



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE INGENIERÍA

**IMÁGENES Y DIMENSIONES DE GOTAS EN FLUJOS DE
UN MOLINO DE DOS RODILLOS**

TESIS

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN TELECOMUNICACIONES**

PRESENTAN:

**JAIME RAMÍREZ RODRIGUEZ
ROMAN CISNEROS CORONADO**



**DIRECTOR DE TESIS:
DR. ENRIQUE GEFFROY AGUILAR**

CIUDAD UNIVERSITARIA, MÉXICO D.F., 2009



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A mis profesores en la Universidad

Al Dr. Víctor García Garduño por permitirme ingresar a la carrera de Telecomunicaciones.

Al Dr. Enrique Geffroy Aguilar por la dirección, supervisión y corrección de esta tesis. Y cuyos consejos y observaciones me impulsan a seguir avanzando.

Al Dr. Marco Antonio Reyes Huesca por su asesoría y compañerismo.

Agradecemos la beca para tesis de licenciatura del proyecto NC-204 del CONACyT.

A mis amigos.

En especial a Román y Miguel con quienes hice equipo en las actividades universitarias.

A mis familiares.

En especial a mis tíos Jaime y Betty cuyo apoyo hizo más factible lograr este objetivo.

Jaime Ramírez

*A mi madre, Margarita, por su esfuerzo incansable, y sobre todo, por su amor.
A mis abuelos Cirenio y Cointa, por creer siempre en mí.
A mis hermanos Ciro y Joab, por acompañarme en este viaje.
A mis amigos, porque sé que siempre puedo contar con ellos.*

Al Dr. Enrique Geffroy, por todas las facilidades otorgadas y por su invaluable guía.

Román Cisneros

Índice general

1. Introducción	3
1.1. El molino de dos rodillos	6
1.2. Geometría del dispositivo experimental	9
1.3. Geometría del flujo elongacional con vorticidad	12
I Investigación	15
2. La captura de imágenes de la dinámica de una gota	17
2.1. MTF	18
2.2. La transformada de Fourier	19
2.3. Una lente como un sistema lineal	20
2.3.1. Un ejemplo sencillo	27
2.4. Parámetros en un sistema de generación de imágenes	28
2.5. Resolución, contraste y gráficas de MTF	31
2.5.1. Encontrando la resolución en una cámara Nikon	34
2.6. Utilizando la MTF en la selección de una lente	50
3. La imagen digital y el filtrado en el dominio de la frecuencia	51
3.1. Modelo de imagen	51
3.2. Muestreo y cuantificación	52
3.3. La transformada discreta de Fourier de una dimensión y su inversa	53
3.4. La DFT de 2 dimensiones y su inversa	56
3.5. Filtrado en el dominio de la frecuencia	57
3.5.1. Algunas propiedades básicas del dominio de la frecuencia	57
3.5.2. Bases del filtrado en el dominio de la frecuencia	58
4. Detección de bordes	60
4.1. Operadores gradiente	64
4.2. Laplaciano	66
4.3. Máscara de diferencia absoluta	67
4.3.1. Máscara semigaussiana	67

4.3.2. Cálculo de pendiente y dirección del borde	68
II Experimentación	71
5. Implementación de algoritmos	73
5.1. Adquisición de video	74
5.2. Obtención de imágenes	75
5.3. Filtrado digital (emulación de una lente)	77
5.4. Detección de bordes	78
5.5. Identificación de gotas	85
6. Conclusiones	92
A. Código fuente	94
B. Breve introducción a la física del sistema	110
B.1. Representación matemática de las propiedades físicas de un fluido	111
B.2. Campo de velocidades y condición de compatibilidad	112
B.3. Ecuación de Navier-Stokes	115
B.4. Tipos de flujo	116
B.4.1. Flujos viscométricos	116
B.4.2. Flujos elongacionales	117
B.4.3. Flujos fuertes	118
B.5. Importancia de los flujos fuertes	119
B.6. Una gota en un flujo generado por un molino de dos rodillos	120
B.6.1. Solución analítica	121
B.6.2. Solución numérica	126
B.6.3. La gota en el flujo	128
Bibliografía	132

Capítulo 1

Introducción

Aquí presentamos una breve explicación del experimento para estudios de la deformación de gotas inducidas por flujos fuertes. Este experimento busca ampliar los conocimientos de hidrodinámica de objetos deformables bajo condiciones que han sido poco estudiadas hasta ahora, en particular, bajo condiciones en las que compiten tanto los efectos de la vorticidad como la rapidez de deformación de un flujo.

Frecuentemente en la naturaleza como en procesos científicos y tecnológicos se presentan flujos de sustancias que están constituidas por partículas y una fase fluida, como las suspensiones, emulsiones, sangre, tintas, etc. A este tipo de sustancias se les conoce como fluidos complejos pues presentan una estructura interna caracterizada por su tamaño mesoscópico (100 a 10,000 nanómetros). Debido a la existencia de esta enorme gama de sustancias es entonces necesario estudiar y entender su comportamiento físico para el desarrollo científico y tecnológico.

Los fluidos complejos deben la mayoría de sus propiedades y comportamiento al

carácter dinámico de su estructura. Esto es, los flujos complejos tienen propiedades típicas de sólidos como elasticidad y muestran también la viscosidad característica de los gases. Además, dado que un fluido complejo cambia su mesoestructura cuando se somete a condiciones de flujo externas, entonces la caracterización de estos fluidos requiere de su estudio en forma esencialmente instantánea. Puesto que generalmente los flujos impuestos sobre el fluido tienen dimensiones mucho mayores que las dimensiones de la estructura del fluido, entonces se tienen unas condiciones que se pueden llamar globales, debidas únicamente a las causas que producen el flujo, y unas condiciones que se pueden llamar locales, debidas a la presencia de la estructura inducida en la microestructura.

Dentro de los fluidos complejos se encuentran las suspensiones y emulsiones, las cuales contienen pequeñas gotas que le confieren una estructura dinámica al medio. En este caso el fluido externo o continuo es el que impone las condiciones de flujo mientras que las gotas o partículas embebidas –el fluido interno– con su dinámica propia establece el comportamiento de la estructura interna de pequeñas dimensiones.

Dada la complejidad de estudiar sistemas de muchos cuerpos, cada uno de ellos con dimensiones diferentes es entonces conveniente analizar el problema más simple que consiste en considerar el ambiente alrededor de una sola gota. Así pues y como punto de partida para estudiar estos fluidos, se estudia una gota aislada en un flujo fuerte bien caracterizado. Este esquema se puede construir cuidadosamente y del cual se puede obtener información muy detallada de la dinámica de la gota y de la interacción de los dos fluidos individualmente para posteriormente proceder a entender situaciones más complicadas.

El problema que en esta tesis se plantea corresponde a la problemática experimen-

tal para visualizar una gota en un medio exterior. El problema específico que se atiende está relacionado a las técnicas de procesamiento de imágenes que deben aplicarse para que un experimento eminentemente de mecánica de fluidos pueda alcanzar los objetivos de conocer la física que domina la dinámica propia de una gota y los esfuerzos externos aplicados por el flujo. En realidad, este trabajo no tiene una relación directa con la física de la dinámica de la gota que se estudia con este experimento, de manera que no es necesario profundizar en las enormes complicaciones matemáticas que representan al sistema en general [Reyes, 1997, Reyes, 2005]. Este trabajo está directamente relacionado con el tratamiento de las señales que se generan de este fenómeno físico: en este caso, la adquisición de secuencias de fotografías digitalizadas, que deben procesarse en forma rápida y precisa. Así pues, realizamos un procesamiento digital de imágenes de gotas inmersas en un flujo transparente. La idea es obtener información de las magnitudes geométricas de la gota y de su posición con respecto a un sistema de referencia en el flujo. Esto se obtiene por medio de fotografías y de este modo se minimiza la invasión en el fenómeno que deseamos estudiar, lográndose objetivos que de otra manera sería prácticamente imposible de lograr.

La metodología de trabajo experimental propuesta es un procedimiento que aprovecha los conocimientos que tenemos del comportamiento de la luz al cruzar algún medio material como puede ser una lente o un conjunto de lentes. Y entonces podemos capturar y almacenar información que es función de este comportamiento por medio de un CCD¹ y una estación de trabajo. A partir de entonces se comienza a obtener la información encerrada en las imágenes digitales.

Para proceder con el objetivo general anterior, es necesario comprender la capaci-

¹Del inglés *Charge-Coupled Device*

dad de adquisición de imágenes y su relación con el mundo real de un sistema óptico. De esta forma evitaremos malas interpretaciones a la información obtenida. Bien, comencemos pues con las características generales del experimento en cuestión.

1.1. El molino de dos rodillos

Un molino de dos rodillos es un dispositivo que tiene dos rodillos, los cuales giran en el mismo sentido, inmersos en un fluido denominado flujo soporte o exterior. La característica básica del dispositivo es la capacidad para generar un flujo esencialmente elongacional o *fuerte*. Las características hidrodinámicas del flujo que genera este dispositivo están determinadas exclusivamente por (a) la geometría del mismo (radio de los rodillos y la separación de éstos), así como (b) por las velocidades de giro de los rodillos. La idea de utilizar molinos de dos rodillos es porque tienen un *punto de estancamiento*, con las propiedades de flujo fuerte de interés, localizado en la región central del dispositivo, para el que además se tiene una solución analítica del flujo generado. Estas características conforman un conjunto único entre los flujos del tipo elongacional que hasta la fecha se han estudiado y resultan ser de gran relevancia tecnológica y técnica. Así, en el punto de estancamiento –y una pequeña región alrededor de este punto– las velocidades ahí presentes son casi nulas, permitiendo estudiar toda partícula contenida en esa región por tiempos muy largos, esencialmente infinitos. Entonces, un objeto embebido en esa región del fluido exterior, se deforma y gira bajo condiciones hidrodinámicas perfectamente conocidas.

Por tanto, el uso de molinos de dos rodillos permitirá abordar problemas aún poco estudiados –y aún no reportados en la literatura– sobre la dinámica de la deformación, ori-

entación, así como del proceso de ruptura y coalescencia de gotas, todo ello bajo condiciones de flujo interior y exterior donde los fenómenos hidrodinámicos dominantes pueden evaluarse y compararse con modelos analíticos y numéricos. Estas características son únicas para esta clase de dispositivos, y representa una valiosa oportunidad para el trabajo relevante a la industria de fluidos no miscibles como el aceite-agua de relevancia a la industria petrolera.

La manera de llevar a cabo la investigación anterior inició con estudios, de forma analítica y numérica, del flujo –unifase– generado por molinos de dos rodillos. La caracterización de todas las propiedades del campo de flujo, y con base en esos resultados, permitió diseñar un dispositivo cuyas características han sido previamente optimizadas.

En estos flujos se da énfasis a dos parámetros: (a) la razón de las magnitudes de la rapidez de deformación con respecto de la vorticidad y (b) la rapidez de deformación. A partir del cociente se obtiene un número adimensional denominado *parámetro del tipo de flujo*. Este parámetro es una medida local y objetiva de la cantidad de deformación que un flujo puede inducir sobre una celda de fluido. El parámetro de tipo de flujo está acotado entre -1 –que representa a un vórtice puro– y $+1$ –para un flujo puramente extensional–; el valor de 0 corresponde a la cinética típica en un flujo cortante simple. Los campos de flujos que van de -1 a 0 se llaman flujos débiles, pues son incapaces de inducir grandes deformaciones en una gota –o cualquier objeto embebido–, mientras que los que van de 0 a $+1$ se llaman flujos fuertes ya que su vorticidad es pequeña. Los flujos fuertes son los de mayor relevancia, pues éstos son capaces de inducir grandes deformaciones, así como rotaciones o traslaciones en un objeto embebido en un fluido.

El segundo parámetro del flujo, igualmente relevante para el estudio de la deformación de gotas, es la rapidez de corte – o del cortante–, que corresponde a la magnitud del tensor gradiente de velocidad; esto es, el primero caracteriza la clase de flujos mientras que este segundo la intensidad del flujo. Estos dos parámetros se pueden modificar independientemente en los molinos de dos rodillos estudiados.

En el punto de estancamiento, una gota colocada en tal región permanece estática por tiempo indefinido a pesar de la rotación de los rodillos. Sin embargo, estas condiciones ideales no ocurren en un experimento real debido a que las condiciones no son perfectas; la partícula relevante –sea una gota, o cualquier otro objeto– difícilmente se ubica con exactitud en el punto de estancamiento, y ésta siempre está sujeta a una condición pequeña de deriva. Por ello, es necesario aplicar un esquema de control sobre la posición de la partícula, que mantenga las condiciones de flujo deseadas sobre el objeto y que al mismo tiempo afecte lo menos posible las características básicas de flujo. Para ello, el dispositivo cuenta con un sistema de control computarizado autónomo y que opera en *tiempo real* para mantener indefinidamente la posición de una gota. Este esquema de control es necesario porque si bien las velocidades en el punto de estancamiento son casi nulas, éstas son finitas y aumentan en magnitud para cualquier punto que se aleja de la región central. Entonces, el esquema de control ajusta en forma cuasi-instantánea las velocidades de los rodillos de modo que la residencia de la gota en la vecindad del punto de estancamiento es tan duradera como sea necesario. De esta manera, es factible llevar a cabo estudios de la dinámica de gotas que pueden compararse con estudios numéricos, así como simular numéricamente un gran número de las características del dispositivo experimental que son difíciles de incluir

en los modelos analíticos.

1.2. Geometría del dispositivo experimental

El sistema de control propuesto dicta las velocidades de los rodillos, y como resultado del cambio de las velocidades de rotación el punto de estancamiento se desplaza en forma conocida permitiendo al operario mantener a la gota dentro de la región del punto de estancamiento.

El sistema de control se compone básicamente de los siguientes elementos:

- a) La celda de flujos que contiene a los rodillos.
- b) Dos motores con sus respectivas fuentes de poder.
- c) Dos cámaras fotográficas.
- d) Una computadora.

La figura 1.1 muestra el diseño del arreglo experimental, con los sistemas para la captura de imágenes del lado derecho, y los motores del lado izquierdo.

Los cálculos numéricos y analíticos del flujo fuerte permiten evaluar los factores más importantes para lograr un dispositivo lo más apegado al caso ideal a estudiar. De esta manera, la calidad del control de la posición de la gota depende de maquinados de alta precisión, la resolución del sistema óptico y la rapidez de respuesta del sistema de control—este último dependiente de los sistemas de captura de imágenes.

Desde una perspectiva del algoritmo de control, la región alrededor del punto de estancamiento se convierte en un rectángulo de tolerancia o área de trabajo del cual se conocen todas las propiedades hidrodinámicas en forma instantánea y precisa. En el

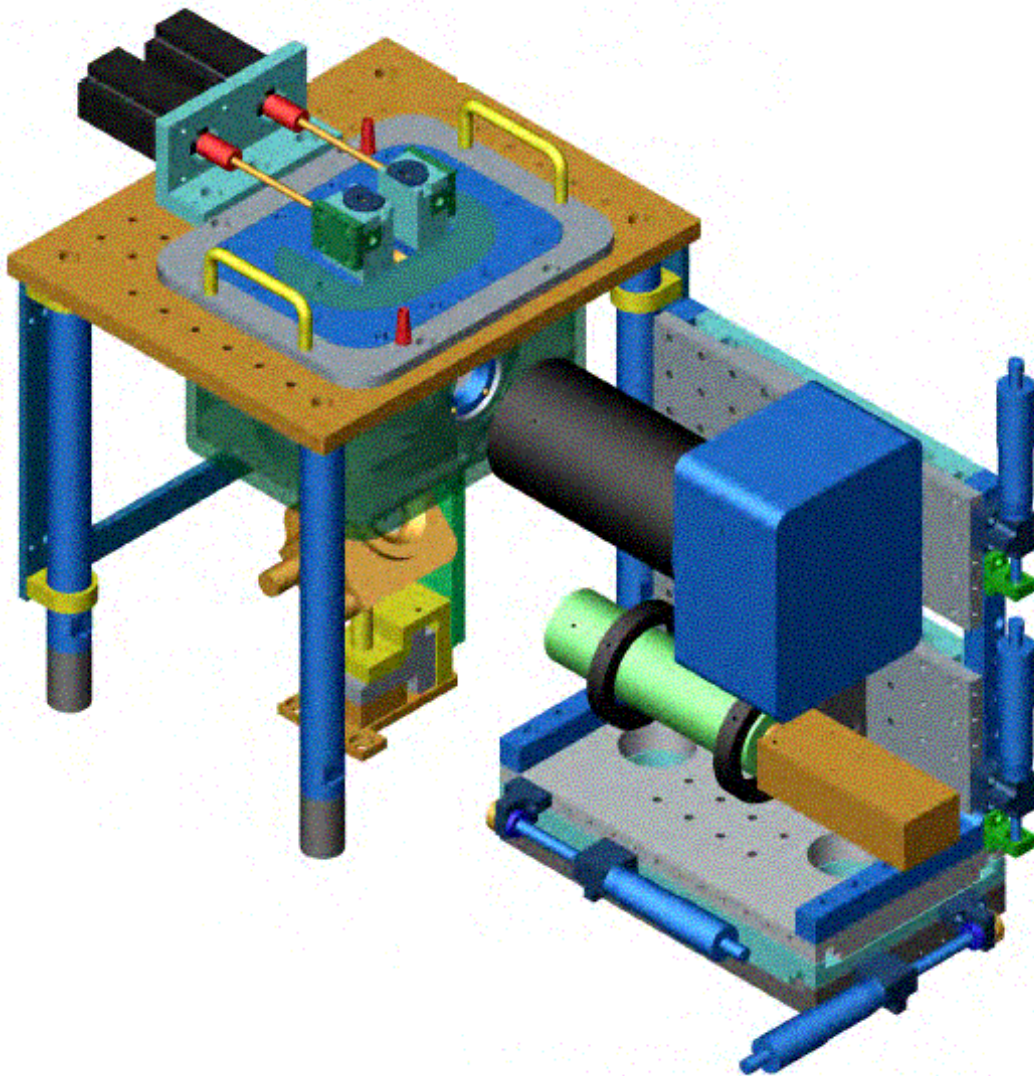


Figura 1.1: Arreglo experimental del molino de dos rodillos.

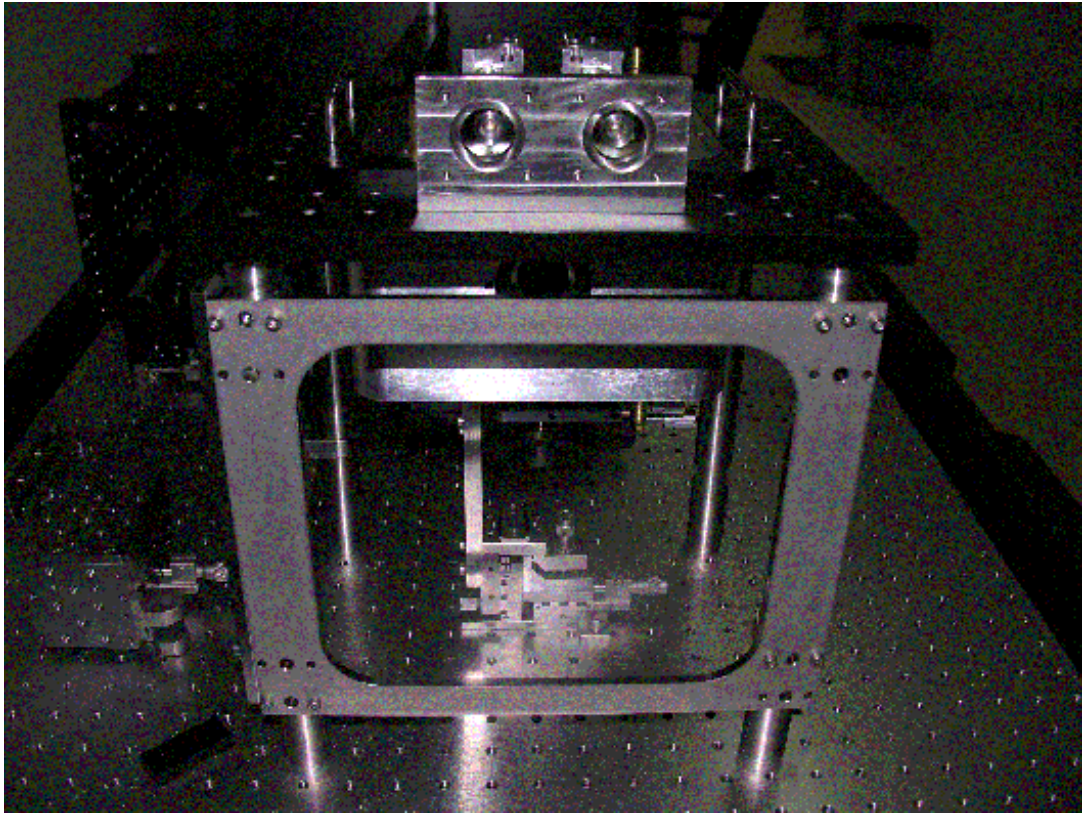


Figura 1.2: Molino de dos rodillos.

experimento, las dimensiones de tal área dependen de la resolución del sistema óptico, de la precisión de los maquinados y del tiempo de respuesta del fluido a los cambios de velocidades angulares en los rodillos que depende de la viscosidad del fluido. Entre más pequeña es el área, entonces el campo de flujo en esa región presenta un comportamiento más lineal –e idealizado, para el cual se conoce una solución analítica exacta–, lo que simplifica el problema de predicción de sus características hidrodinámicas y hace más estable al algoritmo de control.

1.3. Geometría del flujo elongacional con vorticidad

La geometría propuesta para este experimento es idealmente bidimensional y se define básicamente a partir del diámetro de los rodillos y de la distancia entre los centros de los rodillos como se muestra en la Figura 1.3.

Las soluciones del flujo generado por esta geometría están dadas en el trabajo de tesis de doctorado del Dr. Marco Antonio Reyes Huesca [Reyes, 2005]. En dicho trabajo se dan la solución analítica para el caso de un flujo externo no acotado y la solución numérica para el caso acotado. *No acotado* implica la ausencia de fronteras en el flujo externo –un caso ideal. Y es acotado como se desarrolla en el experimento, pues es necesario colocar al molino en una celda para contener el fluido externo. A pesar de que los estudios se realizan en dos dimensiones, los resultados obtenidos se pueden extender a tres dimensiones y en muchos casos es suficiente para las aplicaciones requeridas.

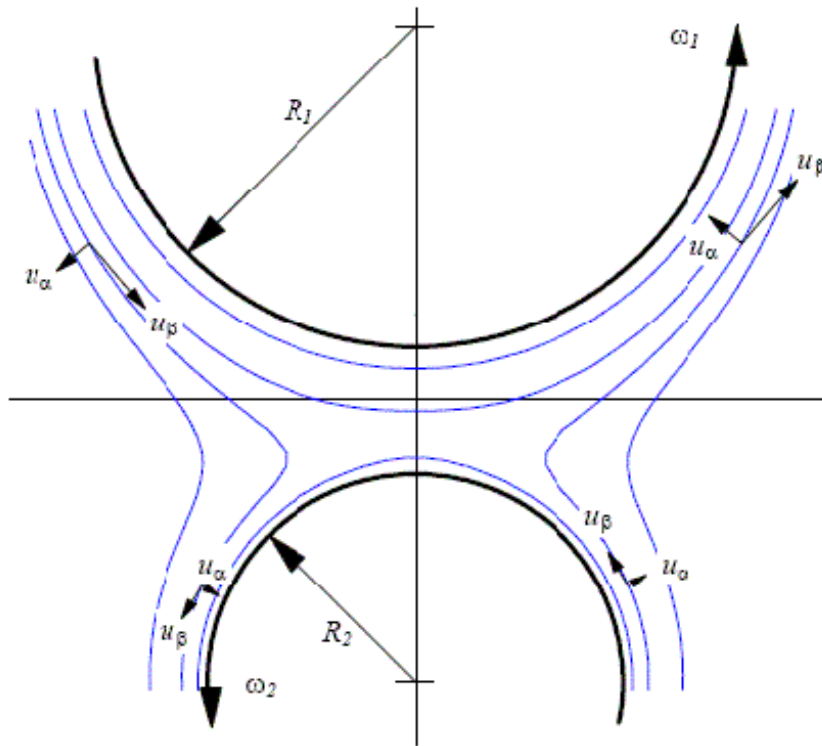


Figura 1.3: Geometría para el molino de dos rodillos.

Parte I

Investigación

Capítulo 2

La captura de imágenes de la dinámica de una gota

La parte correspondiente a este trabajo de tesis consiste básicamente en obtener la información geométrica de la dinámica de una gota en el molino de dos rodillos. Dicha información se obtiene a partir de las imágenes digitales capturadas por el sistema óptico. Para ello, es necesario encontrar la relación entre los datos obtenidos y los valores reales con que se generaron. Dadas las limitaciones físicas de todo sistema óptico real, no es posible obtener al cien por ciento la información real a partir de una imagen; sin embargo nos podemos acercar tanto a ella como nos lo permita la capacidad del equipo empleado y el modo en que éste se utilice. Entonces parte del trabajo consiste en determinar los datos reales a partir de los datos digitales. Y la otra parte se ocupa de analizar los datos para calcular los parámetros geométricos del objeto embebido, haciendo las consideraciones necesarias y que deben ser establecidas en la primera parte.

Cada imagen digital sólo es un conjunto de píxeles que representan a la gota y el entorno real que nos interesa del experimento en un instante dado. Los píxeles tienen una posición en el espacio bidimensional y una magnitud que representa su intensidad entre el color blanco y el negro (en el caso de imágenes en escala de grises). A su vez la imagen representa un instante del experimento. Y entonces el conjunto de imágenes o video representan al experimento para este trabajo de tesis.

La magnitud o el color de los píxeles se utiliza para delimitar las fronteras entre un objeto y otro. En este caso con ello podemos distinguir entre la gota y el fluido. Una vez definidos los cuerpos, entonces sólo la cantidad de píxeles de cada objeto y su ubicación en la imagen son útiles para calcular sus características geométricas.

2.1. MTF

Para el sistema óptico, como en cualquier sistema que genera una señal de salida dada una de entrada, también se tiene una función de transferencia que representa, de manera matemática, las propiedades, calidad y *el comportamiento* del sistema físico, en este caso óptico. A tal función se le conoce como OTF¹. Esta función se descompone a su vez en dos partes que son: la MTF² y la PTF³; es decir, la primera función asociada a la magnitud –o intensidad– de la señal, y la segunda parte asociada a la fase de toda señal electromagnética. Entonces, se puede suponer que la OTF representa mucho más que sólo la capacidad de resolución de un sistema óptico. Obtener la función de transferencia de un sistema óptico real no es tan sencillo y en la mayoría de los casos es prácticamente imposible

¹*Optical Transfer Function* por sus siglas en Inglés.

²*Modulation Transfer Function*.

³*Phase Transfer Function*.

obtener una expresión analítica que la represente. De modo que con frecuencia se obtienen aproximaciones, tan buenas como sea necesario, y con ellas se procede a obtener las señales de respuesta para señales de entradas específicas.

