0	26	-2,5	46	-3,9
7	-0,2	27	-3,1	47	-3,8
8	-0,1	28	-2,6	48	-3,9
9	0,0	29	-2,7	49	-3,8
10	-0,2	30	-3,5	50	-3,6
11	0,0	31	-3,0	51	-3,8
12	-1,5	32	-3,5	52	-3,9
13	-1,6	33	-3,6	53	-3,6
14	-1,9	34	-3,7	54	-3,8
15	-2,0	35	-3,9	55	-3,8
16	-1,8	36	-3,9	56	-3,8
17	-1,7	37	-3,7	57	-3,6
18	-1,6	38	-3,9	58	-3,6
19	-1,2	39	-3,4	59	-3,7
20	-1,7	40	-3,5	60	-3,7

Figura 6-1:

TABLA I (Continuación). Desviaciones medidas desde 0 hasta 200 mm.

Valor nominal mm	Desviación μm	Valor nominal mm	Desviación μm	Valor nominal mm	Desviación μm
61	-3,4	108	-3,9	155	-4,4
62	-3,4	109	-3,9	156	-3,9
63	-3,3	110	-4,0	157	-4,4
64	-3,4	111	-3,5	158	-4,2
65	-3,6	112	-3,5	159	-4,3
66	-3,1	113	-3,9	160	-3,9
67	-3,0	114	-3,5	161	-4,2
68	-3,9	115	-4,0	162	-4,2
69	-3,4	116	-4,0	163	-4,5
70	-3,9	117	-4,2	164	-4,2
71	-4,0	118	-4,5	165	-4,1
72	-3,9	119	-4,4	166	-4,0
73	-3,6	120	-4,6	167	-4,1
74	-4,1	121	-4,8	168	-4,1
75	-2,9	122	-4,9	169	-3,8
76	-3,2	123	-4,8	170	-4,0
77	-4,7	124	-4,7	171	-3,9
78	-4,7	125	-4,5	172	-3,8
79	-4,7	126	-4,5	173	-4,5
80	-4,0	127	-4,5	174	-4,4
81	-4,1	128	-4,4	175	-4,4
82	-4,4	129	-4,5	176	-4,2
83	-3,7	130	-4,4	177	-3,8
84	-4,0	131	-4,1	178	-4,4
85	-3,8	132	-4,0	179	-3,6
86	-4,2	133	-4,0	180	-4,1
87	-4,1	134	-4,4	181	-4,1
88	-4,0	135	-4,1	182	-4,2
89	-3,9	136	-4,2	183	-3,3
90	-4,0	137	-4,4	184	-3,3
91	-3,4	138	-4,4	185	-3,7
92	-3,9	139	-4,4	186	-3,7
93	-3,7	140	-3,9	187	-3,9
94	-4,3	141	-4,4	188	-4,1
95	-4,1	142	-4,1	189	-4,2
96	-4,3	143	-4,3	190	-4,2
97	-4,2	144	-3,4	191	-4,3
98	-4,5	145	-3,8	192	-4,2
99	-4,0	146	-4,1	193	-4,5
100	-4,2	147	-4,2	194	-4,3
101	-4,5	148	-4,7	195	-4,2
102	-4,2	149	-4,8	196	-4,5
103	-4,0	150	-4,4	197	-4,0
104	-3,6	151	-4,1	198	-4,1
105	-3,7	152	-4,1	199	-4,3
106	-4,1	153	-4,1	200	-3,8
107	-3,3	154	-4,2		

Figura 6-2:

Bibliografía

- [1] A. Lassila, E. Ikonen, and K. Riski, "Interferometer for calibration of graduated line scales with a moving CCD camera as a line detector", *Applied Optics*, 1994.
- [2] H. Heister, "Zur automatischen Kalibrierung geodätischer Längeneinstrumente", *Studiengang Vermessungswesen Universität der Bundeswehr München, Schriftenreihe*, 1988.
- [3] J. S. Beers and K. B. Lee, "Interferometric measurement of length scales at the National Bureau of Standards", *Precis. Eng.* 4, 205-214 (1982).
- [4] D. Ullrich, "Vergleich der interferometrischen Längenmessung von Strichmasstäben mit Kryptonlampe und stabilisierte Zweifrequenzlase", *PTB-itt.* 91, 165-169 (1981).
- [5] K. H. Hart, "A laser-source photo-electric microscope for sensing moving scale lines", *Metrologia* 13, 63-66, 1981.
- [6] H. Matsumoto, S. Seino, and Y. Sakurai, "Automatic recording laser interferometer for line standard up to 2 m", *Metrologia* 16, 169-175 (1980).
- [7] K. J. Gasvik, "Optical Metrology", John Wiley and Sons, 1987.
- [8] Tony Flint, "Dimensional Metrology Course", National Research Council Canada, 1997.
- [9] Lewis V. Judson and Benjamin L. Page, "Calibrations of the line standards of length of the National Bureau of Standards", Volume 13, 1934.

- [10] Eugene Hecht, Alfred Zajac, "Optics", Addison-Wesley Iberoamericana, 1986.
- [11] Bernd Jahne, "Digital Image Processing - Concepts, algorithms and Scientific Applications", pp. 223, 4th Edition, Springer-Verlag, 1997.
- [12] C. González, Richard E. Woods, "Tratamiento digital de imágenes", Addison-Wesley, 1996.
- [13] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw and W.A. Stahel, "Robust Statistics: The Approach Based in Influence Functions", John Wiley and Sons, N.Y., 1986.
- [14] Jana Jureckova, Pranab Kumar Sen, "Robust Statistical Procedures, asymptotics and interrelations", Wiley-Intersciences, 1996.
- [15] Peter J. Huber, "Robust Statistics", Wiley Series in Probability and Mathematical Statistics, 1981.
- [16] Michael J. Black, Anand Rangarajan, "On the Unification of Line Processes, Outlier Rejection, and Robust Statistics with Applications in Early Vision", *International Journal of Computer Vision*, 19(1), 57-91, 1996.
- [17] Peter Meer, "Robust Techniques for Computer Vision", Tutorial Rutgers University, www.caip.rutgers.edu/~meer, 1999.
- [18] Gosta H. Granlund and Has Knutsson, "Signal Processing for Computer Vision", Kluwer Academic Publishers, 1995.
- [19] Edgar Román Arce Santana, "Métodos de Calibración de un Digitalizador Optico", Tesis de maestría en ciencias de la computación, CIMAT, 2000.