Para el caso del experimento, es imperativo contar con una señal de vídeo capturada con el sistema óptico mostrado en el arreglo (ver Sección 1.2); es con ésta que se trabaja posteriormente para caracterizar posiciones y formas de los objetos embebidos. Y para medir la calidad de la información, se busca una correspondencia entre los objetos y las imágenes obtenidas empleando la MTF del sistema óptico. Como primer paso se obtiene la función de transferencia OTF asociada.

En primer lugar obtendremos lo análogo a la respuesta impulso de un sistema en general. En nuestro caso el sistema óptico se compone de un sistema de lentes buscando así minimizar las diversas aberraciones posibles; este sistema se representa matemáticamente como una sola lente, equivalente al conjunto de lentes del sistema óptico. Para facilitar el manejo matemático, se aplica la transformada de Fourier para expresar de una manera práctica a la función de transferencia de un sistema óptico.

2.2. La transformada de Fourier

La transformada de Fourier, $F(u)$, de una función continua de una variable, $f(x)$, está definida por la ecuación

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx \quad , \quad (2.1)$$

donde $j = \sqrt{-1}$. En forma equivalente, dado $F(u)$, podemos obtener $f(x)$ por medio de la Transformada Inversa de Fourier

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du \quad . \quad (2.2)$$

Por medio de esta ecuación, una función se puede recuperar a partir de su transformada. Se dice que las ecuaciones 2.1 y 2.2 forman el par de transformadas de Fourier. Estas ecuaciones pueden extenderse al espacio bidimensional caracterizado por dos variables

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \quad , \quad (2.3)$$

y, de manera similar para la transformada inversa

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} dudv \quad . \quad (2.4)$$

2.3. Una lente como un sistema lineal

Un sistema lineal está definido a partir de sus relaciones de entrada-salida. Supongamos que una señal de entrada $f(y, z)$, pasando a través de algún sistema óptico, resulta en una salida $g(Y, Z)$. El sistema se dice que es lineal si:

1.-Multiplicando $f(y, z)$ por una constante a produce una salida $ag(Y, Z)$.

2.-Si cuando la entrada se representa como la una suma de dos funciones ponderadas, i.e., $af_1(y, z) + bf_2(y, z)$, entonces la salida tiene la forma $ag_1(Y, Z) + bg_2(Y, Z)$, donde $f_1(y, z)$ y $f_2(y, z)$ generan $g_1(Y, Z)$ y $g_2(Y, Z)$, respectivamente.

Un sistema lineal es además invariante en el espacio si tiene la propiedad de ser estacionario bajo translaciones; esto quiere decir que si la posición de la entrada se cambia, entonces únicamente cambia la posición de la salida, sin alterar su forma funcional. Esta

propiedad es de gran importancia para los sistemas ópticos pues implica que su salida puede considerarse como la superposición lineal de las salidas resultado de cada uno de los puntos individuales que conforman el objeto. Así, una representación de un sistema lineal es:

$$g(Y, Z) = L \{f(y, z)\} \quad .$$

Y utilizando la función Delta de Dirac para expresar la contribución de cada punto del objeto (entrada):

$$g(Y, Z) = L \left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(y', z') \delta(y' - y) \delta(z' - z) dy' dz' \right\} \quad . \quad (2.5)$$

Entonces $f(y, z)$ se expresa como la integral de las funciones delta elementales, cada una de ellas con su respectivo peso dado por la densidad $f(y', z')$. De las propiedades de linealidad del operador se deduce que puede actuar directamente sobre cada una de las funciones elementales, y por lo tanto se tiene que

$$g(Y, Z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(y', z') L \{ \delta(y' - y) \delta(z' - z) \} dy' dz' \quad , \quad (2.6)$$

en donde la cantidad $L \{ \delta(y' - y) \delta(z' - z) \}$ es la respuesta –salida– impulso del sistema. De modo que si se conoce la respuesta impulso de un sistema óptico lineal e invariante, entonces se conoce la salida directamente de la Ec. 2.6.

Ahora si consideramos que nuestras fuentes elementales son coherentes –como sería el caso de señales de láser u objetos iluminados con fuentes de láser coherentes–, entonces las señales de entrada y salida serán campos eléctricos, mientras que si son fuentes no-coherentes, entonces hablaremos de densidades de flujo pues la información de la fase no

puede medirse, y por ello no contiene ninguna información.

Consideremos una fuente autoluminosa y no-coherente como la que se muestra en la Fig. 2.1. Entonces, cada punto sobre el plano objeto PL0 emite luz que el sistema óptico procesa y sale para formar su imagen sobre el plano imagen PL1[Zvérev, 1978]. Si suponemos que la magnificación —o aumento de la lente— entre el plano objeto e imagen es uno, y si $I_0(y, z)$ es la distribución de irradiancia sobre el plano objeto, un elemento $dydz$ localizado en (y, z) emitirá un flujo radiante igual a $I_0(y, z)dydz$. Debido al fenómeno de difracción —de la radiación electromagnética— y la presencia de aberraciones —producto de las imperfecciones del sistema óptico— la fuente puntual de luz se ve *esparcida*, como una mancha difusa, sobre una pequeña región en el plano imagen, en vez de sólo proyectar un punto. El flujo radiante se describe matemáticamente por una función $S(y, z; Y, Z)$ tal que la densidad de flujo que llega al punto imagen de $dydz$ es

$$dl_i(Y, Z) = S(y, z; Y, Z) I_0(y, z) dydz .$$

Esta es la representación de la *mancha* difusa en el plano imagen —asociada a la fuente de luz objeto. Y a la función $S(y, z; Y, Z)$ se le conoce como *función de punto extendido*. Debido al carácter no-coherente de la fuente, las contribuciones de densidad de flujo de cada uno de sus elementos son aditivas y se tiene

$$I_i(Y, Z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S(y, z; Y, Z) I_0(y, z) dydz . \quad (2.7)$$

Si el sistema óptico fuese perfecto, es decir libre de aberraciones y limitado únicamente por el fenómeno de difracción electromagnético, entonces la función de punto ex-

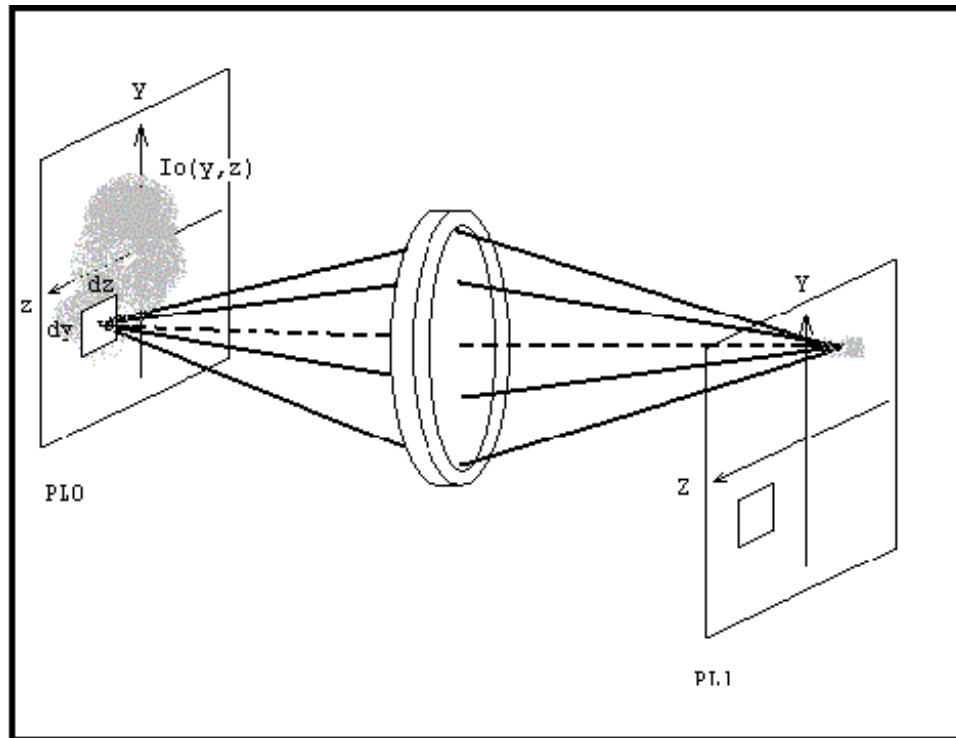


Figura 2.1: Un sistema óptico formando una imagen.

tendido correspondería en forma a la figura matemática resultado de la difracción de una fuente puntual de luz en (y, z) . Si la entrada es un fuente puntual centrada en (y_0, z_0) , es decir, $I_0(y, z) = A S(y - y_0) S(z - z_0)$, en donde la constante A tiene magnitud uno con unidades de irradiancia por unidad de área, entonces de la Ec. 2.7, y utilizando la función Delta para representar a $f(x_0, y_0)$, se tiene que $I_i(Y, Z) = A S(y_0, z_0; Y, Z)$. Si el sistema es invariante en el espacio, una fuente puntual de entrada se puede mover sobre todo el plano objeto sin afectar más que el cambio en la localización de la imagen. De modo que la función de punto extendida es la misma para cualquier punto (y, z) ; así, bajo circunstancias de una señal no-coherente y un sistema invariante en el espacio:

$$I_i(Y, Z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_0(y, z) S(Y - y, Z - z) dydz \quad . \quad (2.8)$$

Un parámetro muy útil para evaluar el comportamiento (y la calidad) de un sistema es el *contraste* o *modulación*, definido por

$$Modulación = \frac{I_{máx} - I_{mín}}{I_{máx} + I_{mín}} \quad . \quad (2.9)$$

La Figura 2.2 muestra las respuestas de modulación de la imagen, en función de la frecuencia espacial, para dos sistemas ópticos hipotéticos: 1 y 2. Si uno de ellos se acopla al detector cuya frecuencia de corte se muestra, entonces y de acuerdo al gráfico la lente 1 tiene una mayor frecuencia de resolución, mientras que la lente 2 ofrece una mejor imagen cuando se acopla al detector; esto es, el ojo humano percibe la imagen mejor bajo condiciones de contraste altas, como sería este segundo Caso.

Con frecuencia se utiliza un *patrón de entrada* para la caracterización de un sistema

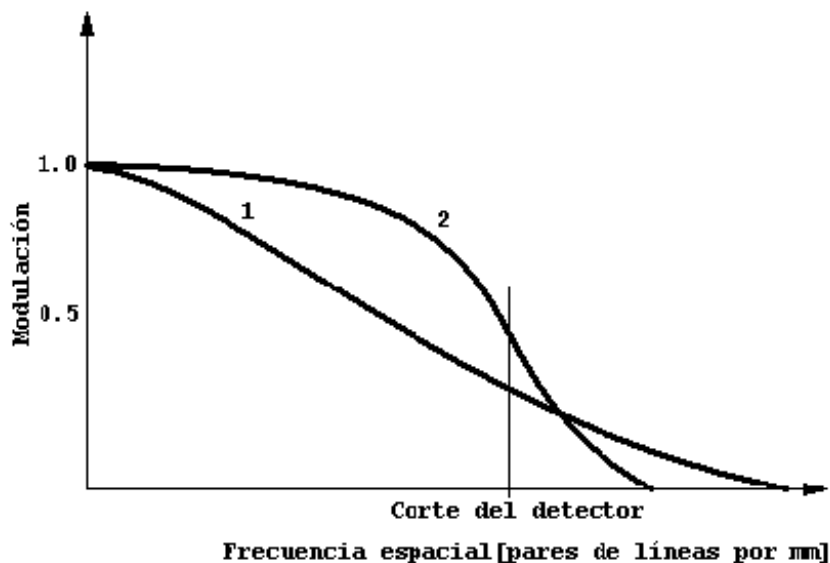


Figura 2.2: Modulación vs Frecuencia en dos lentes diferentes

óptico. Por ejemplo, un *patrón de barras cuadradas –blancas y negras–* ofrece una señal de entrada equivalente a una secuencia de pulsos cuadrados equidistantemente espaciados. El contraste de la imagen es entonces una superposición de variaciones de contraste debidas a las componentes de Fourier constitutivas. Si el sistema óptico funciona como un operador lineal, entonces transforman una entrada senoidal en una salida senoidal no distorsionada. A pesar de ello, las distribuciones de irradiancia a la entrada y salida no serán idénticas. La amplificación del sistema afectará a la frecuencia espacial de la salida. La difracción y las aberraciones reducirán la amplitud de la senoide (contraste) de salida. Y finalmente, las aberraciones asimétricas y un mal centrado de los elementos producirán un corrimiento en la posición de la senoide a la salida correspondiente a un corrimiento de fase.

Considerando al objeto como un conjunto de componentes de Fourier, la forma en

que el sistema óptico transforma estas componentes en las correspondientes armónicas de su imagen es la idea principal del proceso. La función mediante la cual esta transformación se realiza se denomina *la función de transferencia óptica* FTO. Esta función representa una cantidad compleja dependiente de la frecuencia espacial, cuyo módulo es la *función de transferencia de modulación* FTM, y cuya fase es la *función de transferencia de fase* FTF. La FTM es una medida de la reducción en contraste del objeto en la imagen sobre todo el espectro mientras que la FTF representa el corrimiento relativo en la fase. En general, los corrimientos de fase en sistemas ópticos centrados ocurren principalmente fuera del eje y la FTF suele entonces despreciarse. La FTM se ha convertido en una herramienta analítica muy utilizada para especificar el desempeño de un sistema óptico.

A partir de la Ec. 2.8, la imagen se puede expresar como la convolución de la irradiancia del objeto y la función de punto extendido, es decir,

$$I_i(Y, Z) = I_0(y, z) * S(y, z) . \quad (2.10)$$

Y en el dominio de la frecuencia por medio de la transformada de Fourier,

$$F \{I_i(Y, Z)\} = F \{I_0(y, z)\} F \{S(y, z)\} . \quad (2.11)$$

De manera que el espectro de frecuencias de la distribución de irradiancia de la imagen es igual al producto de la transformada de la distribución de irradiancia del objeto y la transformada de la función extendida. Es decir, $F \{S(y, z)\}$ transfiere el espectro del objeto en el espectro de la imagen. Éste es precisamente el resultado de la FTO, quedando definida como

$$F \{k_y, k_z\} = F \{S(y, z)\} . \quad (2.12)$$

El módulo de la FTO, Ec. 2.12, genera un cambio en las amplitudes de las componentes de frecuencia del espectro del objeto, mientras que su fase altera la fase de estas componentes. De la Ec. 2.10, únicamente el factor $S(y, z)$ es dependiente de nuestro sistema óptico real y por lo tanto es razonable que la función de punto extendida sea la contraparte espacial de la función de transferencia óptica.

2.3.1. Un ejemplo sencillo

Supóngase que $I_0(z) = 1 + a \cos(k_z z + \varepsilon)$ representa a nuestro objeto –en donde por simplicidad sólo se manifiesta una dimensión. Utilizando la Ec. 2.10 se tiene

$$I_i(Z) = I_0(z) * S(z) = S(z) * I_0(z) , \quad (2.13)$$

o de otra forma,

$$I_i(Z) = \int_{-\infty}^{\infty} \{1 + a \cos[k_z(Z - z) + \varepsilon]\} S(z) dz .$$

Ahora, desarrollando la expresión anterior se tiene que

$$\begin{aligned} I_i(Z) = & \int_{-\infty}^{\infty} S(z) dz + a \cos(k_z Z + \varepsilon) \int_{-\infty}^{\infty} \cos k_z z S(z) dz + \\ & + a \sin(k_z Z + \varepsilon) \int_{-\infty}^{\infty} \sin k_z z S(z) dz . \end{aligned}$$

Reconociendo en la segunda y tercera integral –del lado derecho– expresiones de las transformadas seno y coseno de $S(z)$, entonces la expresión anterior puede reescribirse

como

$$I_i(Z) = \int_{-\infty}^{\infty} S(z) dz + F_c \{S(z)\} a \cos(k_z Z + \varepsilon) + F_s \{S(z)\} a \sin(k_z Z + \varepsilon) . \quad (2.14)$$

Empleando ahora la representación de Argand donde

$$F \{S(z)\} \equiv F(k_z) = M(k_z) e^{i\Phi(k_z)} ,$$

se obtiene así la expresión equivalente de la Ec. 2.14 como:

$$I_i(Z) = \int_{-\infty}^{\infty} S(z) dz + aM(k_z) \cos[k_z Z + \varepsilon - \Phi(k_z)] . \quad (2.15)$$

Como se puede observar, ésta es una función similar a aquella de la señal de entrada: Ec. 2.13, pero con una amplitud y fase distinta como resultado de la FTO (función de transferencia óptica). En un caso perfecto se obtiene la misma función, alterada ésta a lo mucho en su magnitud. Sin embargo, a pesar de las consideraciones, ello no es así. Ello es consecuencia del tamaño –apertura– finito de la lente, pues es incapaz de capturar toda la luz emitida –en todas direcciones, y consecuentemente se conoce sólo parcialmente la información de la onda electromagnética– por el objeto.

2.4. Parámetros en un sistema de generación de imágenes

En la sección anterior hemos hecho un estudio para caracterizar a una lente. Sin embargo un sistema óptico se compone de más elementos que solo una lente o un conjunto de ellas, como a continuación se explica. Ello se debe a que en gran medida la calidad de una imagen consiste de un número fundamental de parámetros como se muestra en la Figura 2.3. Éstos son[Edmund Optics]:

- El Campo de Vista (FOV, i.e., Field of View): Es el área del objeto bajo inspección; es decir, corresponde a la porción del objeto capaz de ser capturada por el sensor de la cámara.
- La Magnificación Primaria. Es la propiedad de la lente que relaciona al campo de vista y el tamaño del sensor en el sistema óptico .

$$MAGP = \frac{\text{Tamaño Sensor [mm]}}{\text{Campo de Vista [mm]}}$$

Y así por ejemplo una *magnificación* de 1 implica que el tamaño de la imagen corresponde con el tamaño del sensor.

- La Distancia de Trabajo: es la distancia entre la lente –la parte frontal del primer elemento óptico– y el objeto.
- La Resolución: El detalle más pequeño que el sensor puede detectar o resolver.
- La Profundidad de Campo (DOF: Depth of Field): La profundidad –distancia– máxima en la dirección del objeto que la lente puede resolver con precisión.
- La Distorsión. Se presenta como una deformación consecuencia de un valor de la magnificación heterogéneo, el cual depende de la distancia al centro de la lente –condición ideal. De modo que no se pierde información sino que ésta se reconstruye de modo erróneo. La Distorsión es proporcional a la distancia de la imagen y a la del centro de la lente. Así, entonces es un defecto que se puede corregir *a posteriori*.
- El Tamaño del Sensor: El tamaño del sensor, típicamente especificado por su dimensión horizontal.



Figura 2.3: Parámetros fundamentales de un sistema de adquisición de imágenes.

2.5. Resolución, contraste y gráficas de MTF

La gran importancia de la función de transferencia de modulación o MTF se puede comprender considerando la relación entre la resolución y el contraste. La resolución es una medida de la capacidad del sistema de reproducir los detalles de un objeto. Por ejemplo, imagínese un objeto conformado por un par de cuadros negros en un fondo blanco. Si en la imagen capturada, los cuadros cubren pixeles vecinos, entonces éstos se verán como una sola figura oscura y con forma rectangular. De manera que para distinguir los cuadros objeto en la imagen se requiere dejar un espacio –blanco– suficientemente amplio entre ellos. Determinando la mínima distancia requerida para lograr la distinción, entonces se determina la capacidad máxima de resolución del sistema de captura de imágenes. Esta idea de alternar cuadros -o rectángulos- blancos y negros se cuantifica frecuentemente con base en los "pares de líneas por milímetro" [ppmm] .

Existen dos diferentes resoluciones pero que están estrechamente relacionadas. Una es la resolución del tamaño mínimo de los elementos del objeto que pueden ser resueltos por el sistema de lentes y la otra es la resolución que la cámara –el sensor propiamente– puede capturar o almacenar. La resolución de pares de líneas que el sensor distingue no es mayor que la mitad del número total de pixeles que abarcan al sensor puesto que se requieren al menos dos pixeles para representar un cambio en la imagen. Tales resoluciones están relacionadas por la magnificación primaria del sistema.

El límite de resolución de cualquier sistema óptico se evalúa experimentalmente observando –midiendo– la respuesta del sistema para una clase de objetos como la que se muestra en la Figura 2.4. Ésta es una tarjeta estándar para calificar la resolución ópti-

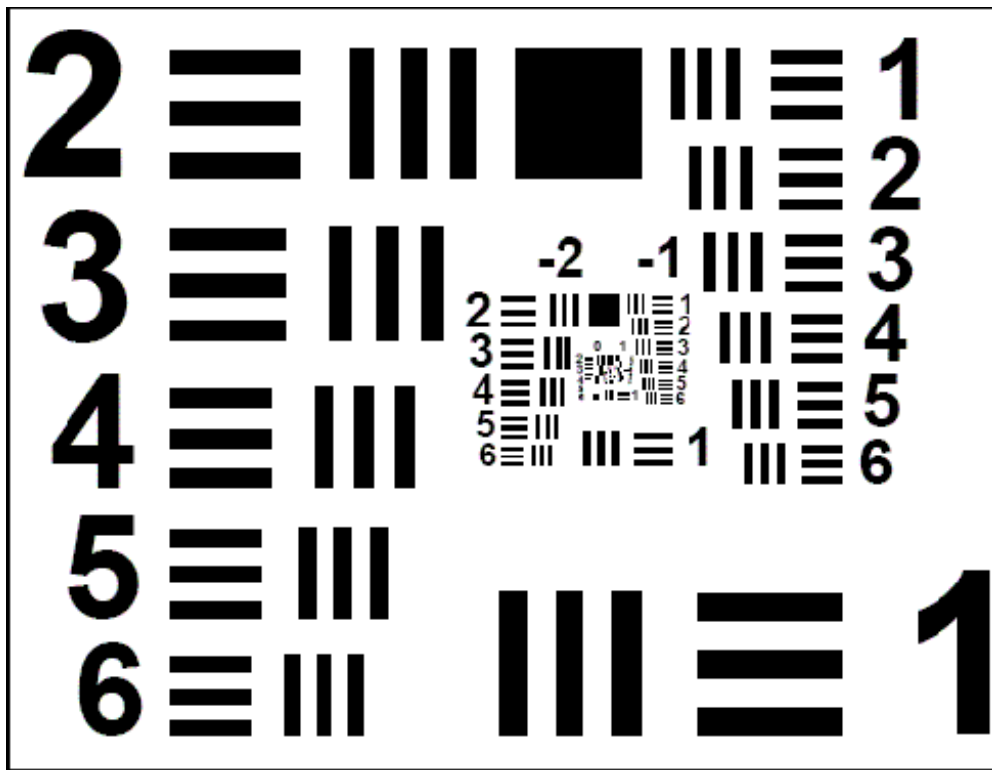


Figura 2.4: Tarjeta de barras.

ca. El patrón básico utilizado consiste en tres pares de líneas negras orientadas vertical y horizontalmente, separadas por dos líneas blancas; esto es, cada Elemento tiene seis barras –negras– orientadas de forma horizontal y vertical sobre un fondo blanco. Dado que cada sextilla de líneas tiene diferente grosor de sus líneas, entonces un *Elemento* representa a una Frecuencia espacial diferente. La tarjeta se organiza en *Grupos* conformados por seis Elementos cada uno. Cada Frecuencia se identifica por un Grupo y un Elemento de dicho Grupo, de modo que existen seis Frecuencias diferentes para cada Grupo. Podemos utilizar una expresión aproximada para calcular la frecuencia dado un grupo y un elemento.

$$Frecuencia [ppmm] = 2^{(grupo + \frac{Elemento - 1}{6})} \quad (2.16)$$

Igualmente, se puede emplear la siguiente Tabla en la que se indican las Frecuencias correspondientes al Elemento; la resolución máxima es de 228 pares de líneas por milímetro, correspondiente al Elemento seis del Grupo siete.

		Número de pares de líneas por milímetro									
		Número de Grupo									
	-2	-1	0	1	2	3	4	5	6	7	
1	0.250	0.500	1.00	2.00	4.00	8.00	16.00	32.0	64.0	128.0	
2	0.280	0.561	1.12	2.24	4.49	8.98	17.95	36.0	71.8	144.0	
3	0.315	0.630	1.26	2.52	5.04	10.10	20.16	40.3	80.6	161.0	
4	0.353	0.707	1.41	2.83	5.66	11.30	22.62	45.3	90.5	181.0	
5	0.397	0.793	1.59	3.17	6.35	12.70	25.39	50.8	102.0	203.0	
6	0.445	0.891	1.78	3.56	7.13	14.30	28.50	57.0	114.0	228.0	

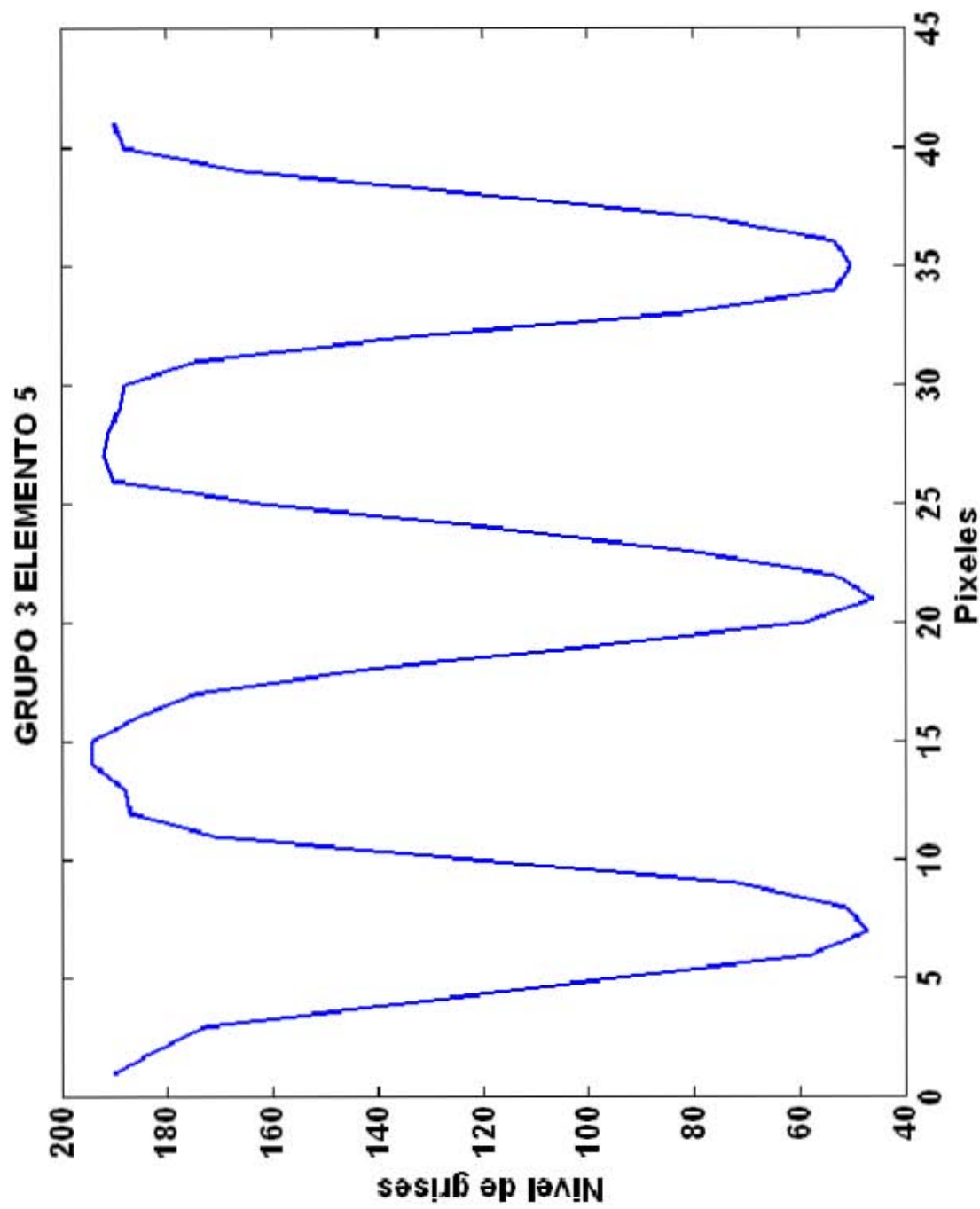
Con la expresión 2.16, naturalmente se puede calcular las frecuencias correspondientes a los Grupos superiores; sin embargo, experimentalmente no vale la pena si no se cuenta aun con la tarjeta que permita medir las resoluciones del sistema óptico a partir de la respuesta a dichos Grupos.

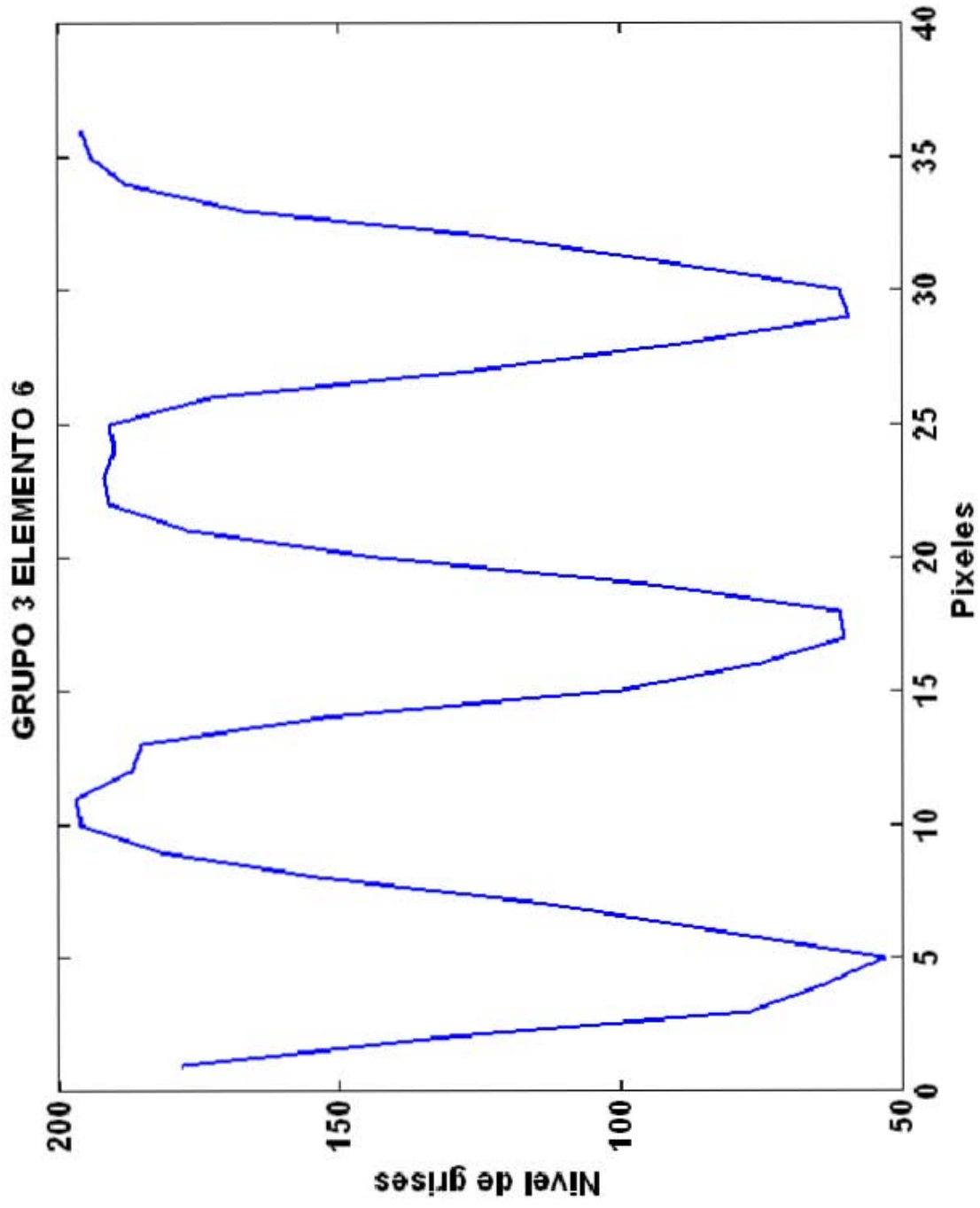
2.5.1. Encontrando la resolución en una cámara Nikon

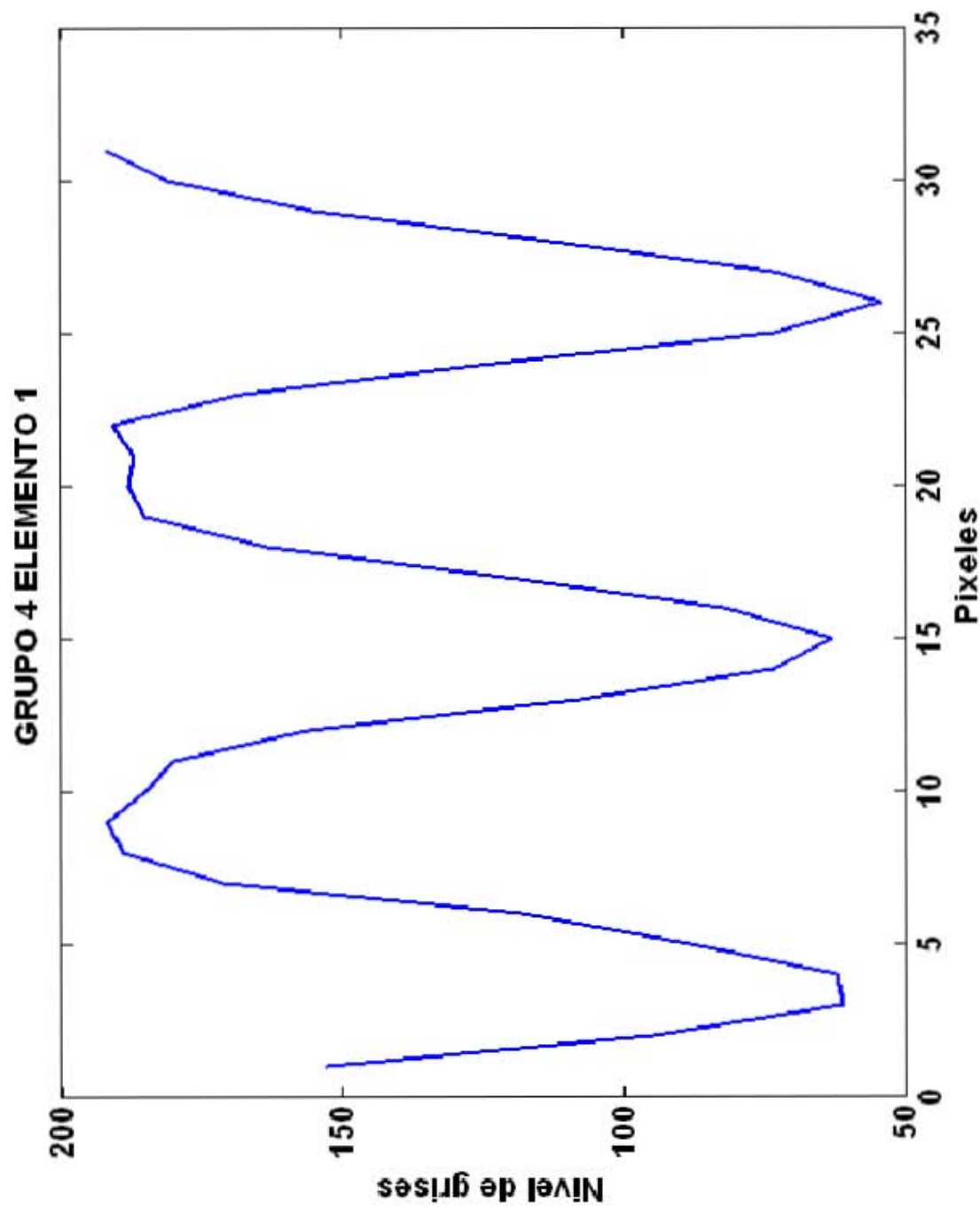
El modelo en estudio es una cámara Nikon D70s. Con ésta se tomaron 10 fotografías de una tarjeta estándar. Las Gráficas a continuación, muestran los resultados obtenidos –en escala de grises– para diferentes frecuencias de una tarjeta de prueba fabricada por Edmund Optics y siguiendo el estándar de la Fuerza Aérea de los EEUU. El sensor tiene una escala de grises de 8 bits, por lo que resuelve de 0 a 255 tonalidades. Los resultados se resumen en la Tabla 2.5 y se muestra una Gráfica aproximada de MTF en la figura 2.6. La gran limitante del método anterior es que no se obtiene una curva continua de la MTF sino más bien sólo tantos puntos discretos de ella como imágenes de diferentes frecuencias apliquemos.

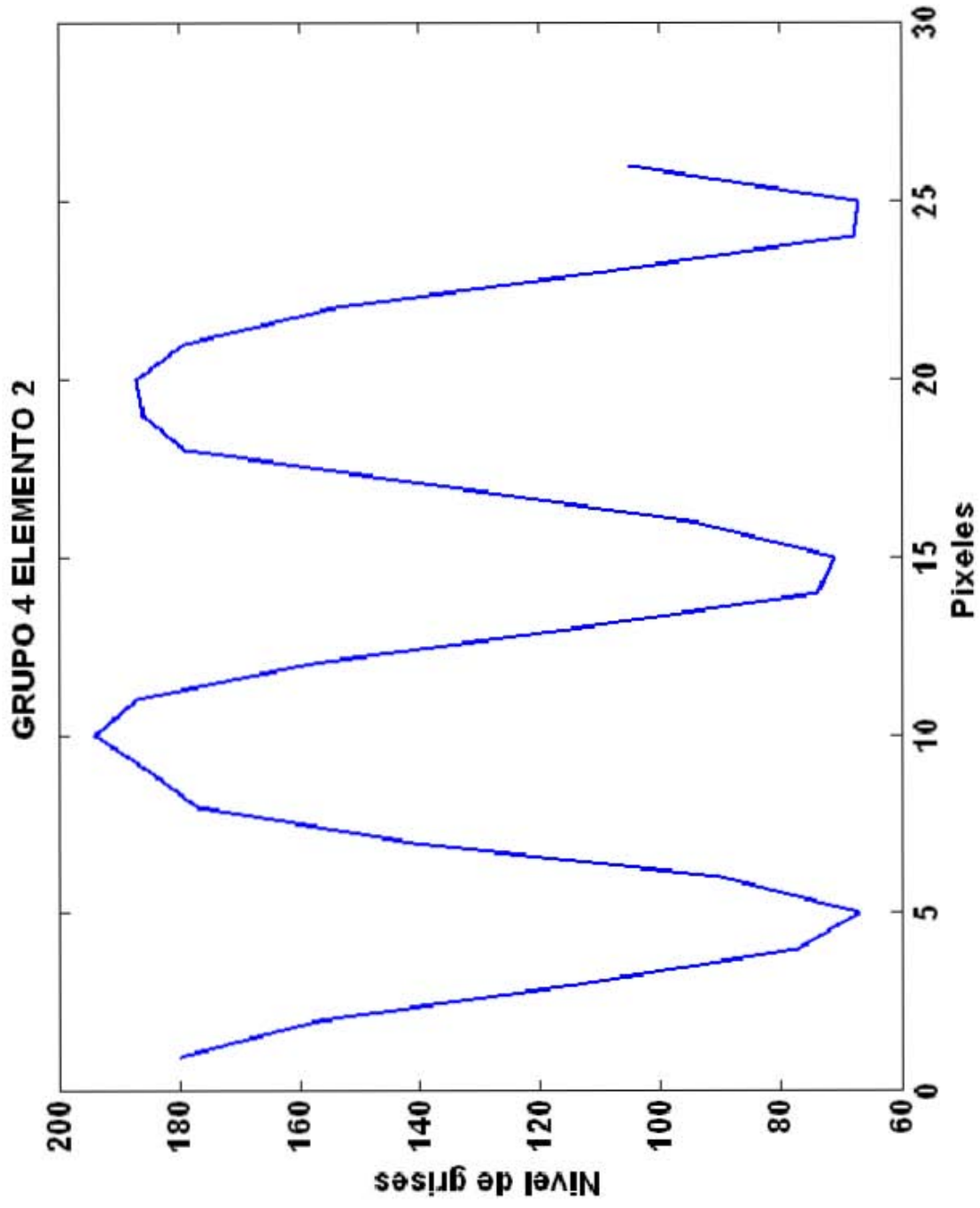
Analizando ahora el caso de un detector monocromático que captura imágenes en blanco y negro exclusivamente, entonces el contraste describe la capacidad del sistema de distinguir un color –tonalidad del gris– de otro; por ejemplo, el caso más simple es entre el blanco y el negro –puros– que corresponderían a tonalidades de 0 –para el negro– y de 255 (blanco) en un sistema óptico ideal. En la *vida real* los diferentes colores se encuentran entre blanco y negro, es decir son descritos por una escala de grises. En la caracterización de las propiedades de contraste de la imagen resultante, las propiedades de las lentes, los sensores y de los sistemas de iluminación juegan un rol determinante.

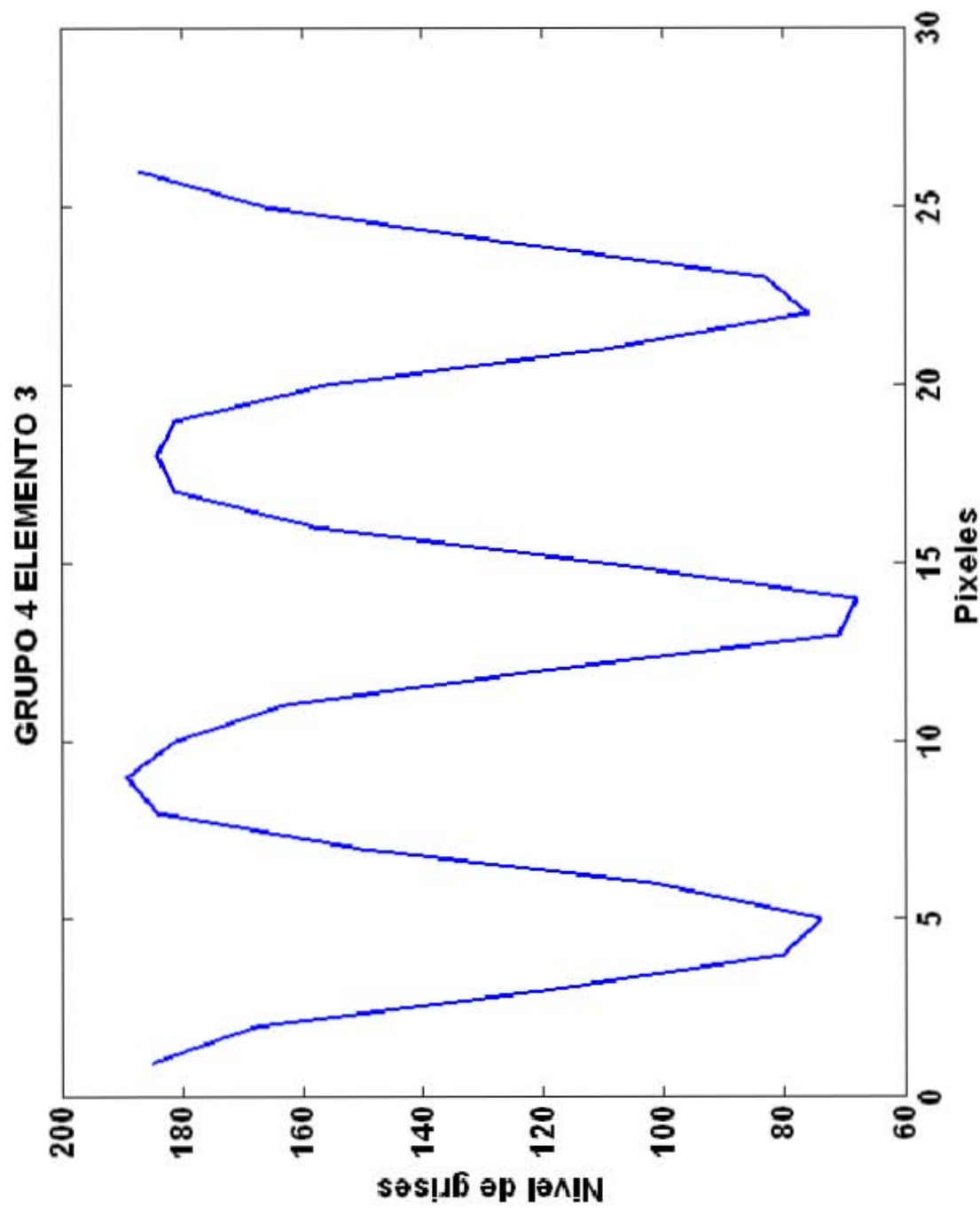
El contraste de un sistema óptico se define generalmente a partir del porcentaje que reproduce del contraste del objeto. Así, el contraste es la separación relativa entre claros y oscuros y es importante para distinguir detalles relevantes en la imagen. Por ello, el contraste es esencialmente la resolución. Con frecuencia, la habilidad de un sensor de

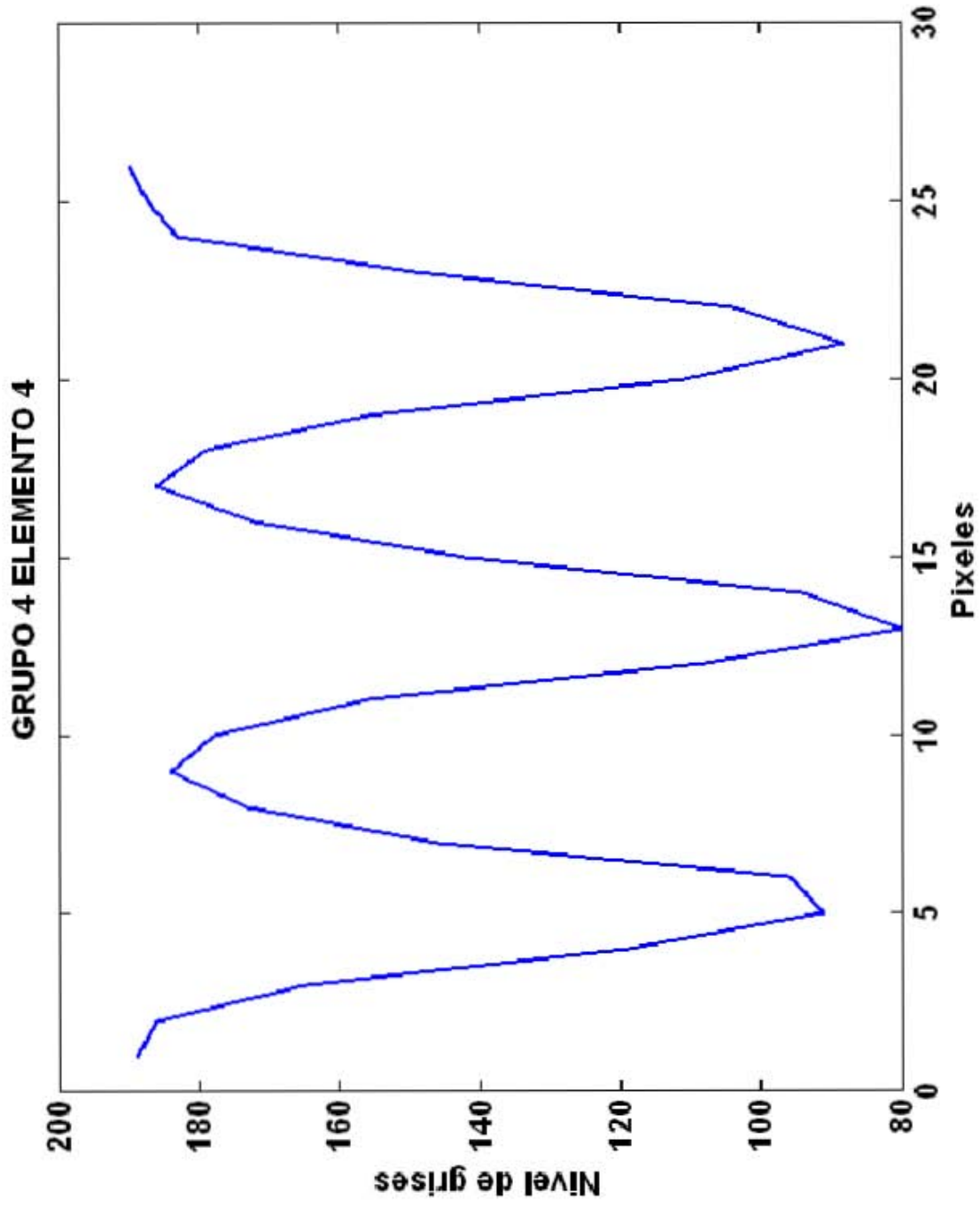


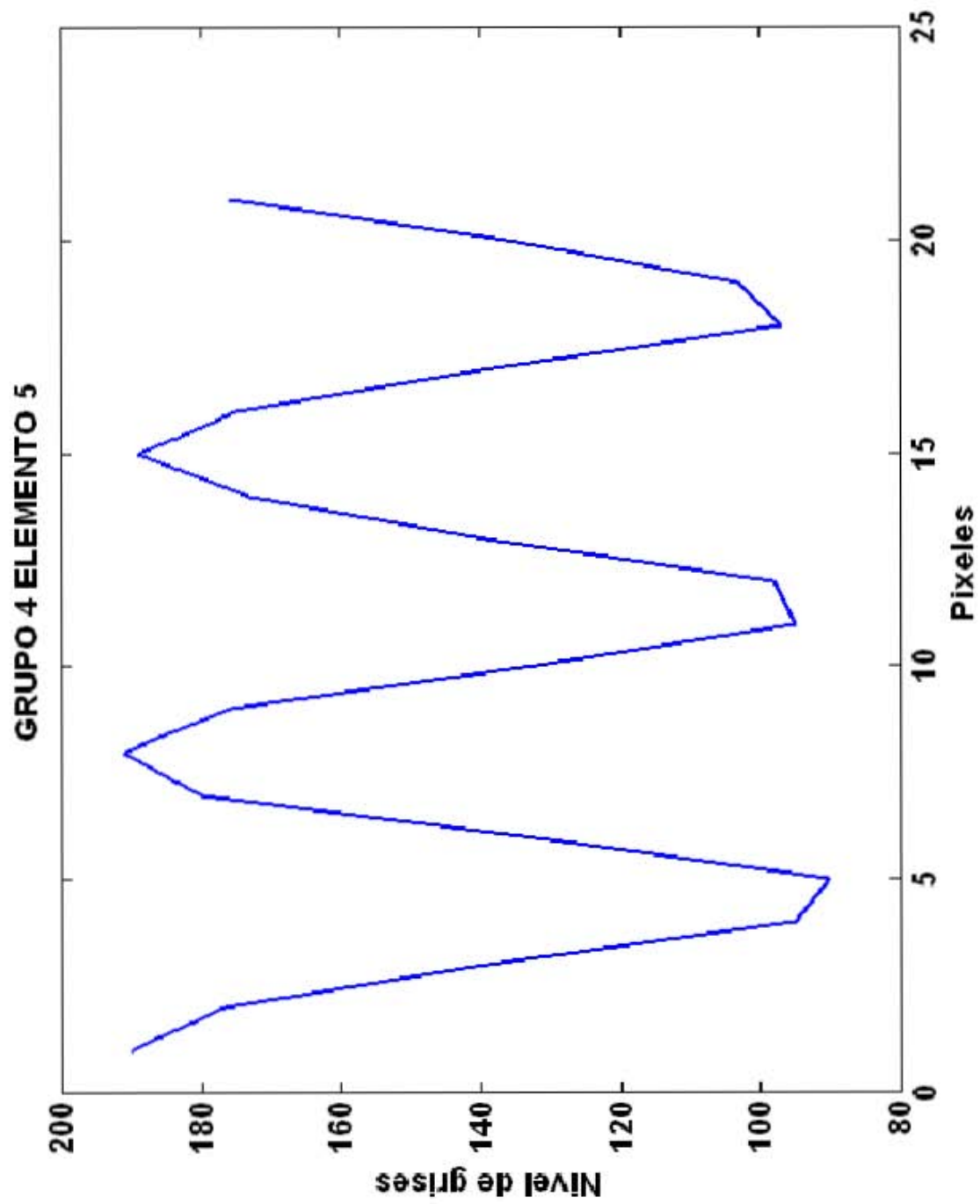


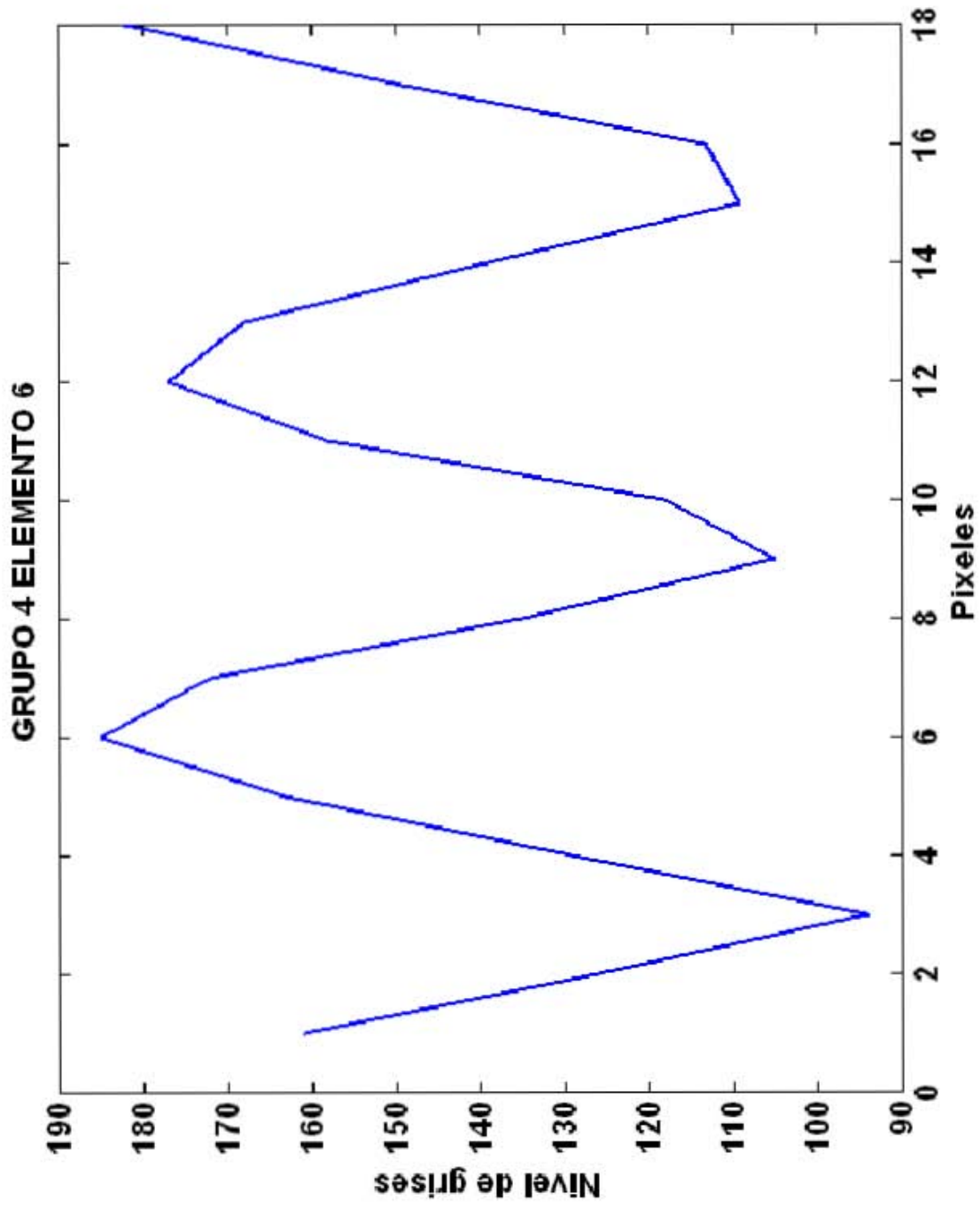


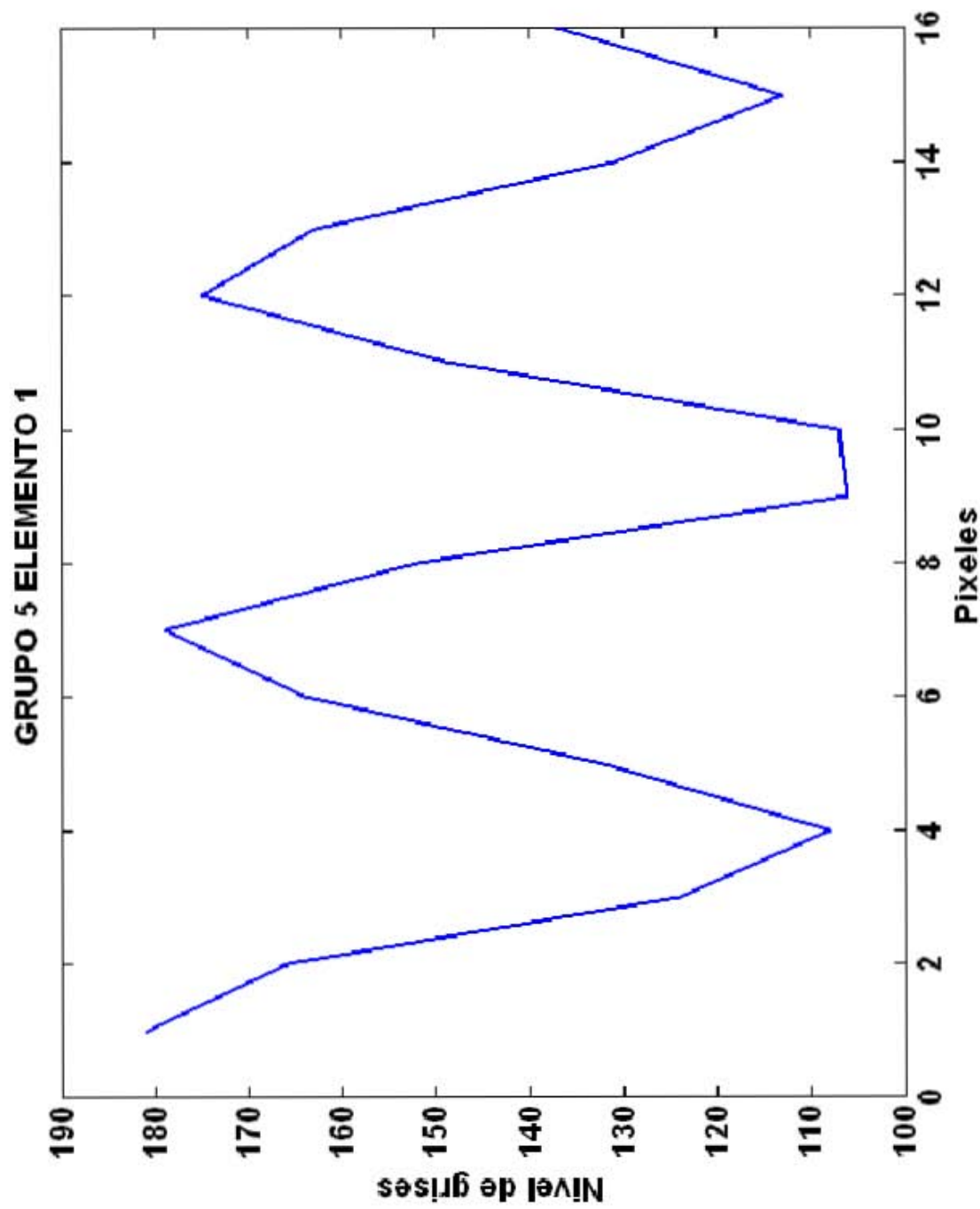


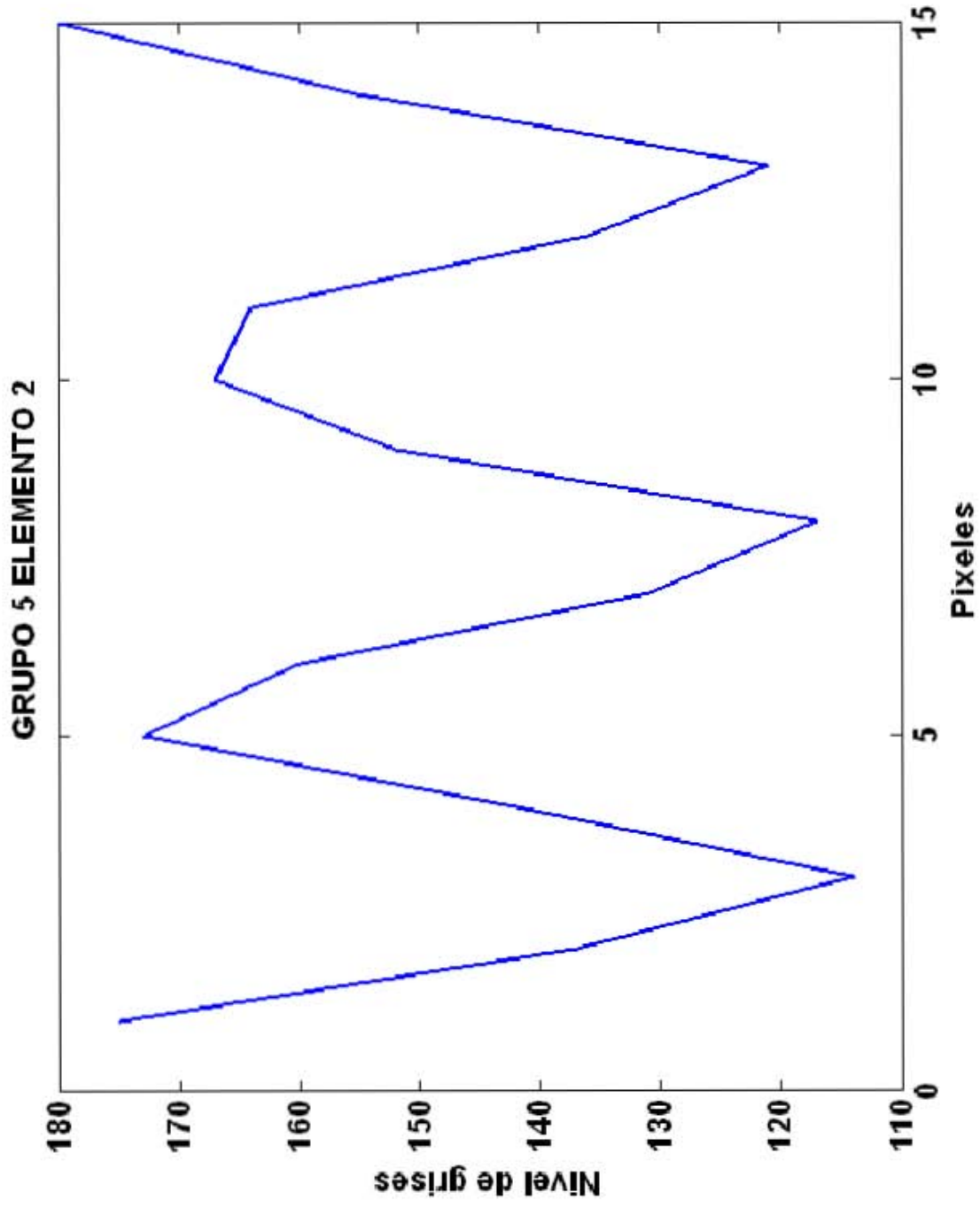












Grupo, elemento	frecuencia[ppmm]	$C=(I_{max}-I_{min})/(I_{max}+I_{min})$
3,5	12.7	$(194-46)/(194+46)=0.62$
3,6	14.3	$(197-53)/(197+53)=0.58$
4,1	16	$(192-54)/(192+54)=0.56$
4,2	17.95	$(194-67)/(194+67)=0.49$
4,3	20.16	$(189-68)/(189+68)=0.47$
4,4	22.62	$(190-80)/(190+80)=0.41$
4,5	25.39	$(191-90)/(191+90)=0.36$
4,5	28.5	$(185-94)/(185+94)=0.33$
5,1	32	$(181-106)/(181+106)=0.26$
5,2	36	$(180-114)/(180+114)=0.22$

Figura 2.5: Contraste en una cámara nikon D70S.

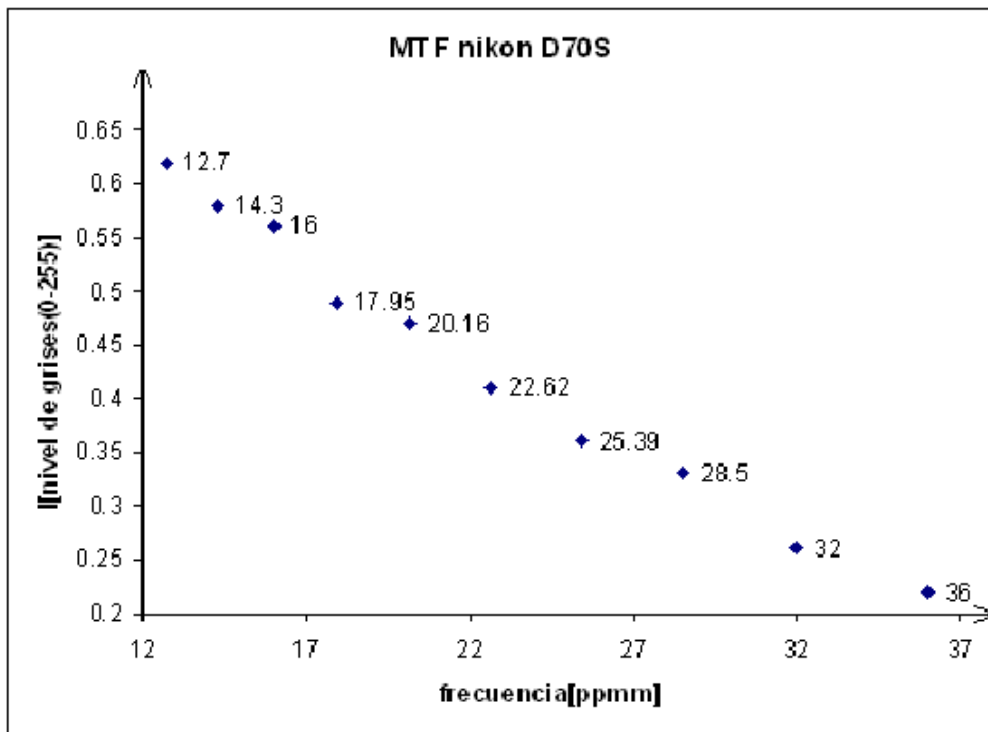


Figura 2.6: Un ejemplo de MTF experimental.

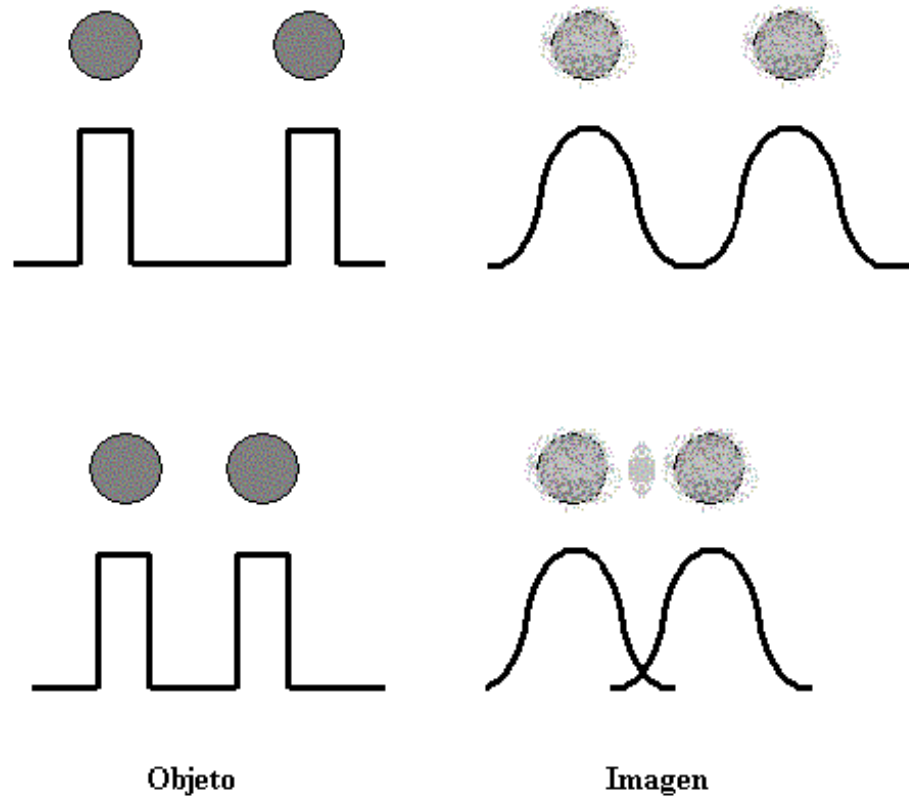


Figura 2.7: El contraste disminuye a medida que los objetos se acercan.

reproducir el contraste se especifica en decibeles –en el caso de cámaras analógicas– o en bits, para cámaras digitales.

Las capacidades de resolución y de contraste de un sistema se definen de forma individual, pero debido a su gran inter-dependencia es conveniente presentarlas como una sola característica o propiedad del sistema.

Considere dos puntos colocados uno cerca del otro como se muestra en la Figura 2.7.

Debido a la naturaleza intrínseca de la luz, aun cuando se tuviese una lente dis-

añada perfectamente ésta no reproduciría cabalmente los detalles y el contraste del objeto. Por ejemplo, si en la Figura anterior los puntos se alejan –entre sí– entonces será más fácil distinguir uno del otro; pero aún así, se presentan las contribuciones no deseables –espúreas– debidas a la difracción de la luz. Es por ello que la resolución del sistema depende de la separación de los puntos, de la difracción de la luz y de la habilidad del sistema de detectar el contraste.

Al analizar la calidad de un sistema óptico generalmente miramos la resolución que presenta, y entre mayor es, entonces decimos que el sistema es mejor. En este caso, un sistema tiene una mayor resolución entre más capacidad tenga de identificar patrones de barras paralelas blancas y negras alternadas, si éstas son muy angostas. De manera que en el límite tales barras son líneas. Tal caso no se presenta en la realidad y correspondería a un sistema con resolución perfecta o infinita.

Así vemos que todo sistema óptico presenta un límite de resolución a partir del cual ya no le es posible distinguir entre una línea (barra) y otra, dicho límite se conoce como *rayado de Ronchi*. De lo anterior podemos entonces definir una frecuencia espacial que se mide en pares de líneas por unidad de longitud por ejemplo pares de líneas por milímetro [*ppmm*].

Sin embargo no es posible caracterizar bien a un sistema óptico tan solo conociendo su capacidad de resolución. Incluso puede darse el caso de que no funcione correctamente en frecuencias inferiores si solo se considera una buena respuesta a altas frecuencias.

Bien, como podemos observar en la figura 2.6 a un contraste del 40 % tenemos una frecuencia aproximada de 22 [*ppmm*]. De modo que con esta cámara podemos captar

NIKON D70S (captación de imagen)

Sensor de imagen	Tipo	CCD
	Dimensiones	23.70 x 15.60 mm
Filtro de color	RGB	
Espacio de color		
Resolución	Total	6.24 MP
	Efectiva	6.10 MP

Figura 2.8: Especificaciones de fabricante para una cámara nikon D70S.

con buenos resultados hasta dicha resolución. Si vamos a las especificaciones de la cámara, ver la Figura 2.8 encontraremos claramente que la resolución que puede soportar el sensor (CCD) es mucho mayor. Considerando el tamaño del sensor y el número de píxeles o células en que se divide el CCD podemos entonces calcular el tamaño aproximado de la célula que representa un píxel efectivo. Y si se utiliza una lente con magnificación unitaria, entonces la resolución máxima corresponde con el doble del tamaño del píxel. Así entonces podemos desarrollar una simple expresión 2.17, en donde además consideramos que los píxeles son cuadrados.

$$resolucion_aproximada = \frac{1}{2 \times \sqrt{\frac{Area}{NoCeldas}}} [ppmm] \quad (2.17)$$

Con la expresión anterior obtenemos una resolución de : 64 ppmm.

2.6. Utilizando la MTF en la selección de una lente

Cada componente de un sistema óptico tiene una curva de MTF asociada –aún a los elementos no ópticos como cables, tarjetas de captura de datos, etc.– y se les puede asociar una función de transferencia. La MTF de cada elemento describe los efectos que estos causan en la imagen. Y de ello depende el contraste y la resolución. Es útil saber esto para evitar componentes que sólo incrementen el costo económico de nuestro sistema y no ayuden en la obtención de una imagen con la calidad que necesitamos. Considere, por ejemplo, un sistema de visión para una máquina de una línea de ensamblado. Si asumimos que no se requieren imágenes con gran detalle para distinguir entre un componente y otro, entonces pondremos énfasis en el contraste más que en la resolución, es decir a baja resolución deseamos alto contraste. Así tendremos un sistema rápido (para ensamblar) con baja resolución en vez de un sistema lento con alta resolución.

La MTF describe el contraste y la resolución, sin embargo en todo experimento hay otros factores que también afectan la calidad de nuestra imagen tales como la profundidad de campo y errores de geometría. Físicamente, la difracción de la luz limita el desempeño una lente, siendo el límite de difracción de una lente afectado por el tamaño de la apertura. La apertura es inversamente proporcional al parámetro f que describe la habilidad de la lente para concentrar la luz de la imagen. Como la apertura de la lente decrece cuando f crece, entonces concluimos que el límite de difracción dicta que el menor detalle que puede proporcionar una lente es proporcional a el parámetro f . Frecuentemente sin embargo el factor limitante principal de una lente no es la difracción sino los errores ópticos y la tolerancia en manufactura son quienes limitan más el desempeño.

Capítulo 3

La imagen digital y el filtrado en el dominio de la frecuencia

3.1. Modelo de imagen

El concepto de imagen acepta una representación en términos de una función bidimensional $f(x, y)$ donde el valor de f representa la intensidad luminosa en este punto. Puesto que la luz es una forma de energía, $f(x, y)$ debe ser estrictamente mayor que cero así como finita; es decir,

$$0 \leq f(x, y) < \infty \quad . \quad (3.1)$$

A la intensidad f de una imagen monocromática en las coordenadas (x, y) se le denomina nivel de gris l de la imagen en este punto y está en el intervalo

$$L_{mín} \leq l \leq L_{máx} \quad . \quad (3.2)$$

En teoría, el único requisito sobre $L_{mín}$ es que sea positiva y sobre $L_{máx}$ que sea finita. Al

intervalo $[L_{mín}, L_{máx}]$ se le denomina escala de grises. Una práctica habitual consiste en desplazar numéricamente este intervalo hasta el intervalo $[0, L]$, donde en la escala $l = 0$ se considera como negro y $l = L$ se considera como blanco. Todos los valores intermedios son tonos de gris que varían de forma continua y monotónica entre el negro y el blanco[Gonzalez, 2002].

3.2. Muestreo y cuantificación

Para el procesamiento digital, una función de imagen $f(x, y)$ requiere ser digitalizada. La digitalización de las coordenadas espaciales (x, y) se denomina *muestreo* de la imagen y la digitalización de la amplitud de la luminosidad se conoce como cuantificación del nivel de gris. Una imagen continua $f(x, y)$ se describe de forma aproximada por una serie de muestras igualmente espaciadas organizadas en forma de una matriz $N \times M$ donde cada elemento de la matriz es una cantidad discreta:

$$f(x, y) \approx \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, M-1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, M-1) \\ \vdots & \vdots & & \vdots \\ f(N-1, 0) & f(N-1, 1) & \cdots & f(N-1, M-1) \end{bmatrix} \quad (3.3)$$

El segundo miembro de la ecuación 3.3 representa lo que comúnmente se denomina una imagen digital. A cada elemento de la matriz se le puede llamar elemento de la imagen o pixel. De este modo, los términos de imagen y pixel se emplean en procesamiento de imágenes para indicar una imagen digital y sus elementos. Una imagen digital es una función bidimensional cuyas coordenadas y valores de amplitud son números enteros. En

el procesamiento digital de imágenes por conveniencia se hace que

$$N = 2^n, \quad M = 2^k, \quad G = 2^m \quad (3.4)$$

donde G indica el número de niveles de gris, N y M las dimensiones de la imagen en píxeles y con esto se hace que dichas cantidades sean potencias enteras de 2.

3.3. La transformada discreta de Fourier de una dimensión y su inversa

La Transformada de Fourier de una función discreta –de una variable– $f(x)$, para $x = 0, 1, 2, \dots, M - 1$, está dada por la ecuación

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M}, \quad \text{para } u = 0, 1, 2, \dots, M - 1. \quad (3.5)$$

La transformada discreta de Fourier o DFT¹ es la base para el desarrollo de las ideas y trabajo en este Capítulo. De manera similar, dado $F(u)$, podemos obtener la función original usando la DFT inversa :

$$f(x) = \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M}, \quad \text{para } x = 0, 1, 2, \dots, M - 1. \quad (3.6)$$

Así, la DFT y su inversa conforman un par de transformadas pues se obtiene $F(u)$ a partir de $f(x)$, y viceversa. Nótese que algunas veces, el factor $1/M$, que multiplica a la Transformada de Fourier, se coloca multiplicando a la inversa. Otras veces, ambas ecuaciones se multiplican por $1/\sqrt{M}$. La posición de ese término no importa; si se utilizan dos, la única restricción es que su producto sea $1/M$. Una propiedad importante de la DFT

¹Del inglés *Discrete Fourier Transform*

es que no debemos preocuparnos por la existencia de la transformada o la transformada inversa: Estas siempre existen cuando usamos valores finitos [Proakis, 1999]. Este es el caso del procesamiento de imágenes digitales.

La fórmula de Euler permite expresar las transformadas en función las transformadas de senos y cosenos y así conocer la frecuencia asociada de la señal. La fórmula de Euler está dada por

$$e^{j\theta} = \cos \theta + j \sin \theta. \quad (3.7)$$

Sustituyendo esta expresión en 3.5, y tomando en cuenta que $\cos(-\theta) = \cos \theta$, obtenemos

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) (\cos 2\pi ux/M - j \sin 2\pi ux/M), \quad (3.8)$$

para $u = 0, 1, 2, \dots, M - 1$. De aquí se observa que cada término de $F(u)$ se compone de la suma de todos los valores de $f(x)$ multiplicados por senos y cosenos. El dominio (valores de u) es apropiadamente llamado dominio de la frecuencia, porque u determina la frecuencia de los componentes de la transformada. Cada uno de los M términos de $F(u)$ es un componente de frecuencia de la transformada.

En general, de las ecuaciones 3.5 y 3.8, se ve que los componentes de la transformada de Fourier son números complejos. Por esta característica se puede expresar $F(u)$ en forma polar

$$F(u) = |F(u)| e^{-j\phi(u)}, \quad (3.9)$$

donde

$$|F(u)| = \sqrt{R^2(u) + I^2(u)} \quad (3.10)$$

es la magnitud o espectro de la transformada de Fourier, y

$$\phi(u) = \text{ang tan} \left(\frac{I(u)}{R(u)} \right) \quad (3.11)$$

es la fase o espectro de fase de la transformada. $R(u)$ y $I(u)$ son las partes real e imaginaria de $F(u)$, respectivamente.

En la DFT (ecuación 3.5), la función $f(x)$ para $x = 0, 1, 2, \dots, M - 1$, representa M muestras de una señal continua. Es importante tener en mente que esas muestras no necesariamente corresponden a valores enteros de x en el intervalo $[0, M - 1]$. Las muestras son equiespaciadas con un intervalo fijo Δx . Existe un punto x_o que denota el primer punto de la secuencia. El primer valor de la función muestreada es $f(x_o)$. La siguiente muestra se toma a Δx y su valor es $f(x_o + \Delta x)$. La k -ésima muestra es $f(x_o + k\Delta x)$, y la muestra final $f(x_o + (M - 1)\Delta x)$. Entonces, en el caso discreto, si se escribe $f(k)$, se entiende que es una forma breve de escribir $f(x_o + k\Delta x)$. Con esta notación, la función $f(x)$ de muestras es

$$f(x) \triangleq f(x_o + x\Delta x) , \quad (3.12)$$

cuando se usan variables discretas. La variable u tiene una interpretación similar, pero la secuencia siempre comienza en frecuencia cero y es $0, \Delta u, 2\Delta u, \dots, (M - 1)\Delta u$, y $F(u)$ se expresa como

$$F(u) \triangleq F(u\Delta u) ; \quad (3.13)$$

para $u = 0, 1, 2, \dots, M - 1$. Δx y Δu están relacionados inversamente por la expresión

$$\Delta u = \frac{1}{M\Delta x}. \quad (3.14)$$

3.4. La DFT de 2 dimensiones y su inversa

La extensión de la DFT de una a dos dimensiones es fácil de llevar a cabo. La DFT de una función $f(x, y)$ de tamaño $M \times N$ es

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)}. \quad (3.15)$$

Esta expresión se calcula para valores de $u = 0, 1, 2, \dots, M-1$ y para $v = 0, 1, 2, \dots, N-1$.

A partir de $F(u, v)$ se puede obtener $f(x, y)$ por medio de la transformada inversa, dada por la expresión

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M+vy/N)} \quad (3.16)$$

para $x = 0, 1, 2, \dots, M-1$ y $y = 0, 1, 2, \dots, N-1$. La magnitud y la fase son

$$|F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)} \quad (3.17)$$

$$\phi(u, v) = \text{ang} \tan \left(\frac{I(u, v)}{R(u, v)} \right) \quad (3.18)$$

donde $R(u, v)$ y $I(u, v)$ son las partes real e imaginaria de $F(u, v)$.

Para el procesamiento de imágenes es común multiplicar la función que representa la luminosidad de la imagen $f(x, y)$ por $(-1)^{x+y}$ antes de obtener la DFT. Debido a las propiedades de las exponenciales, se tiene que

$$\mathfrak{F} [f(x, y) (-1)^{x+y}] = F \left(u - \frac{M}{2}, v - \frac{N}{2} \right) \quad (3.19)$$

donde $\mathfrak{F} [\]$ denota la transformada de Fourier del argumento. Esta ecuación muestra que al multiplicar por $(-1)^{x+y}$ se produce un desplazamiento en la frecuencia, y el origen pasa de $(0, 0)$ a $(\frac{M}{2}, \frac{N}{2})$, que es el centro del área de $M \times N$ ocupada por $F(u, v)$. Del mismo modo

que en el caso unidimensional, se tienen las siguientes relaciones entre resolución espacial y de frecuencia

$$\Delta u = \frac{1}{M\Delta x} \quad (3.20)$$

$$\Delta v = \frac{1}{M\Delta y} \quad (3.21)$$

donde Δx y Δy son los intervalos entre muestras en x y y y Δu y Δv son los intervalos de frecuencia en el dominio de la frecuencia.

3.5. Filtrado en el dominio de la frecuencia

El dominio de la frecuencia no es más que el espacio definido por los valores de la transformada de Fourier y sus variables u y v , y su comprensión resulta muy útil en el procesamiento de imágenes.

3.5.1. Algunas propiedades básicas del dominio de la frecuencia

De la ecuación 3.15 se observa que cada término de $F(u, v)$ es una combinación lineal de todos los valores de $f(x, y)$ modificados por valores de la exponencial $e^{-j2\pi(ux/M+vy/N)}$. A partir de $F(u, v)$ algunas deducciones son factibles respecto de la relación entre los componentes de la transformada de Fourier y las características espaciales de una imagen. Como la frecuencia está relacionada con la tasa de cambio, no es difícil asociar frecuencias con variaciones de intensidad en una imagen. La componente de menor frecuencia ($u = v = 0$) en el origen corresponde al nivel de gris promedio –espacial– de la imagen. Cerca del origen, las frecuencias bajas corresponden a los componentes de cambio lento de la tonalidad de grises de la imagen. En una imagen de una recámara, por ejemplo, podrían corresponder a

variaciones suaves de nivel de gris en las paredes y piso. En la parte alejada del origen, la frecuencias altas corresponden a cambios bruscos de nivel en la imagen, como pueden ser los bordes de los objetos o el ruido.

3.5.2. Bases del filtrado en el dominio de la frecuencia

La operación de filtrado de una imagen es conveniente para depurar las contribuciones de ruido presentes. Para ello hay que realizar los siguientes pasos, que se ilustran en la Figura 3.19:

1. Multiplicar la imagen de entrada $f(x, y)$ por $(-1)^{x+y}$ para desplazar el origen de la transformada al centro.
2. Calcular la DFT: $F(u, v)$.
3. Multiplicar $F(u, v)$ por una función de filtrado $H(u, v)$.
4. Calcular la DFT inversa.
5. Obtener la parte real de la inversa.
6. Multiplicar el resultado por $(-1)^{x+y}$ para obtener la imagen de salida.

La Figura 3.1 muestra los pasos básicos para el filtrado. A la función $H(u, v)$ se le denomina el filtro, o función de transferencia de filtrado, porque atenúa algunas frecuencias –particularmente las altas frecuencias– y dejando otras casi intactas.

Para la imagen filtrada, la transformada de Fourier de la imagen de salida está dada por

$$G(u, v) = H(u, v) F(u, v). \quad (3.22)$$

La multiplicación de H y F es de funciones de dos dimensiones y esta definida como la

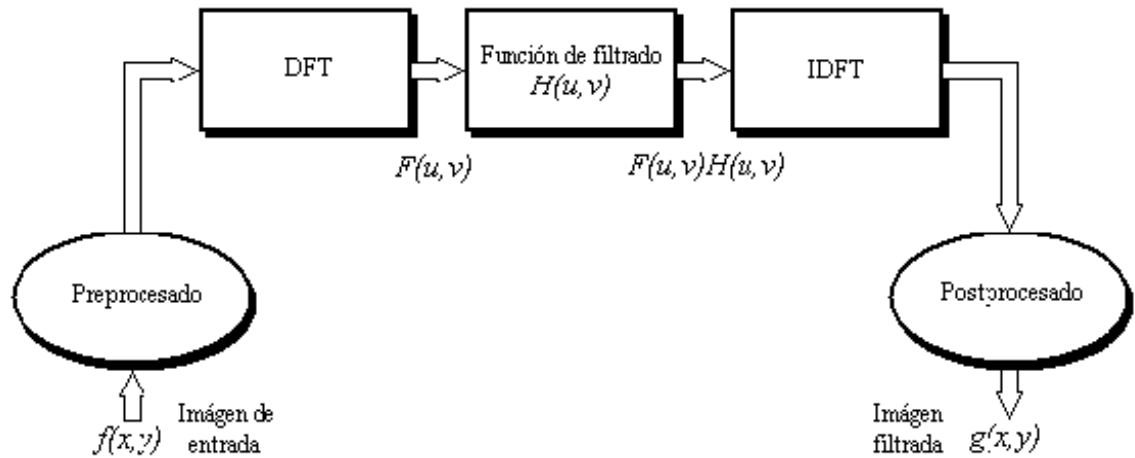


Figura 3.1: Pasos del filtrado en el dominio de la frecuencia.

multiplicación elemento a elemento. Si las matrices H y F son de la forma

$$H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1(N-1)} \\ h_{21} & h_{22} & \cdots & h_{2(N-1)} \\ \vdots & \vdots & & \vdots \\ h_{(M-1)1} & h_{(M-1)2} & \cdots & h_{(M-1)(N-1)} \end{bmatrix} \quad (3.23)$$

$$F = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1(N-1)} \\ f_{21} & f_{22} & \cdots & f_{2(N-1)} \\ \vdots & \vdots & & \vdots \\ f_{(M-1)1} & f_{(M-1)2} & \cdots & f_{(M-1)(N-1)} \end{bmatrix} \quad (3.24)$$

Cada elemento h_{ij} se multiplica con el correspondiente f_{ij} para obtener g_{ij} . En general, los coeficientes de F son complejos, pero los filtros de procesamiento de imágenes son típicamente reales. Estos filtros no cambian la fase de F , como se describe en el capítulo 2 sobre la OTF.

Capítulo 4

Detección de bordes

Uno de los procesos clave en el procesamiento de imágenes de gotas embebidas en un segundo fluido es la detección de los bordes para la identificación precisa de la interfaz entre los líquidos. En general para la detección de bordes en imágenes se han desarrollado diferentes algoritmos de detección, los cuales difieren unos de otros en aspectos tales como el costo computacional, su eficiencia y la facilidad para su implementación. Los Operadores de Gradiente como los de Sobel, Robert y Prewitt representan la primera generación de algoritmos de detección de bordes. La desventaja de estos algoritmos es su limitada capacidad para detectar bordes débiles y su pobre eficiencia bajo la presencia de ruido. En contraparte, el Operador de Marr –conocido como Laplaciano del Gaussiano (LoG)– es un algoritmo comúnmente utilizado que se basa en los cruces por cero después de aplicar el operador; su desventaja es que introduce algunos bordes falsos. El operador de Canny es uno de los algoritmos más ampliamente usados debido a su eficiencia[Canny, 1986]. De manera similar a Marr, Canny utiliza un proceso de suavizado con una máscara gaussiana.

El algoritmo calcula la primera derivada del borde en cada pixel. La calidad de la detección depende principalmente de los parámetros del operador utilizados.

Un algoritmo de desarrollo reciente es el algoritmo de *máscara de diferencia absoluta*, el cual siendo similar a Canny tiene un bajo costo computacional debido a que evita operaciones complejas como multiplicación y división[Alzarahni, 1997]. Este algoritmo se describe detalladamente más adelante. Este algoritmo aplica una máscara semigaussiana para reducir el efecto del ruido, calcula la pendiente del borde y su dirección en cada pixel y finalmente adelgaza y enlaza los bordes. En la Figura 4.1 se muestran las salidas correspondientes a la fotografía de una gota para algunos de los algoritmos de detección de bordes más conocidos. En dicha Figura se puede observar la alta sensibilidad al ruido de LoG y Sobel, siendo poco aptos para las imágenes con las que se desea trabajar. El desempeño de Canny es adecuado para los objetivos de este trabajo, sin embargo con Alzaharani se obtienen resultados similares y con menor costo de cómputo.

Un borde es un conjunto de pixeles conectados en la frontera entre dos regiones de una imagen con propiedades de nivel de gris distintas. La mayoría de las técnicas de detección de bordes se basan en el cálculo de un operador local de derivación. La Figura 4.2 ilustra este concepto. La Figura 4.2(a) muestra una imagen de una banda clara sobre un fondo oscuro, el perfil del nivel de gris de una línea de exploración horizontal y la primera y segunda derivada del perfil. Se observa en el perfil que un borde está modelado como un cambio suave, no brusco, del nivel de gris. Este modelo refleja el hecho de que los bordes de las imágenes digitales están en general ligeramente suavizados a causa del muestreo.

La Figura 4.2(a) muestra que la primera derivada del perfil de nivel de gris es

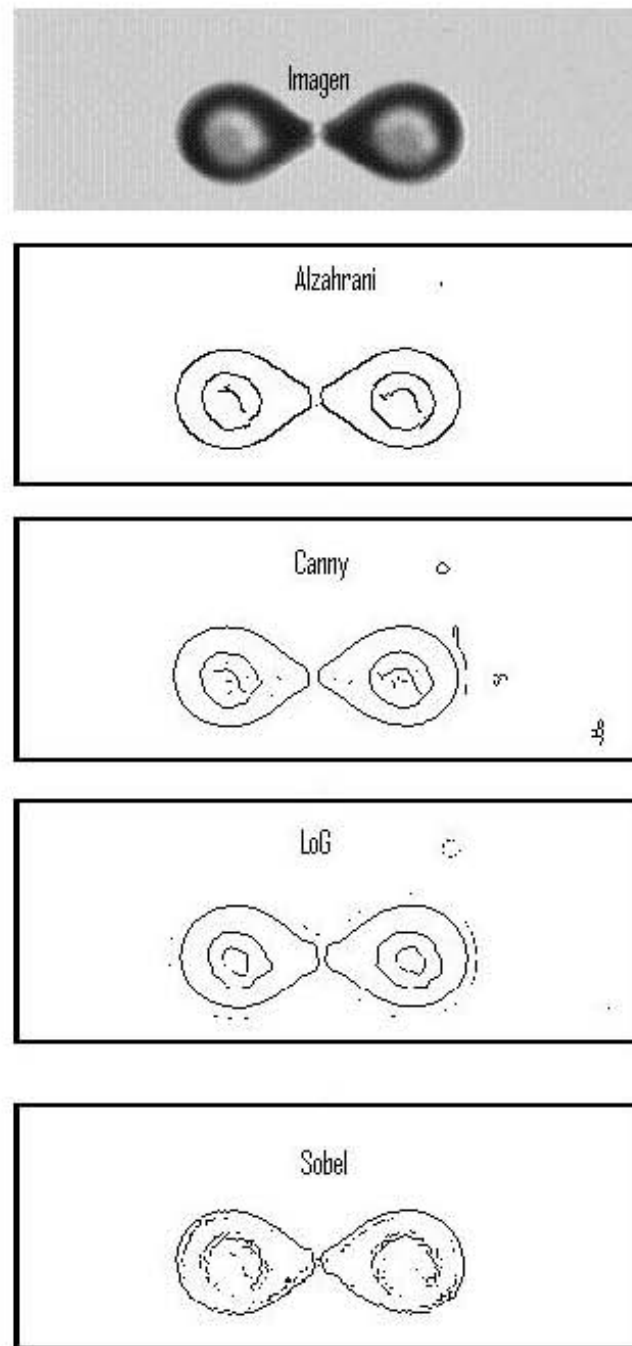


Figura 4.1: Detección de bordes en una imagen utilizando diferentes algoritmos.

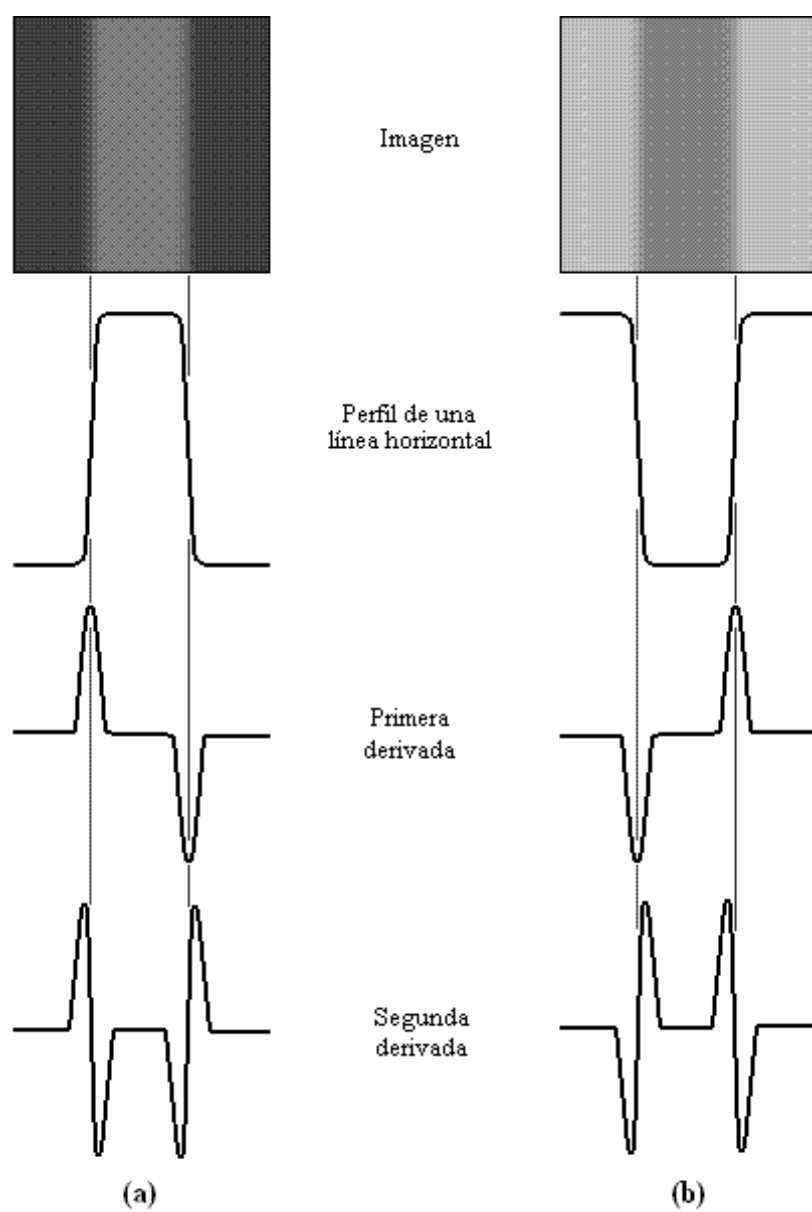


Figura 4.2: Detección de bordes por operadores de derivación. Obsérvese que la segunda derivada tiene un paso por cero en la posición de cada borde.

positiva en el borde de entrada de la transición, negativa en el borde de salida y cero en las zonas de nivel de gris constante. La segunda derivada es positiva en la parte de la transición asociada al lado oscuro del borde, negativa en la parte de la transición asociada al lado claro y cero en las zonas de nivel de gris constante. La primera derivada se puede usar para detectar la presencia de un borde y la segunda para localizar un borde cuando tiene poca pendiente debido al cruce por cero en la parte media de la transición de nivel de gris.

4.1. Operadores gradiente

El gradiente de una imagen $f(x, y)$ en la posición (x, y) es el vector

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (4.1)$$

El vector gradiente de un campo escalar –como las imágenes– indica la dirección de la máxima variación de f en (x, y) . Un elemento importante en la detección de bordes es la función módulo de este vector y se denota con $\|\nabla f\|$ donde

$$\nabla f = \text{mag}(\nabla \mathbf{f}) = \sqrt{G_x^2 + G_y^2} \quad (4.2)$$

Esta cantidad es igual a la máxima variación de $f(x, y)$ por unidad de distancia en la dirección de $\nabla \mathbf{f}$. Una buena aproximación al gradiente es

$$\nabla f = |G_x| + |G_y| \quad (4.3)$$

con la ventaja de que es menos costoso de calcular numéricamente. De las Ecuaciones 4.1 y 4.2 se deduce que el cálculo del gradiente de una imagen requiere conocer las derivadas parciales en cada posición de pixel.

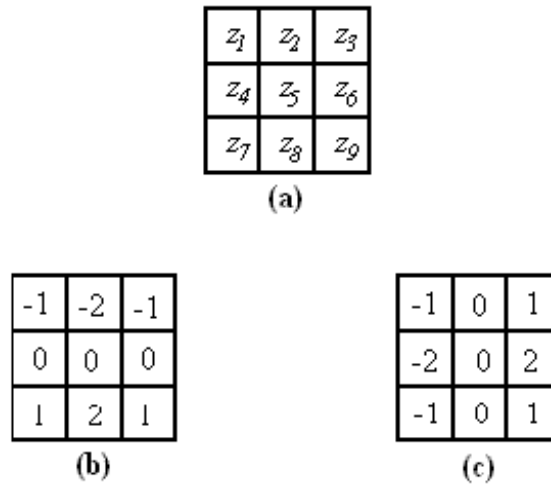


Figura 4.3: (a) Región imagen 3×3 ; (b) máscara para calcular G_x en el punto central de la región 3×3 ; (c) máscara para calcular G_y en ese punto. A estas máscaras se les denomina operadores de Sobel.

El cálculo de las derivadas parciales se puede implementar en forma digital. Una forma es mediante el uso de los Operadores de Sobel, con la ventaja de proporcionar además un efecto de suavizado. Como las derivadas realzan el ruido, este efecto de suavizado es una buena característica del Operador de Sobel. En la Figura 4.3, las derivadas basadas en las máscaras de Sobel son:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad (4.4)$$

y

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad (4.5)$$

donde las z_i corresponden con los niveles de gris de los pixeles pertenecientes a las máscaras en cualquier posición de una imagen. Entonces se calcula el gradiente en la ubicación del centro de las máscaras utilizando las Ecuaciones 4.2 y 4.3 que proporcionan un valor del gradiente. Para obtener el valor siguiente, se desplazan las máscaras a la posición del pixel

siguiente y se repite el proceso. Una vez que se ha cubierto todos los pixeles, el resultado es una imagen-gradiente del mismo tamaño que la original. En los extremos de la imagen se utiliza el entorno parcial apropiado.

4.2. Laplaciano

El laplaciano de una función bidimensional $f(x, y)$ es una derivada de segundo orden definida por

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (4.6)$$

Como en el caso del gradiente, la Ecuación 4.6 puede aplicarse digitalmente a imágenes de varias maneras. Para una subregión 3×3 de la imagen, la forma más frecuente es:

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8) \quad (4.7)$$

donde las z_i son las posiciones de los pixeles vecinos definidas previamente. El laplaciano rara vez se utiliza en la práctica porque por ser derivada de segundo orden es muy sensible al ruido. Además produce bordes dobles (figura 4.2). El laplaciano juega un papel secundario; sólo se utiliza para determinar si un pixel está en la parte clara u oscura de un borde.

En contraparte, la gran utilidad del laplaciano radica en su habilidad para ubicar con precisión los bordes utilizando su propiedad de paso por cero. Este concepto está basado en la convolución de una imagen con el laplaciano de una función gaussiana bidimensional de la forma:

$$h(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (4.8)$$

0	-1	0
-1	4	-1
0	-1	0

Figura 4.4: Máscara para calcular el laplaciano

donde σ es la desviación estándar. Si $r^2 = x^2 + y^2$, de la ecuación 4.6 el laplaciano de h es:

$$\nabla^2 h = \left(\frac{r^2 - \sigma^2}{\sigma^4} \right) \exp \left(-\frac{r^2}{2\sigma^2} \right) \quad (4.9)$$

La utilidad de esta ecuación radica en sus pasos por cero.

La detección de bordes por operadores gradiente es efectiva para los casos de imágenes con variaciones de intensidad claramente definidas y bajo ruido. El uso de los pasos por cero ofrece una alternativa en los casos con mucho ruido. Además, el paso por cero es confiable en la localización de bordes. En contraparte, el paso por cero es más complejo e invariablemente los algoritmos son más lentos.

4.3. Máscara de diferencia absoluta

Este algoritmo consta de tres etapas[Alzarahni, 1997].

4.3.1. Máscara semigaussiana

Para atenuar las componentes de alta frecuencia en la imagen –que corresponden al ruido– sin eliminar los bordes de interés se utiliza una máscara de 5×5 con perfil semigaussiano, lo que da más peso a los píxeles más cercanos al centro de la máscara, esta

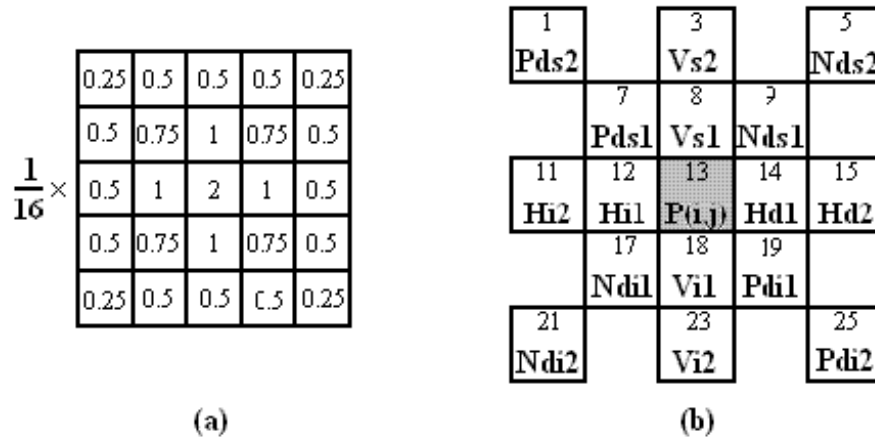


Figura 4.5: (a) Máscara de suavizado; (b) Máscara de diferencia absoluta.

máscara se muestra en la Figura 4.5(a). Las distancias entre píxeles se miden de centro a centro y el peso de cada elemento de la máscara es inversamente proporcional a la distancia al píxel central. Los valores se redondean a números enteros para simplificar el algoritmo.

4.3.2. Cálculo de pendiente y dirección del borde

Para calcular la pendiente del borde se utiliza la máscara de diferencia absoluta que se muestra en la Figura 4.5(b). En este caso, se toma la imagen filtrada de la etapa anterior y se obtiene para cada píxel la pendiente del borde en cada una de las cuatro direcciones: positiva diagonal (Pd), vertical (V), negativa diagonal (Nd) y horizontal (H). Se toma la pendiente máxima, ésta es la pendiente en ese punto (máxima diferencia absoluta). También se obtiene la dirección en la que se encuentra la mínima diferencia absoluta porque representa la dirección del borde.

Los pasos para realizar el procedimiento comentado arriba son:

1. Cálculo de diferencias absolutas para todos los pixeles

$$\begin{aligned}
 V &= |Vs_2 + Vs_1 - Vi_1 - Vi_2| & H &= |Hd_2 + Hd_1 - Hi_1 - Hi_2| \\
 Pd &= |Pds_2 + Pds_1 - Pdi_1 - Pdi_2| & Nd &= |Nds_2 + Nds_1 - Ndi_1 - Ndi_2|
 \end{aligned}
 \tag{4.10}$$

2. Determinación de la pendiente y dirección

$$m_b = \frac{\text{máx}(V, H, Pd, Nd)}{2} \tag{4.11}$$

$$dir_b = dir(\text{mín}(V, H, Pd, Nd)) \tag{4.12}$$

Una vez completado el procedimiento tenemos dos matrices del tamaño de la imagen, la primera con las pendientes y la segunda con las direcciones.

3 Adelgazamiento y enlazado de bordes

La pendiente de un borde varía en forma proporcional al cambio en el nivel de gris. Existen diversos valores de pendiente a lo largo de un borde, con un máximo local. Cada uno de esos valores tiene una dirección de borde relacionada. La pendiente y la dirección se usan para adelgazar y enlazar por medio de los pasos que se muestran en la Figura 4.6. La dirección se usa para decidir la comparación que se va a llevar a cabo. El umbral (threshold denotado como "thres") se aplica a la salida y depende de los bordes que se deseen.

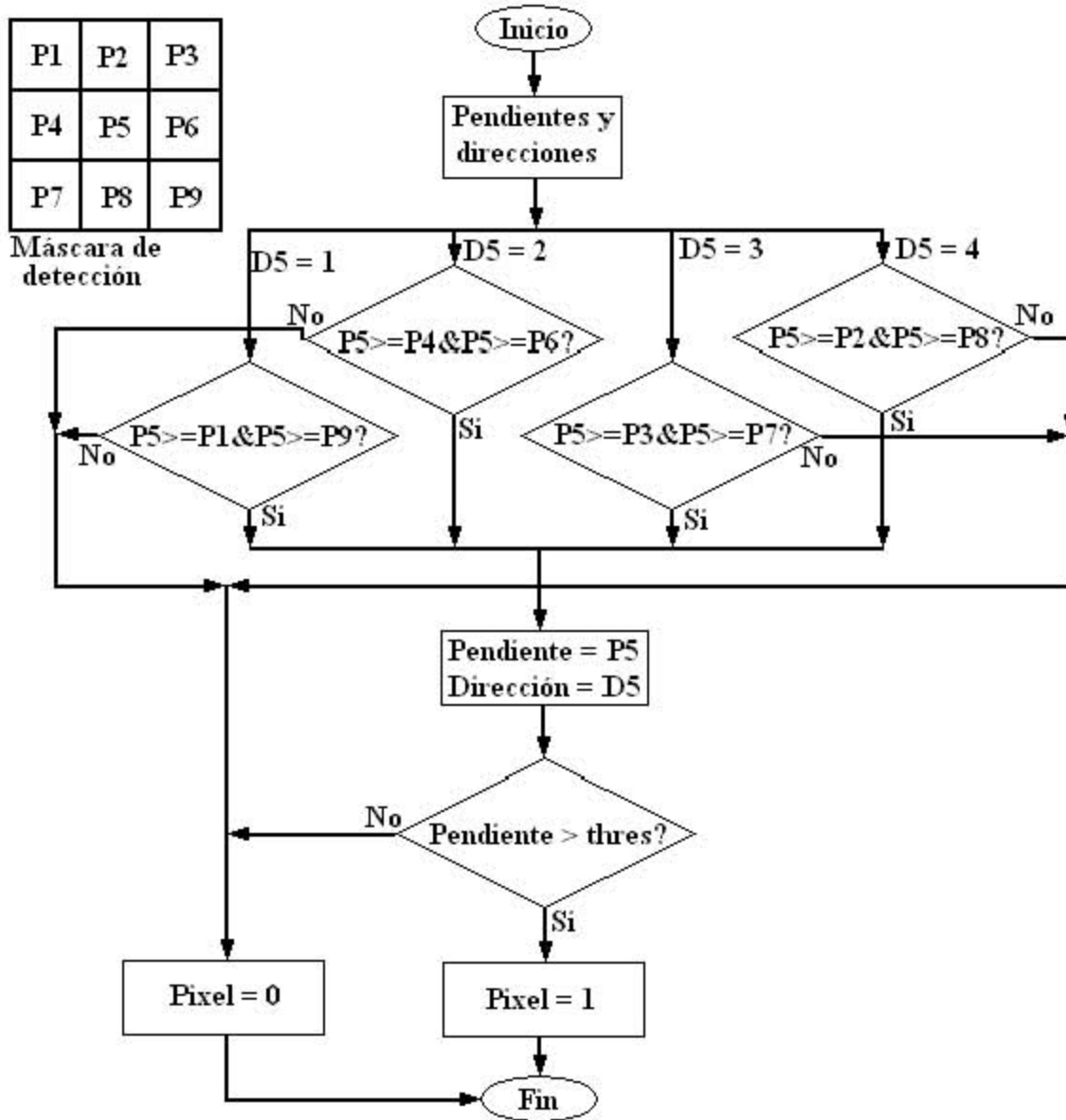


Figura 4.6: Diagrama de flujo de la parte de adelgazamiento y enlazado.

Parte II

Experimentación

Capítulo 5

Implementación de algoritmos

Para la parte experimental se utilizó el arreglo que se describe en el Capítulo 1 (página 3). Optamos por utilizar el software de Matlab© por las ventajas que ofrece su versatilidad para el procesamiento matemático, la visualización y la programación. Se desarrollaron programas que cumplen con las finalidades de las diferentes etapas de la experimentación: (a) adquisición de video, (b) obtención de imágenes a partir de video, (c) filtrado en el dominio de la frecuencia utilizando las características de la MTF, (d) detección de bordes, (e) interpretación de las imágenes a partir de los bordes y (f) obtención de parámetros espaciales y temporales. Estos programas llevan a la práctica los diferentes conceptos teóricos descritos en la parte I (a partir de la página 15) de este Trabajo y permiten llevar a cabo consideraciones sobre los métodos usados, sentando un precedente para posteriores desarrollos.

5.1. Adquisición de video

Esta etapa se llevó a cabo en el laboratorio. Se desarrollaron dos programas, uno con la finalidad de configurar la cámara y controlar la captura y otro para extraer la información y almacenarla en formato AVI. La cámara utilizada es una Sony modelo XCD-X710 (CCD monocromático). La cámara cuenta con un puerto de comunicación del tipo IIDC 1394¹ por lo que fue necesario instalar y configurar el driver CMU DCAM².

El primer programa permite definir la resolución de la cámara de 480×640 pixels así como el número total de tramas que se van a obtener, este último dependiendo esencialmente de la duración del video. El formato de video por default en el controlador define una velocidad de captura de 30 tramas por segundo, y codificación del color en 8 niveles de gris. Con esta configuración se crea una estructura de tipo *video input*. Una vez configurados los parámetros realiza la captura y una vez concluida, la información obtenida queda guardada en la misma estructura de tipo *video input* en que se guardó la información (ver en Apéndice A:*video.m*).

El segundo programa extrae la información trama por trama de la estructura *video input*, aplica detección de bordes a cada una de las tramas, concatena todas las tramas a una estructura de tipo *movie* y finalmente guarda esas estructuras en archivos de video en formato *.avi*. Lo que se obtiene después de correr este programa son dos archivos de video: uno con el video recién capturado y otro con la detección de bordes (ver en Apéndice A:*imagen.m*).

¹Instrumentation and Industrial Digital Camera que usa el estándar 1394 de IEEE también conocido como FireWire.

²Driver desarrollado por Carnegie Melon University Digital CAMera para el Toolbox de adquisición de imágenes de Matlab.

Estas herramientas que desarrollamos fueron de mucha utilidad en el laboratorio porque permitieron guardar las capturas para su posterior procesamiento y seleccionar, gracias a la detección de bordes, los videos adecuados para las etapas posteriores.

5.2. Obtención de imágenes

Esta etapa y las posteriores se llevaron a cabo en equipo de cómputo complementario al del control del experimento, considerando el procesamiento de la información obtenida previamente de manera experimental. El programa desarrollado para esta etapa obtiene fotografías en formato *.bmp* para cada una de las tramas de los videos en Formato *.avi* de la etapa anterior y se recortan para conservar solo el área útil. Asimismo genera nombres de archivo automáticamente para todas las fotografías, las cuales serán analizadas en las etapas siguientes.

El programa lee la información contenida en el archivo *.avi* y la guarda en una estructura de tipo *movie*. También obtiene de la configuración del archivo algunas propiedades útiles como el número de tramas y la resolución. Conociendo la resolución y después de haber observado el video, se puede elegir solo una región de la imagen para su análisis. Esto disminuye la cantidad de información que se va a procesar y agiliza la etapa de procesamiento. Después se inicia un *loop* en el cual una a una, todas las tramas de la estructura de tipo *movie* se guardan en una estructura de tipo *frame*, se convierten a una estructura de tipo *image*, se recortan para conservar solo el área útil, y se guardan en un archivo en formato *.bmp* para el cual se genera un nombre de la forma **nombreamchivoXXXX.bmp** donde **XXXX** depende del número de tramas y puede variar desde 0000 hasta 9999 (ver en Apéndice

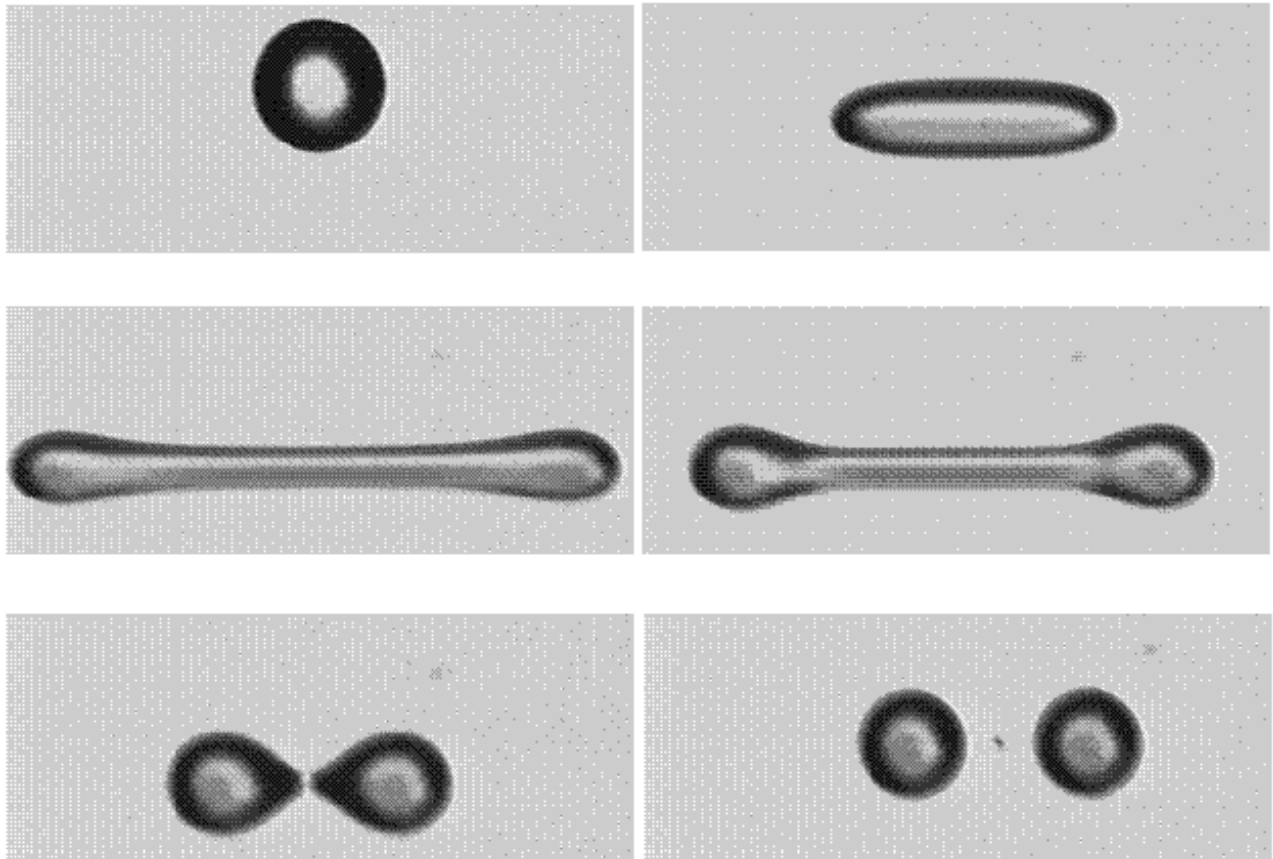


Figura 5.1: Fotografías obtenidas a partir de un video que muestran diferentes etapas de la ruptura de una gota.

A:fotos.m).

En la Figura 5.1 se muestran fotografías correspondientes a tramas en diferentes momentos de un video.

5.3. Filtrado digital (emulación de una lente)

Para realizar el filtrado digital se propusieron diferentes perfiles de MTF, los cuales pretenden ser una aproximación a los obtenidos experimentalmente por los fabricantes de lentes. Esta simplificación resulta útil porque permite conocer previamente el desempeño de la lente cuando el fabricante proporciona la MTF o se obtiene experimentalmente en el laboratorio. Los perfiles propuestos corresponden a las funciones recta, campana de Gauss, exponencial inversa y elíptica, mismos que se muestran en la Figura 5.2. Para todos ellos se programaron las respectivas funciones de filtrado, en las cuales se puede modificar el ancho de banda.

El programa de filtrado lee la imagen de un archivo en formato *.bmp* y la guarda en una estructura de tipo *image*. Después la convierte a escala de grises y la guarda en una matriz de *doubles*. Se multiplica término por término por una matriz $(-1)^{x+y}$ de las mismas dimensiones para desplazar el origen de la transformada al centro, tal como se explicó en la sección 3.5.2 (página 58). Se obtiene la transformada de Fourier en dos dimensiones. El resultado es una matriz de números complejos a la cual se le elimina la fase. Este procedimiento es válido porque al tratarse de una imagen generada con luz no coherente –no de un holograma–, la fase no toma parte activa. Después se genera una matriz con el perfil, tal como el que se observa en la Figura 5.2, incisos b, d, f o h. El filtrado propiamente dicho se lleva a cabo al multiplicar término por término la matriz transformada con la matriz del perfil. Como siguiente paso se obtiene la transformada inversa de Fourier de dos dimensiones, se elimina nuevamente la fase y se convierten los elementos de la matriz a

enteros de 8 bits³. Finalmente se guarda la imagen resultante ya filtrada en un archivo en formato *.bmp*, con el cual es posible determinar hasta qué grado la lente utilizada se apega a las necesidades del experimento o hasta que grado la imagen obtenida con una lente en particular es útil considerando las frecuencias que fueron atenuadas por dicha lente (Ver *filtrado.m* en apéndice).

En la Figura 5.3 se muestran las imágenes que se obtienen al filtrar una imagen con diferentes anchos de banda usando el perfil gaussiano.

5.4. Detección de bordes

Para la detección de bordes, después de experimentar con diferentes algoritmos, se tomó la decisión de usar el algoritmo de Máscara de Diferencia Absoluta por tener la ventaja de que su tiempo de procesamiento es considerablemente más breve, y cuyos resultados son muy similares a los de Canny [Alzarahni, 1997] . Y para las características del objeto de estudio de este trabajo los resultados son igualmente útiles.

La función que realiza la detección de bordes recibe una matriz de enteros de 8 bits con la imagen y entrega dos matrices de las mismas dimensiones: una con la detección de bordes en escala de grises a 8 bits y otra con la detección de bordes en dos colores, negro para ausencia de borde y blanco para presencia de borde. La función convierte los elementos de la matriz de entrada a *doubles*, obtiene las dimensiones de la entrada y aplica el algoritmo de Máscara de Diferencia Absoluta explicado a detalle previamente en la sección 4.3 (página 67) de este trabajo. Posteriormente convierte a enteros de 8 bits las matrices resultantes y

³`uint8` en Matlab.

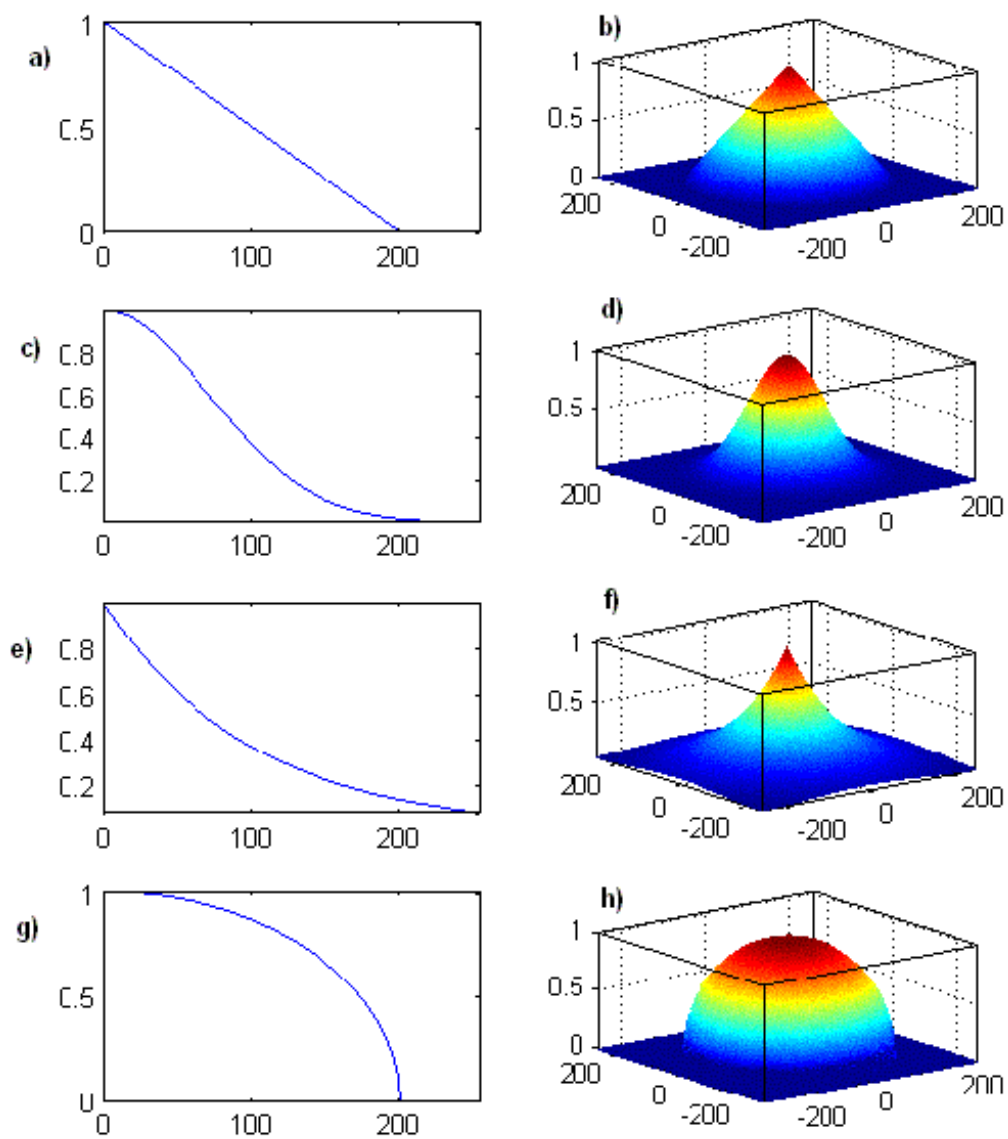


Figura 5.2: Diferentes perfiles de MTF y sus correspondientes filtros: a) y b) MTF y filtro correspondientes al perfil recto. c) y d) Perfil en forma de campana de Gauss. e) y f) Perfil en forma de exponencial inversa. g) y h) Perfil en forma de elipse.

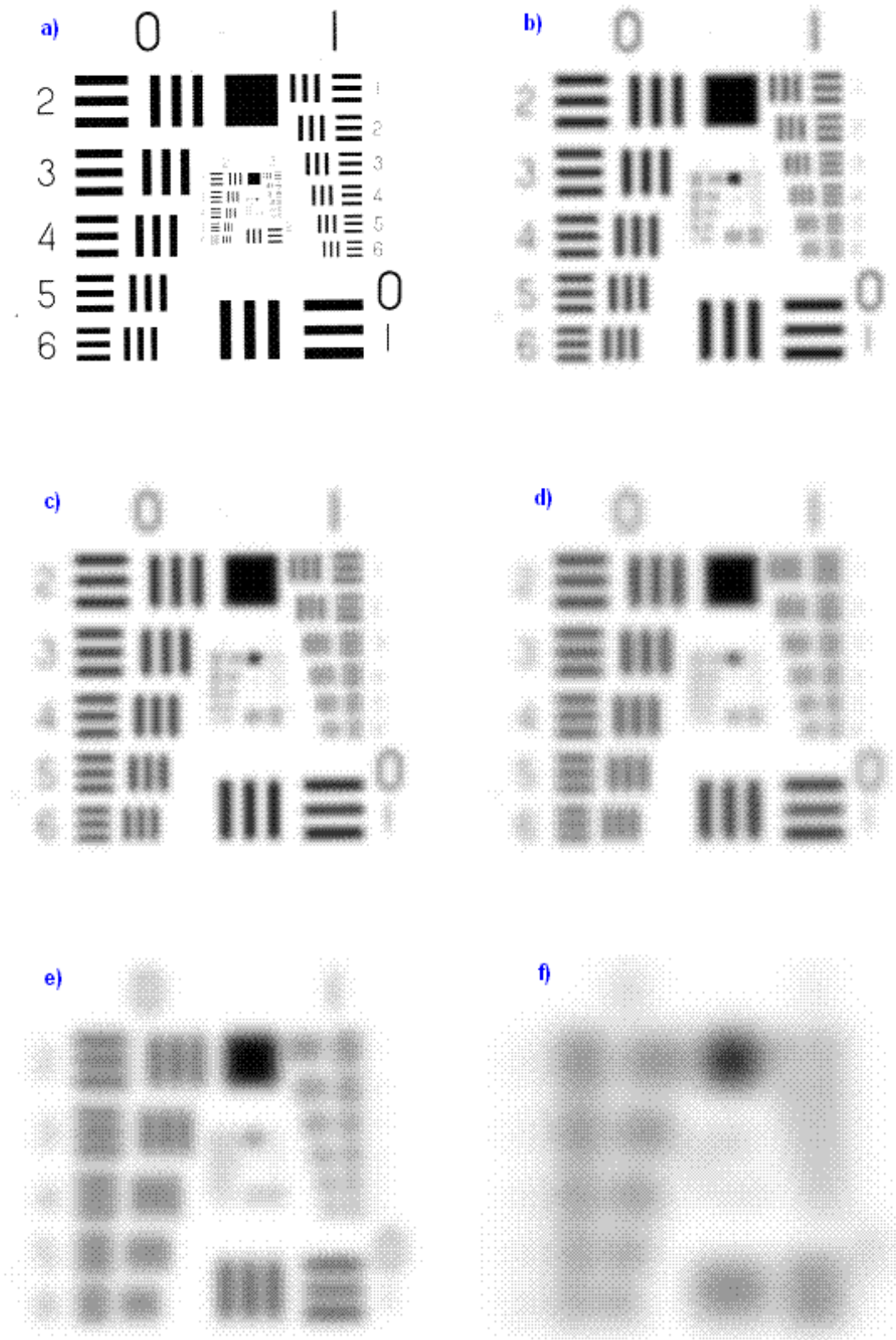


Figura 5.3: Filtrado de una plantilla para calibración de lentes usando el algoritmo para perfil gaussiano. a) imagen original. b) hasta f) anchos de banda de 25, 20, 15, 10 y $5 \times \frac{1}{256}$ ciclos/pix respectivamente.

las entrega como la salida de la función (ver en Apéndice A:alzahrani.m).

En la Figura 5.6 se muestran los resultados después de aplicar la detección de bordes a las imágenes correspondientes a la Figura 5.1. La Figura 5.1 corresponde con el tipo de imágenes que se procesarán con los algoritmos desarrollados en esta tesis. En dicha figura, una gota que es sometida a un flujo externo, se rompe para formar dos gotas.

Ahora presentamos el procesamiento completo de un video tomado de un molino de 4 rodillos. En dicho video, dos gotas sometidas al flujo generado por el molino y ubicadas en la región de estancamiento, se acercan lentamente y bajo condiciones hidrodinámicas propicias éstas se fusionan en una sola. Estos efectos son tema de estudio actualmente.

Como se explicó en la Sección 5.2 se descompone el video en imágenes y entonces se procesan las imágenes de manera individual. En la Figura 5.4 se muestran imágenes del video a diferentes tiempos. A continuación se aplica el algoritmo de detección de bordes (ver en Apéndice A:Alzahrani.m). En la Figura 5.5 se muestran los resultados de la detección de bordes correspondientes a las imágenes de la Figura 5.4. Haciendo una comparación visual podemos distinguir los bordes correspondientes a los contornos de las gotas, pero también encontramos bordes que preferiríamos omitir con el fin de hacer más simple el procesamiento posterior que consiste en identificar las gotas. También es preferible que los bordes sean cerrados con el fin de delimitar regiones para identificar objetos. Con el fin de contrarrestar estas limitantes aplicamos más procesamiento a las imágenes después de haber aplicado la detección de bordes.

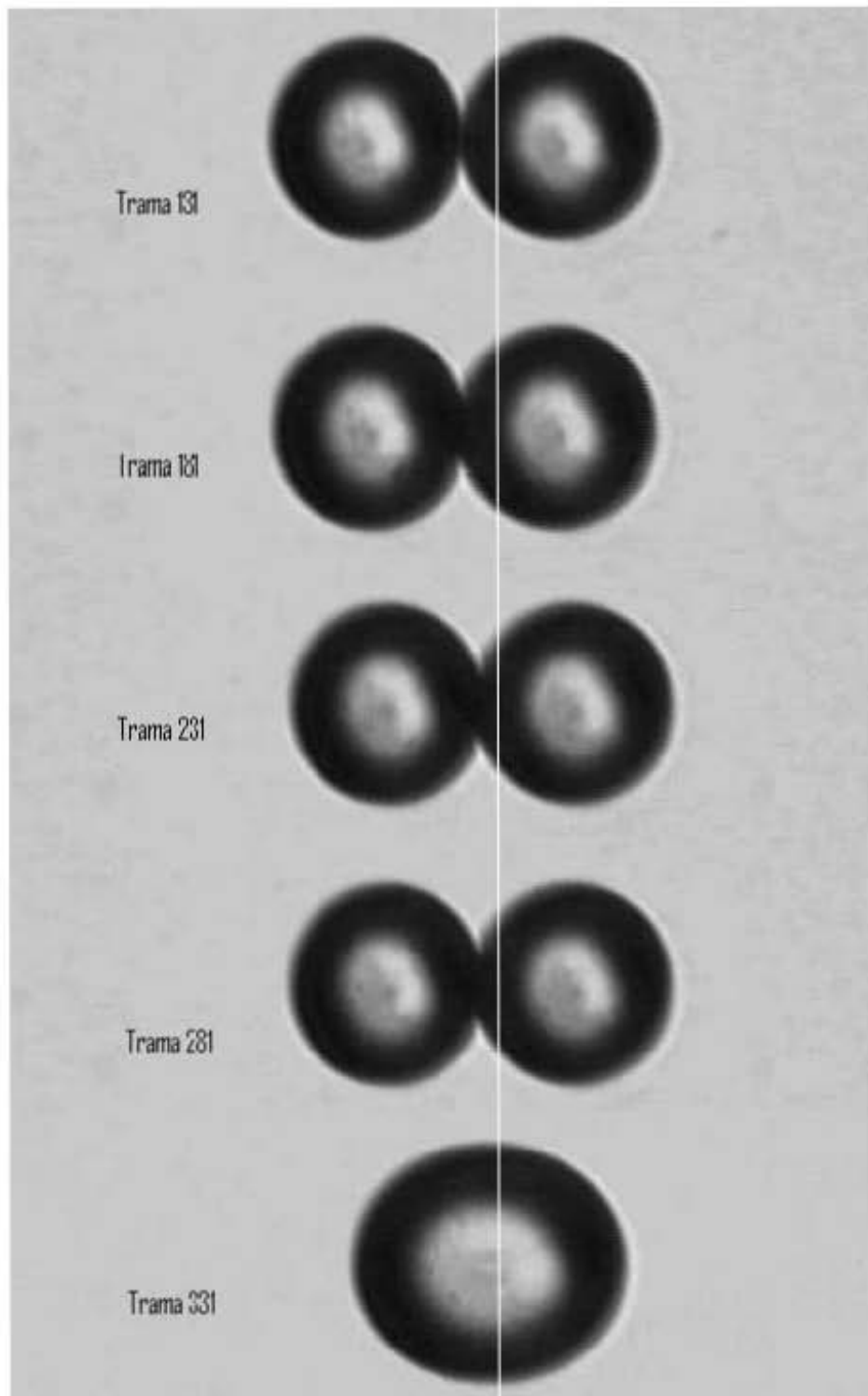


Figura 5.4: Fotografías tomadas en diferentes instantes de dos gotas sometidas a un flujo generado por un molino de 4 rodillos.

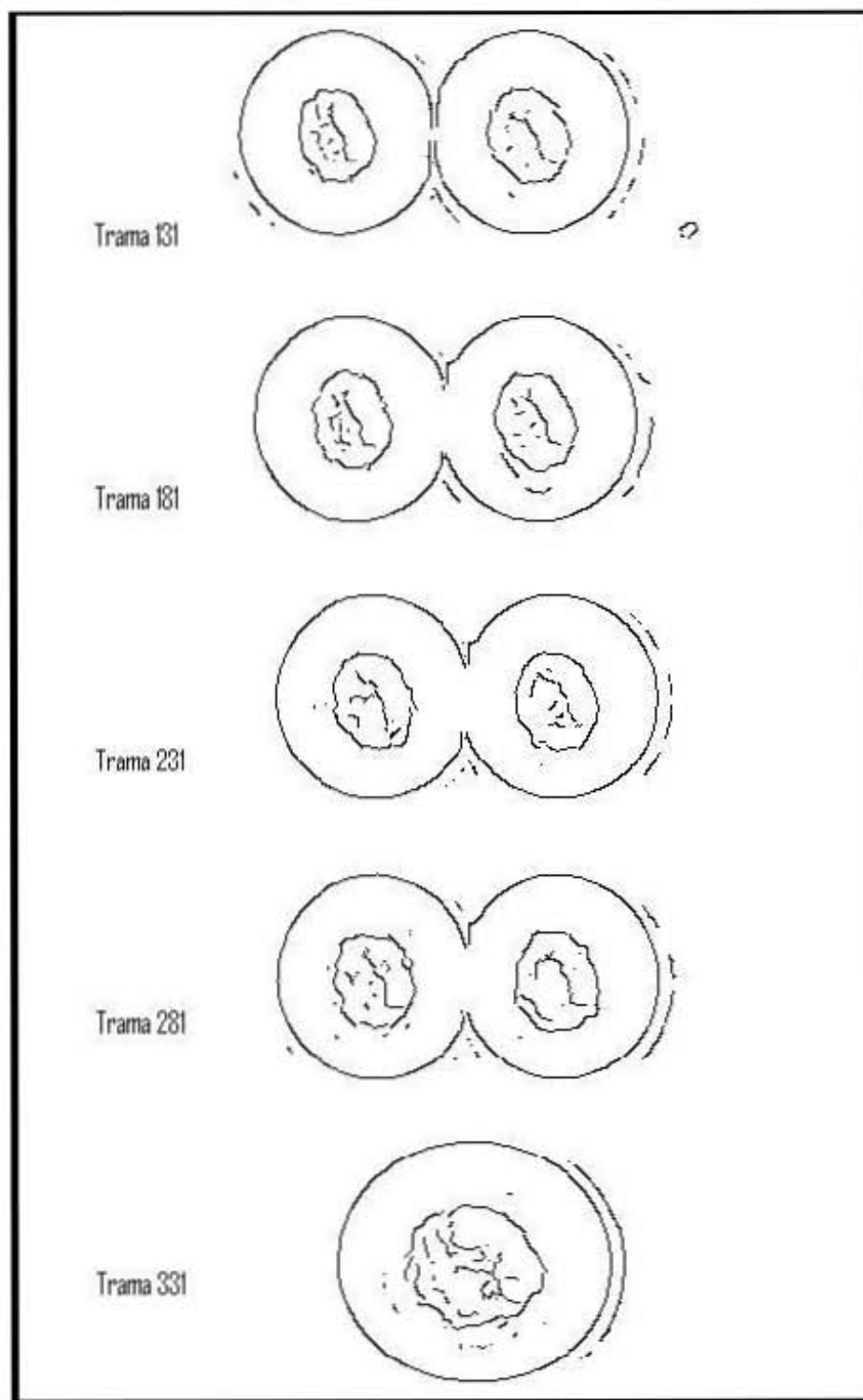


Figura 5.5: Resultados obtenidos después de aplicar la detección de bordes a las fotografías anteriores.

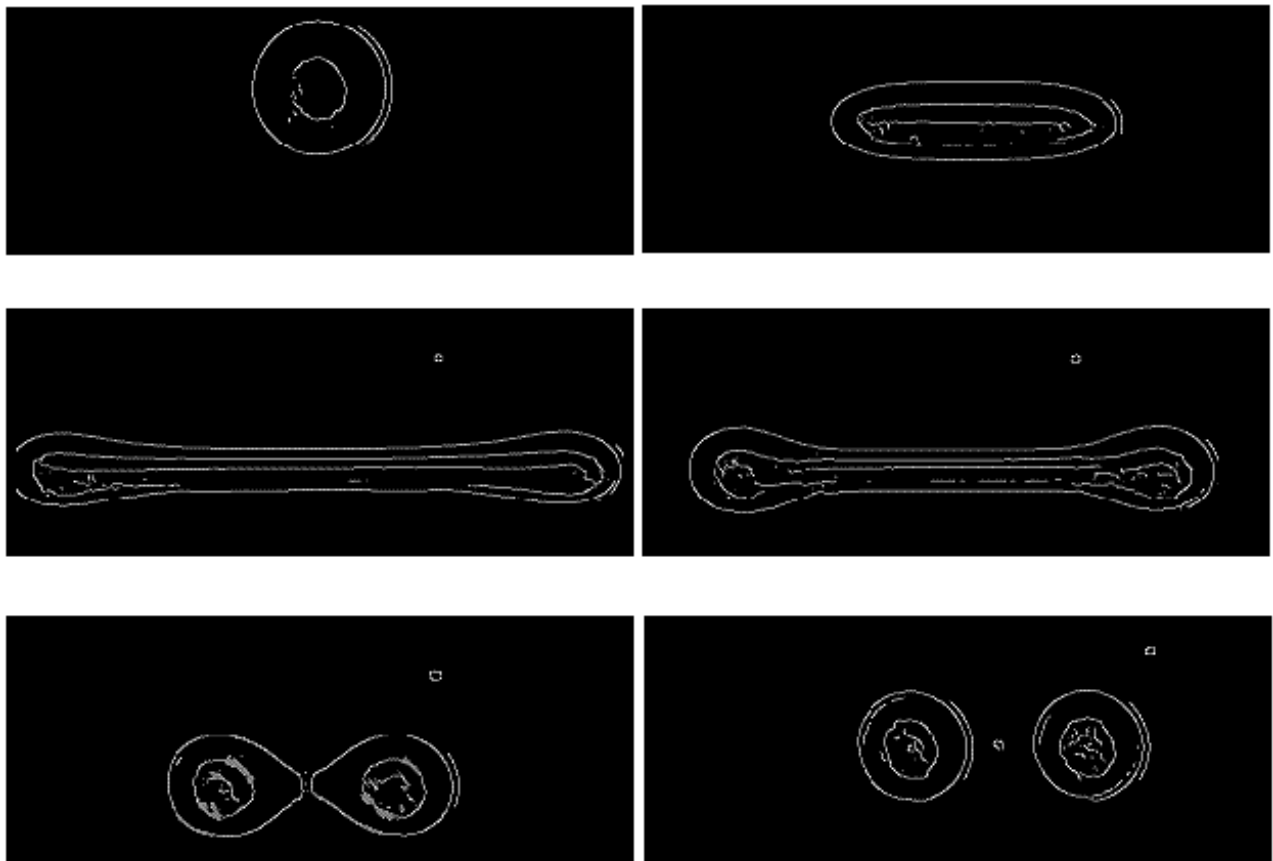


Figura 5.6: Resultados de procesar las fotografías de la figura 5.1 usando el algoritmo de Máscara de Diferencia Absoluta.

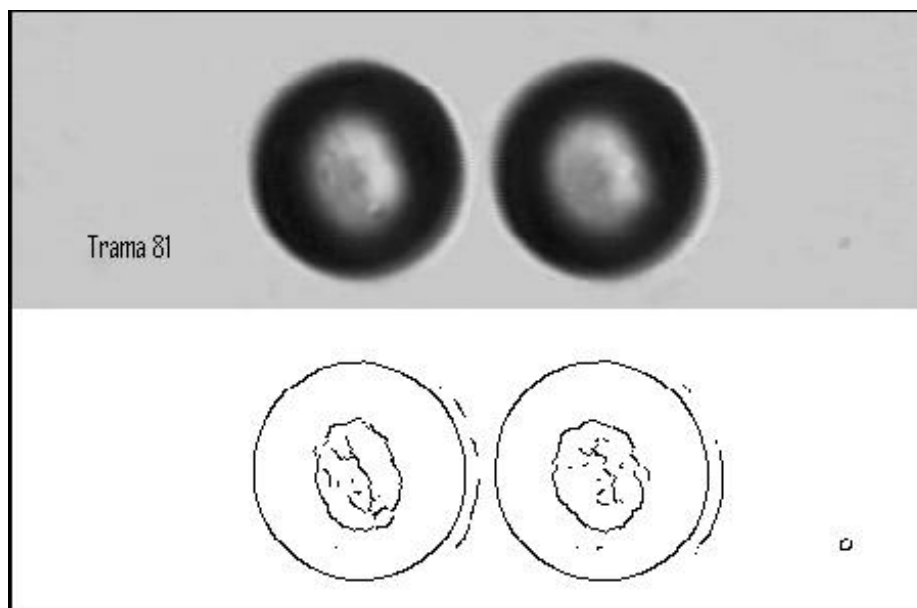


Figura 5.7: Fotografía y el resultado obtenido después de aplicar la detección de bordes.

5.5. Identificación de gotas

Como ya ha sido mencionado, después de aplicar la detección de bordes, es necesario identificar el número de objetos presentes en la imagen. Debido a la naturaleza de los procedimientos de adquisición y detección de bordes, no es posible detectar de manera perfecta los bordes de los objetos presentes en la imagen. Por una parte, están las limitantes del sistema óptico –ver la Sección 2.4–, y, por otra, las limitantes del algoritmo de detección empleado como se vio en el Capítulo 4. Un algoritmo ideal de detección de bordes arrojaría los contornos cerrados que definen a los objetos presentes en la imagen. Como se observa en la Figura 5.6, no todos los bordes definen un objeto. En este caso los objetos son gotas inmersas en un fluido que para fines de iluminación juega el papel de fondo.

Con el fin de discriminar aquellos píxeles que no pertenecen al objeto y de agrupar

aquellos que sí, se determinan los umbrales máximo y mínimo en la escala de grises de los píxeles presentes en una región que pertenece claramente a una gota. Para realizar este paso empleamos una imagen que arroje contornos cerrados al aplicar la detección de bordes como en la imagen de la Figura 5.7. A pesar de haber obtenido bordes no deseados en la imagen anterior, también se obtienen dos curvas cerradas con las cuales se puede delimitar dos objetos, sin necesidad de aplicar procesamiento extra. De esta manera, podemos conocer las características de los píxeles que pertenecen a las gotas y diferenciarlos de aquellos que no. Esto es posible porque un análisis preliminar mostró una variación significativa de los niveles de gris entre los píxeles pertenecientes a una gota y aquellos externos pertenecientes al flujo generado por el molino. Con las consideraciones anteriores es posible obtener contornos cerrados, con los cuales se calculan los parámetros geométricos de los objetos.

En las imágenes obtenidas encontramos una o dos gotas. Puede ser que desde un principio haya dos gotas que con el tiempo terminarán siendo una sola como se observa en la Figura 5.8, dependiendo de las condiciones de flujo. En otros casos como en la Figura 5.1 una gota se divide para formar dos gotas. Por default se considera que existe una o como máximo dos gotas en el área de observación, de manera que el algoritmo (ver en Apéndice A:define.m) se diseñó para identificar como máximo a dos gotas. Sin embargo se puede eliminar esta restricción para permitirle identificar mas de dos gotas si fuera necesario.

A pesar de la consideración anterior, es preciso que el algoritmo identifique también el punto a partir del cual las dos gotas se fusionan en una, o cuando una se divide en dos. Pues como se observa en la Figura 5.9, la detección de bordes indica la presencia de un objeto solo; sin embargo, podemos intuir que son dos. Para lograr que el programa identifique estas

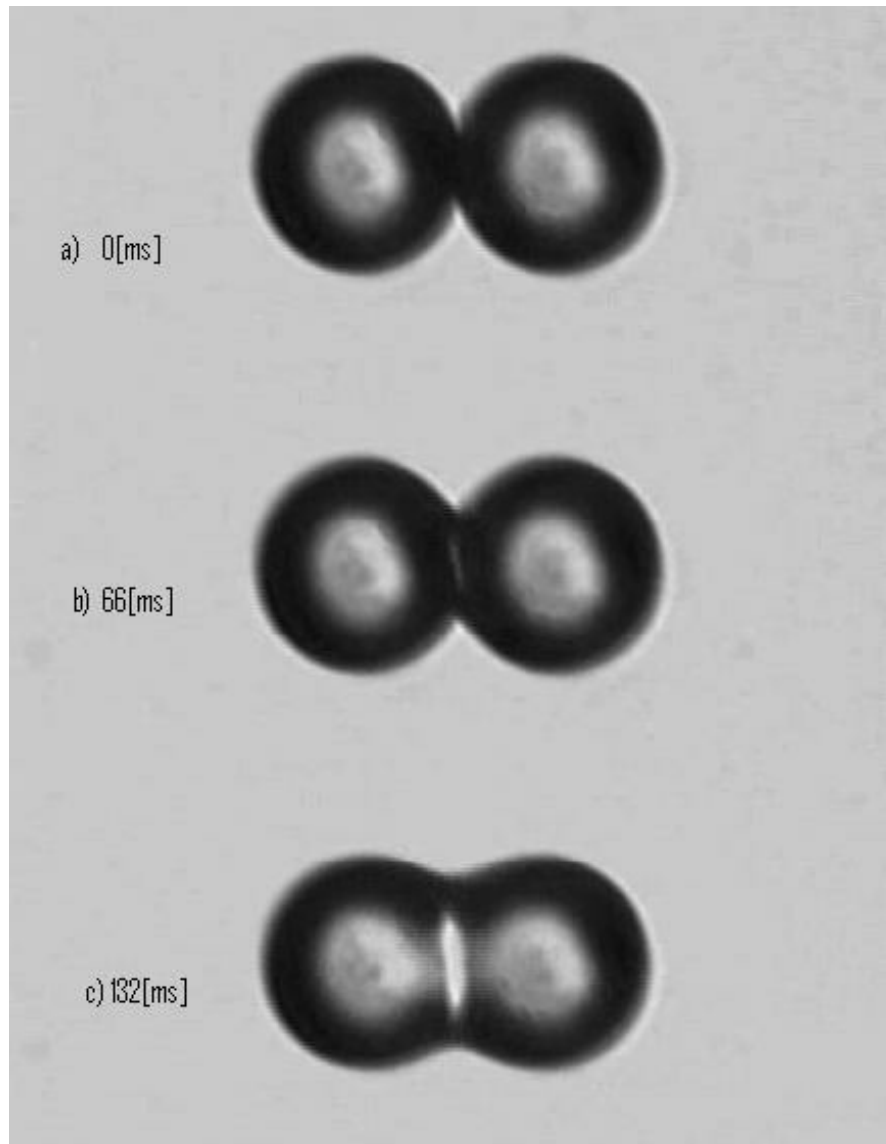


Figura 5.8: Fusión de dos gotas inmersas en el punto de estancamiento de un flujo generado por un molino de dos rodillos.



Figura 5.9: Resultado después de aplicar el algoritmo de detección de bordes. Corresponde con la parte a) tiempo 0 [ms] de la Figura anterior.

sutiles diferencias utilizamos dos parámetros geométricos: (a) el momento angular y (b) el área, considerando a ambas gotas como un solo objeto.

En la Figura 5.10 se muestra una Gráfica de los valores del momento angular y del área con respecto al tiempo, considerando ambas gotas como un solo cuerpo. En cada una de las curvas se observan dos cambios bruscos simultáneamente. El primer cambio es extremadamente rápido, pero no tan significativo como el segundo, en el cual ambas gotas se fusionan en una sola. En la Figura 5.11 se observan las gotas en los instantes correspondientes a t_1 y t_2 respectivamente, señalados en la Figura 5.10. Se distingue en el primer caso la presencia de dos gotas muy cercanas –que visualmente no es factible de definir el grado de contacto que existe entre las gotas–, mientras que en el segundo caso se presenta la fusión de ambas gotas.

Se desarrolló un programa que a partir de los bordes hace el reconocimiento de las gotas y descarta el ruido y los bordes no deseados. Este programa identifica el fondo y determina la cantidad de gotas presentes en la imagen (ver en Apéndice A:histograma.m).

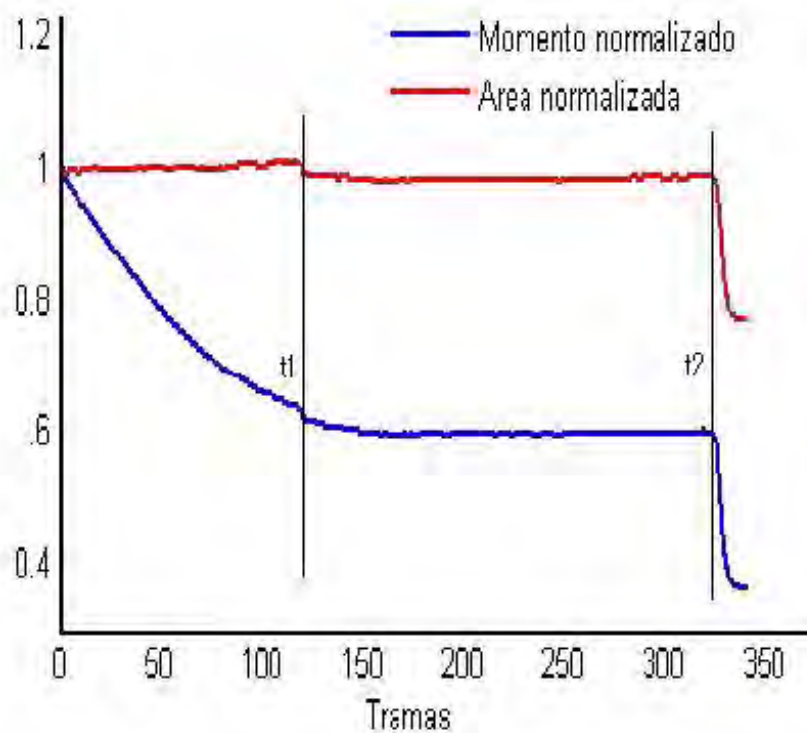


Figura 5.10: Área y momento normalizado para ambas gotas consideradas como un solo cuerpo. En el instante t_1 sucede el primer acercamiento entre ambas gotas, pero no hay fusión. En el segundo instante t_2 sucede la fusión.

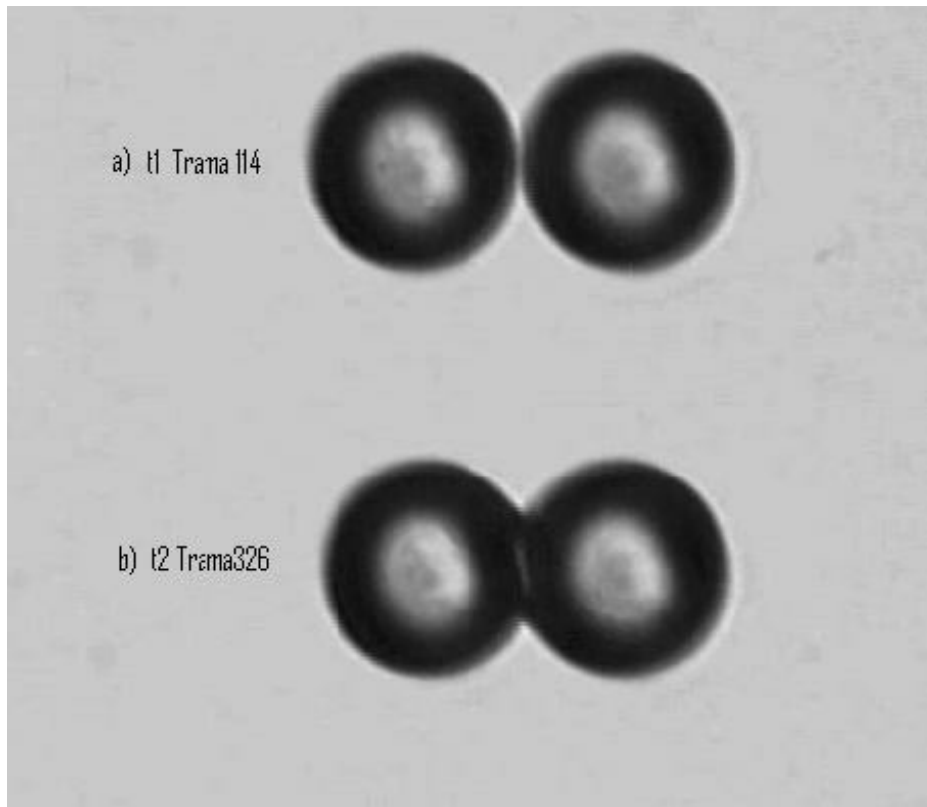


Figura 5.11: a) En el instante t_1 correspondiente a la trama 114 se observa el primer cambio brusco en área y momento para ambas gotas. b) En t_2 correspondiente con la trama 326 sucede la fusión de las gotas.

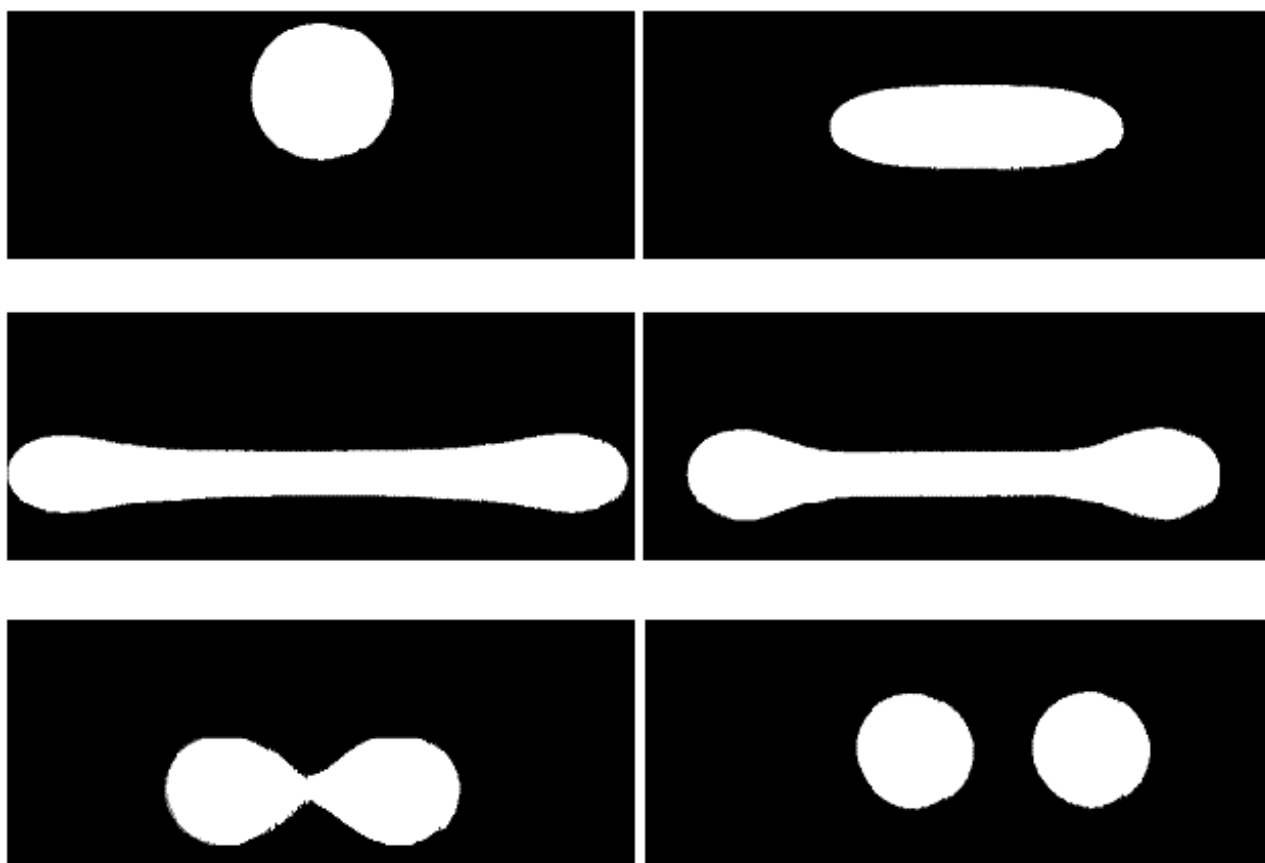


Figura 5.12: Imágenes entregadas por el programa que reconoce las gotas a partir de las imágenes de la figura 5.6.

En la Figura 5.12 se muestran los resultados de este programa tomando como entrada las imágenes de detección de bordes.

Capítulo 6

Conclusiones

En el primer Capítulo se muestra el experimento en el cual se involucra el trabajo desarrollado en esta tesis. Se explica el experimento de manera general y la motivación por la cual es importante su estudio.

A continuación en la primera parte de la tesis que corresponde con los Capítulos 2, 3 y 4 se desarrolla una investigación primordialmente teórica sobre las herramientas necesarias para la realización del objetivo de esta tesis.

Así entonces, en el Capítulo 2 se estudia el sistema de adquisición de imágenes. El cual se enfoca principalmente en la óptica de una lente, pues ésta toma gran importancia siendo el primer elemento en el sistema y del cual depende el aprovechamiento del siguiente componente (CCD) que captura la información obtenida. Se presentan las características y los parámetros que consideran los fabricantes de lentes para su elaboración.

En el Capítulo 3 se estudia a la imagen representada de manera digital y su relación con la imagen analógica y el procedimiento matemático empleado para efectuar una trans-

formación del dominio analógico al digital, es decir el muestreo y la cuantificación.

Para finalizar la primera parte, en el Capítulo 4 se estudian los conceptos y algoritmos de detección de bordes en una imagen digital.

En la segunda parte se desarrolla el trabajo experimental. Una vez montado el aparato experimental consistente en un CCD, una PC y el objeto de estudio, almacenamos la información captada por el CCD en un formato de video digital apto para ser procesado por los algoritmos, programando códigos específicamente diseñados para configurar los drivers y definir las propiedades del video.

Descomponer un video en tramas (fotografías) es indispensable para procesar la información capturada por el CCD. Los algoritmos desarrollados permitieron acotar el área y el lapso en que sucede la transición de fusión o ruptura de las gotas inmersas en el flujo, para obtener mayor velocidad de procesamiento.

Las características de las imágenes obtenidas permiten emplear un algoritmo rápido de detección de bordes. Para determinar el número de gotas presentes en el área de observación un algoritmo desarrollado identifica y elimina bordes no deseados y ruido.

En el momento crítico cuando la detección de bordes indica la presencia de un objeto solo, es preciso identificar el punto a partir del cual las gotas se fusionan en una, o cuando una se divide en dos. Esto se logra utilizando dos parámetros geométricos: el momento angular y el área.

Apéndice A

Código fuente

En las siguientes páginas se incluye el código fuente programado en Matlab, al cual hace referencia y describe el Capítulo 5. En el código se incluyen comentarios con la finalidad de hacer más accesible la comprensión de las funciones y ciclos.


```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   alzhairani.m
%   Función que realiza la detección de bordes a una imagen dada por
%   entrada, entregando dos matrices, una en escala de grises y otra con
%   solo dos colores.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [edgepix, edgebin]=alzhairani(entrada)

bw = double(entrada); % Se convierte a double la entrada y se guarda
                    % en bw.

[alto,ancho]=size(entrada); % Obtiene las dimensiones de entrada.
H=zeros(alto,ancho); % Inicializa las diferentes variables que se van
dir=ones(alto,ancho); % a utilizar.
s=H;
edgepix=H;
edgebin=H;

% -----smoothing-----

mask=[0.25 0.5 0.5 0.5 0.25; ... % Máscara de suavizado.
      0.5 0.75 1 0.75 0.5; ...
      0.5 1 2 1 0.5; ...
      0.5 0.75 1 0.75 0.5; ...
      0.25 0.5 0.5 0.5 0.25] / 16;

for a=3:alto-2 % Se aplica la máscara de suavizado para todos
  for b=3:ancho-2 % los elementos de bw y se guarda en H.
    H(a,b)=sum(sum(bw(a-2:a+2,b-2:b+3).*mask));
  end
end

% -----edge strength-----

for a=3:alto-2 % Cálculo de diferencias absolutas para todos los
  for b=3:ancho-2 % elementos de H.
    v=abs(H(a-2,b)-H(a-1,b)-H(a+1,b)-H(a+2,b));
    h=abs(H(a,b+2)-H(a,b+1)-H(a,b-1)-H(a,b-2));
    pd=abs(H(a-2,b-2)+H(a-1,b-1)-H(a+1,b+1)-H(a+2,b+2));
    nd=abs(H(a-2,b-2)+H(a-1,b+1)-H(a+1,b-1)-H(a+1,b-2));
    s(a,b)=max([v h pd nd]); % Se obtiene el valor de la máxima
                          % pendiente.
    [minidir(a,b)]=min([nd v pd h]); % Se obtiene la dirección de
    % la mínima pendiente.
  end
end

```

```

% -----detection & localization-----

threshold=sum(sum(s))/alto/ancho;    % Se define el valor del umbral.
for a=11:alto-10                      % Para todos los elementos de la matriz
    for b=11:ancho-10                 % se aplica el umbral para determinar la
        if s(a,b)>=threshold          % localización de los bordes.
            switch dir(a,b)
                case 1
                    if (s(a,b)>=s(a-1,b-1)) & (s(a,b)>s(a+1,b+1))
                        edgepix(a,b)=s(a,b);
                        edgebin(a,b)=255;
                    end
                case 2
                    if (s(a,b)>=s(a,b-1)) & (s(a,b)>s(a,b+1))
                        edgepix(a,b)=s(a,b);
                        edgebin(a,b)=255;
                    end
                case 3
                    if (s(a,b)>=s(a-1,b+1)) & (s(a,b)>s(a+1,b-1))
                        edgepix(a,b)=s(a,b);
                        edgebin(a,b)=255;
                    end
                case 4
                    if (s(a,b)>=s(a-1,b)) & (s(a,b)>s(a+1,b))
                        edgepix(a,b)=s(a,b);
                        edgebin(a,b)=255;
                    end
            end
        end
    end
end

edgepix=uint0(edgepix);    % Se convierten a enteros de 0 bits.
edgebin=uint8(edgebin);

```

```

+++++
% Botecito.m
% Función que recibe una matriz que contiene una imagen en escala de
% grises y rellena de color fcolor todos los pixeles vecinos al pixel
% entrada(1,1) que tienen el mismo color de entrada(1,1).
+++++
funcion salida=Botecito(entrada,fcolor)

[alto ancho]=size(entrada); % Obtiene las dimensiones de la imagen.
matriz=entrada; % Inicialize las variables.
indicador=0;

for j=1:alto
    for i=1:ancho
        if (matriz(i,1)==255)
            indicador=i;
            break;
        end
    end
    if(indicador~=0)
        for i=indicador:ancho
            i=(matriz(i,1)==fcolor)
            matriz(j,i)=100;
        end
    end
end
indicador=0;

end
for i=2:alto
    for i=1:ancho
        if{(matriz(j,i)==100)&(matriz(j-1,i)==fcolor)}
            matriz(j,i)=fcolor;
        end
    end
end
for i=alto:-1:1
    for i=1:ancho
        if(matriz(j,i)==100)
            i=(matriz(j+1,i)==fcolor)
            matriz(j,i)=fcolor;
        else
            matriz(j,i)=255;
        end
    end
end
end
salida=matriz;

```

```

%+++++-----+
%   centro_masa.m
% función que obtiene el centro de masa y el área de un objeto. Esta
% función recibe una matriz de elementos binarios que representa una
% imagen donde 1 denota pertenencia al objeto y 0 representa no
% pertenencia al objeto.
%+++++-----+

funcion [cenx,ceny,area]=centro_masa(nmatriz)

[alto,ancho]=size(nmatriz); % Obtiene las dimensiones.
area=sum(sum(nmatriz));    % El área es la suma de los pixeles.
penx=sum(nmatriz);        % Vector de la suma de las columnas.
peny=sum(nmatriz');       % Vector de la suma de las filas.

cenx=0;                   % Se obtiene la coordenada x del centro a
for m=1:ancho             % partir de la suma de los productos del
    cenx=cenx+m*penx(m);  % valor del vector de columnas con su
end                       % respectivo índice.
cenx=round(cenx/area);

ceny=0;                   % Lo mismo para la coordenada y del centro.
for m=1:alto
    ceny=ceny+m*peny(m);
end
ceny=round(ceny/area);

```

```

+++++
% define.m
% Función que recibe una matriz que contiene una imagen en escala de
% grises, identifica las gotas útiles y entrega la imagen modificada y
% los vectores de centros, áreas y momentos angulares de las gotas
% identificadas
+++++
funcion [matriz,centro,area,momento,parametro1,parametro2]=define(bw)

[alto, ancho]=size(bw); % Obtiene las dimensiones.
matriz=zeros(alto, ancho); % Inicializa las variables.
gotas=matriz;
matriz=bw;
negro=0;
blanco=255;

matriz(1,:)=0; % Pinta de color negro todos los pixeles en las
matriz(alto,:)=0; % Cuadro orillas de la matriz de la imagen.
matriz(:,1)=0;
matriz(:,ancho)=0;

matriz=Botecito(matriz,negro); % Pinta de color negro todos los pixeles
% del fondo

cont1=0;
cont2=0;
for i=1:alto*ancho
    if (matriz(i)~=negro)
        c=ceil(i/alto);
        ipos=(i-alto*(c-1),c); % así la almacena, si no, la desecha.
        [matriz,counter]=rellena(matriz,100,ipos);
        if counter>cont2
            cont2=counter;
        end
        if cont2>cont1
            cont1=cont2;
        end
        for a=.:alto
            for b=1:ancho
                gota2(a,b)=gocal(a,b);
                if matriz(a,b)==100;
                    gocal(a,b)=blanco;
                    matriz(a,b)=negro;
                else
                    gocal(a,b)=negro;
                end
            end
        end
    end
end
end

```

```

else
    for a=1:alto
        for b=1:ancho
            if matriz(a,b)==100;
                gota2(a,b)=blanco;
                matriz(a,b)=negro;
            else
                gota2(a,b)=negro;
            end
        end
    end
end
end
end
end
end
end
end

if cont1>(2*cont2) % Si el area de la gota más grande es por lo
    matriz=gota1; % dos veces más grande que el de la segunda, se
else % desecha la segunda. Este criterio se puede
    matriz=gota1+gota2; % modificar de acuerdo a las condiciones de
end % ruptura o unión de las gotas.

nmatriz=matriz/255; % se convierte a binaria, con valores 0 y 1.
[cenx,ceny,area]=centro_masa(nmatriz); % Se obtiene el centro de masa y
% el área.

nmatriz1=nmatriz; % Se separan las gotas y se obtiene el centro de
nmatriz2=nmatriz; % masa y área para cada una de las gotas.
nmatriz1(:,cenx:ancho)=zeros(alto,ancho-cenx+1);
[cenx1,ceny1,area1]=centro_masa(nmatriz1);
parametros1=[cenx1,ceny1,area1];
nmatriz2(:,1:cenx)=zeros(alto,cenx);
[cenx2,ceny2,area2]=centro_masa(nmatriz2);
parametros2=[cenx2,ceny2,area2];

centro=[cenx ceny]; % Vector con las coordenadas del centro.

momento=0; % Se obtiene el momento angular sumando las
for p=1:alto % distancias al centro de masa de todos los
    for q=1:ancho % pixeles que pertenecen a la gota.
        momento=momento+sqrt((p-ceny)^2+(q-cenx)^2)*matriz(p,q);
    end
end
end

matriz(ceny-4:ceny+4,cenx)=127; % Se rotulan en gris 127 sobre
matriz(ceny,cenx-4:cenx+4)=127; % imagen los centros de masa.
matriz(ceny1-4:ceny1+4,cenx1)=127;
matriz(ceny1,cenx1-4:cenx1+4)=127;
matriz(ceny2-4:ceny2+4,cenx2)=127;
matriz(ceny2,cenx2-4:cenx2+4)=127;

```

```

+++++
*   filtrado.m
* Programa que filtra una imagen obtenida de un archivo en formato BMP
* usando un filtro con perfil gaussiano y guarda la salida en otro
* archivo en formato BMP.
+++++
close all, clear all

imagen=imread('Resolucioncharc.bmp'); % Lee el archivo de imagen.
bw = double(rgb2gray(imagen)); % Convierte la estructura image a una
    % matriz en escala de grises.
tam=size(imagen); % Obtiene las dimensiones de la imagen.
ancho=tam(2);
alto=tam(1);
desporrigen=ones(alto,ancho);
II=zeros(alto,ancho);
anchofiltro=40;
for a=1:alto
    for b=1:ancho
        % Loop que genera una matriz de 1 y -1
        % intercalados.
        desporrigen(a,b)={-1}^{(a+b)};
    end
end
bwdesp=bw.*desporrigen; % Al obtener el producto termino a termino de la
    % matriz de la imagen se desplaza el origen de
    % la transformada de Fourier al centro.
F=fft2(bwdesp); % Se obtiene la transformada de Fourier en dos
    % dimensiones.
Fabs=abs(F); % Se elimina la fase a la transformada. Esto es
    % válido porque se creta de una imagen y no de
    % un holograma.
h=ancho/2+1; % Se obtienen las coordenadas del centro de la
k=alto/2+1; % matriz transformada absoluta.
for a=1:alto
    for b=1:ancho
        % Loop que obtiene el perfil gaussiano que se va
        % a multiplicar por la transformada para filtrar
        distc=(a-k)^2+(b-h)^2)^C.5; % la imagen.
        H(a,b)=exp((-distc.*distc)/(anchofiltro^2));
    end
end
G=F.*H; % Se multiplica la matriz del perfil por la
    % matriz transformada.
g=ifft2(G); % Se obtiene la transformada inversa de Fourier
    % en dos dimensiones.
bwnew=abs(g); % Muertamente se elimina la fase.
bwnew=uint8(bwnew); % Se convierten los elementos de la matriz a
    % enteros de 8 bits.
imwrite(bwnew,'salida.bmp','bmp'); % Guarda la salida en un archivo BMP.

```

```

+++++
%
%   fotos.m
% Programa que obtiene una secuencia de imagenes en formato BMP a
% partir de un archivo de video en formato AVI.
+++++

clear all, close all

mov=aviread('entrada.avi');      % Lee el archivo y lo guarda en mov.
conf=aviinfo('entrada.avi');     % Obtiene la configuración del archivo.
nombre='cfoto_0000.bmp';         % Nombre genérico del archivo de imagen.
contunidades=0;
contdecenas=0;
contcentenas=0;
contmillares=0;
xi=146; xf=500; yi=165; yf=320; % Define el area util de la imagen.
num=conf.NumFrames;             % Obtiene el numero de tramas.
maxinumtramas=9999;
fin=min(maxinumtramas,num);     % Si el número de tramas es mayor al
                                % máximo permitido lo recorta
                                % para todas las imagenes.
for i=1:fin
    trama=mov(i);                % Obtiene la trama actual y la guarda
                                % en en una estructura de tipo frame.
    foto=frame2im(trama);        % Guarda la información de trama en una
                                % estructura de tipo image.
    foto=foto(yi:yf,xi:xf);     % Recorta el área útil de la imagen.

    contunidades=contunidades+1; % Procedimiento para generar nombre al
    if(contunidades==10)        % siguiente archivo.
        contunidades=0;
        contdecenas=contdecenas+1;
        if(contdecenas==10)
            contdecenas=0;
            contcentenas=contcentenas+1;
            if(contcentenas==10)
                contcentenas=0;
                contmillares=contmillares+1;
            end
        end
    end
end

nombre(8)=48+contmillares;
nombre(9)=48+contcentenas;
nombre(10)=48+contdecenas;
nombre(11)=48+contunidades;
imwrite(foto,nombre,'bmp');     % Guarda la imagen en un archivo BMP.
end

```



```

+++++
+ histograma.m
+ Programa que realiza la identificación de gotas a una secuencia de
+ imágenes en formato BMP. El resultado es una secuencia de imágenes en
+ formato BMP con la gotas identificadas y los parámetros rotulados en
+ cada imagen.
+++++
close all, clear all

nombre = 'Fusion_0001.bmp'; % Nombre genérico del archivo de entrada.
salida = 'Fusion_ch_0001.bmp'; % Nombre genérico del archivo de salida.
n_arch = 342; % Número de archivos de la secuencia.
centros=zeros(n_arch,2); % Inicialización de variables para
areas=zeros(n_arch,1); % almacenar el vector de centros, área y
momento=zeros(n_arch,1); % momento angular.

for i=1:n_arch % Para cada una de las imágenes.
    imagen=imread(nombre,'bmp'); % Lee la imagen.
    bw = double(imagen); % Convierte a double.
    [alto, ancho]=size(imagen); % Obtiene las dimensiones.

    for a=1:alto % Para cada pixel, si nivel de gris está entre
        for b=1:ancho % 162 y 225 se le asigna color negro, de lo
            if ((bw(a,b)>162)&&(bw(a,b)<225))
                bw(a,b)=0;
            else
                bw(a,b)=255;
            end
        end
    end

    % Utiliza la función define() para seleccionar la o las gotas útiles
    % en la imagen y obtener los vectores de centros, áreas y momentos
    % angulares de las gotas.
    [bw,centros(i,:),area(i),momento(i),parametros1,parametros2] = ...
        -define(bw),

    % Rotula los parámetros en la imagen utilizando la función imprime()
    % que escribe los caracteres sobre la matriz.
    bw=imprime(sprintf('Trama: %d',1),2,4,bw,127);
    bw=imprime(sprintf('Centro: (%d,%d)',parametros1(1), ...
        parametros1(2)),30,4,bw,127);
    bw=imprime(sprintf('Área = %d',parametros1(3)),44,4,bw,127);
    bw=imprime(sprintf('Centro: (%d,%d)',parametros2(1), ...
        parametros2(2)),58,4,bw,127);
    bw=imprime(sprintf('Área = %d',parametros2(3)),72,4,bw,127);
    bw=imprime(sprintf('Centro: (%d,%d)',centros(i,1), ...
        centros(i,2)),30,4,bw,127);
    bw=imprime(sprintf('Área = %d',area(i)),44,4,bw,127);
    bw=imprime(sprintf('Momento = %.0f',momento(i)),58,4,bw,127);

```

```

salida=uint8(bw);           % Convierte a entero de 8 bits.
inwrite(salida,salidal,'bmp'); % Guarda la imagen en un archivo BMP.

nombre(11)=nombre(11)+1;   % Procedimiento para generar nombre al
if nombre(11)==' '        % siguiente archivo.
    nombre(11)='0';
    nombre(10)=nombre(10)+1;
    if nombre(10)==' '
        nombre(10)='0';
        nombre(9)=nombre(9)+1;
    end
end
salidal(14)=nombre(11);
salidal(13)=nombre(10);
salidal(12)=nombre(9);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   imagen.n
% Programa que extrae la información capturada por video.m y guarda la
% salida en archivos AVI.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tottram=600;               % Total de tramas.
data=getdata(vid,tottram); % Se extraen los datos del objeto vid.

for i=1:tottram
    foto=data(:,:,i);           % Se extrae la fotografía.
    [edgepix edgebin]=alzahrani(foto); % Se aplica detección de bordes
                                     % a la fotografía.
    fotorgb=cat(3,foto,foto,foto); % Se concatena para dar la misma
    edgepixrgb=cat(3,edgepix,edgepix,edgepix); % intensidad a los tres
    edgebinrgb=cat(3,edgebin,edgebin,edgebin); % colores.
    M(1)=im2frame(fotorgb);      % Se agrega la nueva trama a una
    Mpix(i)=im2frame(edgepixrgb); % estructura de tipo movie M, Mpix y
    Mbin(i)=im2frame(edgebinrgb); % Mbin.
end
movie2avi(M,'entrada.avi','fps',30) % Se guardan las estructuras
movie2avi(Mpix,'salidapix.avi','fps',30) % movie en archivos en formato
movie2avi(Mbin,'salidabin.avi','fps',30) % AVI.

```

```

+++++
%      imprime.m
% Función que imprime sobre la matriz de imagen lc que recibe en
% texto, a partir de las coordenadas (x,y) del color especificado y
% la matriz rotulada.
+++++

function matriz=imprime(texto,x,y,matriz,color)

load('letras'); % Se lee el archivo letras.mat que contiene el mapa de
                % bits de letras y símbolos y la relación de valores
                % ascii contra la posición de caracter en el mapa.
[temp,numcar]=size(texto); % Se obtiene el tamaño del texto.
for(i=1:numcar)           % Para cada caracter del texto ...
    val_ascii=double(texto(i)); % ... se obtiene su valor ascii, ...
    for(j=1:74)           % ... se busca su posición en el mapa
        if(ascii(j)==val_ascii) % de bits, ...
            vector(j)=1;
            break;
        else vector(j)=0;
        end
    end
    [temp,indice]=size(vector); % ... se lee el caracter del mapa ...
    mapa_car=-letras(1:13,(8*indice-7):(8*indice))*color;
                                % ... y se escribe sobre la matriz.
    matriz(x:x+12,(y+(-1)*9):(y+(i-1)*9)+7)=mapa_car;
    clear vector;
end

```

```

%+++++
%      main.m
% Programa que realiza la detección de bordes utilizando la función
% alzharani() a una secuencia de imágenes en formato BMP. El resultado
% es una secuencia de imágenes en formato BMP después de la detección
% de bordes.
%+++++

clear all, close all

nombre='Fusion_0081.bmp';      % Nombre genérico del archivo de entrada.
salida2='Fusion_bin_0001.bmp'; % Nombre genérico del archivo de salida.
n_arch = 342;                  % Número de archivos de la secuencia.

for i=0:n_arch                  % Para cada una de las imágenes.
    imagen=imread(nombre,'bmp'); % Lee la imagen.
    [edgepix,edgebin]=alzharani(imagen); % Detección de bordes.
    imwrite(edgebin,salida2,'bmp'); % Guarda la imagen en archivo BMP.

    nombre(11)=nombre(11)+1; % Procedimiento para generar nombre al
    if nombre(11)==' '       % siguiente archivo.
        nombre(11)='0';
        nombre(10)=nombre(10)+1;
        if nombre(10)==' '
            nombre(10)='0';
            nombre(9)=nombre(9)-1;
        end
    end
    salida2(15)=nombre(11);
    salida2(14)=nombre(10);
    salida2(13)=nombre(9);
end

```

```

+++++
%
% normaliza.m
% Programa que desplaza el centro de masa de la gcta al centro de masa
% la imagen. Este programa lee una secuencia de imágenes en formato
% BMP. La salida es una secuencia de imágenes en formato BMP y un
% video en formato AVI.
+++++

clear all, close all

nombre='Fusion_th_0001.bmp'; % Nombre genérico del archivo entrada.
salidal='Fusion_th_n_0001.bmp'; % Nombre genérico del archivo salida.
n_arch = 342; % Número de archivos de la secuencia.
load('Fusion_centros'); % Lee el vector de centros de masa
alto=149; % Define las dimensiones de la imagen.
ancho=301;

for i=1:n_arch % Para cada una de las imágenes.
    salida=zeros(alto,ancho);
    imagen=imread(nombre,'bmp'); % Lee la imagen.
    bw = double(imagen); % Convierte a double.

    for a=1:alto % Recorta la imagen de salida.
        for b=1:ancho
            salida(a,b)=bw(a+centros(i,2)-74,b+centros(i,1)-151);
        end
    end

    salida=uint8(salida); % Convierte a entero de 8 bits.
    imwrite(salida,salidal,'bmp'); % Guarda la imagen en un archivo BMP.
    trama=im2frame(cat(3,salida,salida,salida)); % Convierte la imagen
    mov(i)=trama; % a trama y la agrega a una estructura movie.

    nombre(14)=nombre(14)+1; % Procedimiento para generar nombre
    if nombre(14)==' ' % al siguiente archivo.
        nombre(14)='0';
        nombre(13)=nombre(13)+1;
        if nombre(13)==' '
            nombre(13)='0';
            nombre(12)=nombre(12)+1;
        end
    end
end
salidal(16)=nombre(14);
salidal(15)=nombre(13);
salidal(14)=nombre(12);
end
% Guarda la movie en un archivo en formato AVI.
movie2zavi(mov,'nalthre.avi','fps',30.3030);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   rellena.m
% Función que recibe una matriz que contiene una imagen en escala de
% grises y rellena de color fcolor todos los pixeles vecinos al pixel
% entrada(coordenadas) que tienen el mismo color. La función devuelve
% la matriz procesada y el número de pixeles que rellenó.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

funcion [salida,counter]=rellena(entrada,fcolor,coordenadas)

[alto ancho]=size(entrada); % Obtiene las dimensiones de entrada.
matriz=entrada;             % Inicializa variables.
cola=zeros(1000,2);
puntero=1;
counter=0;
cola(1,:)=coordenadas;
icolor=matriz(cola(puntero,1),cola(puntero,2));

while (puntero~=0)          % Toma de la cola las coordenadas del
    posicion=cola(puntero,:); % ultimo elemento, las pinta de color
    cola(puntero,:)=[];      % fcolor y almacena en la cola las
    puntero=puntero-1;      % coordenadas de los vecinos del mismo
                            % color. Este loop termina cuando no
                            % hay más elementos en la cola.

    if (matriz(posicion(1),posicion(2))~=fcolor)
        matriz(posicion(1),posicion(2))=fcolor;
        counter=counter+1;
        if (posicion(1)>1)
            if (matriz(posicion(1)-1,posicion(2))==icolor)
                puntero=puntero+1;
                cola(puntero,:)=[posicion(1)-1,posicion(2)];
            end
        end
        if (posicion(1)<alto)
            if (matriz(posicion(1)+1,posicion(2))==icolor);
                puntero=puntero+1;
                cola(puntero,:)=[posicion(1)+1,posicion(2)];
            end
        end
        if (posicion(2)>1)
            if (matriz(posicion(1),posicion(2)-1)==icolor);
                puntero=puntero+1;
                cola(puntero,:)=[posicion(1),posicion(2)-1];
            end
        end
        if (posicion(2)<ancho)
            if (matriz(posicion(1),posicion(2)+1)==icolor);
                puntero=puntero+1;
                cola(puntero,:)=[posicion(1),posicion(2)+1];
            end
        end
    end
end

salida=matriz;

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   video.m
% Programa que define los parametros para la captura de video y lleva
% a cabo la captura.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all, close all

vid=videoinput('dcs');           % Por default el formato de video define
                                % 30 fps y color coding de 8 bits.
vid.VideoResolution=[640 480];  % Resolución de 640 x 480
vid.framespertrigger=600;       % Para obtener 600 tramas, equivalente a
                                % 20 min de video.
start(vid);                     % Se inicia la captura de video.
```

Apéndice B

Breve introducción a la física del sistema

Para entender el comportamiento general que rige los experimentos de dinámica de gotas sometidas a flujos fuertes se requieren algunos conocimientos básicos sobre la mecánica de los medios continuos y la mecánica de fluidos. En este apéndice se introducen los principios fundamentales que permiten tener una visión general a este problema. Para un lector que desee una información más especializada sobre estos temas se sugiere consultar a Reyes [Reyes, 1997, Reyes, 2005] en donde se explica a detalle la física de los experimentos y las características del molino de dos rodillos.

B.1. Representación matemática de las propiedades físicas de un fluido

La reología es una herramienta que nos permite comprender las deformaciones y el transporte de fluidos complejos, es decir, aquellos en los que no se tiene una relación sencilla entre el esfuerzo aplicado y la rapidez de la deformación lograda. Se puede decir que los esfuerzos y la deformaciones dependen del campo de velocidades del flujo y se interrelacionan a través de las ecuaciones constitutivas.

Un breve análisis de las ecuaciones constitutivas considera en primer lugar al medio como un medio continuo. Desde el punto de vista clásico, un continuo se deriva del concepto matemático de continuidad, que afirma que la materia es continua infinitesimalmente ignorando la estructura atómica. De esta forma el tiempo, la masa (desde el punto de vista macroscópico) y el volumen se pueden considerar como propiedades físicas continuas. Otra de las suposiciones utilizadas para el análisis de fluidos es que no se consideran como cuerpos rígidos, es decir, los fluidos se desplazan y se deforman debido a los campos de fuerza que actúan sobre el fluido. Además, en el caso particular de los experimentos a realizar con el molino de dos rodillos se considera a los fluidos como no compresibles ($\rho = cte$). La última consideración que se realiza sobre la física del fluido es su comportamiento, el cual se puede clasificar como newtoniano o no newtoniano. Un fluido newtoniano es un fluido viscoso para el cual el esfuerzo de corte es linealmente proporcional a la razón de la deformación. Los ejemplos más comunes de este tipo son el agua (ideal) y el aire. Por otra parte, los fluidos no newtonianos son los de mayor interés en el estudio de la ciencia de materiales. Muchos productos como pinturas, barnices, esmaltes, así como la mayoría de las soluciones poliméri-

cas se clasifican como no newtonianos. Con frecuencia, en las etiquetas de estos productos se leen advertencias como *No drip* (sin escurrimientos), *Spread easily* (Poca resistencia al flujo) o *Leaves no brush marks* (Flujo suave sobre la superficie), las cuales implícitamente reconocen propiedades distintas a lo que se observa en un líquido como el agua.

B.2. Campo de velocidades y condición de compatibilidad

El campo de velocidad relaciona el desplazamiento relativo de las partículas dentro de un fluido con respecto al tiempo. Sea $O - xyz$ un marco de referencia y $\mathbf{u}(x, y, z)$ un campo de velocidades vectorial. Entonces $\nabla\mathbf{u}(x, y, z)$ se conoce como vector gradiente de velocidad. En notación índice, esto se puede expresar como

$$\nabla\mathbf{u}(x, y, z) \equiv \frac{\partial u_i}{\partial x_j} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} - \frac{\partial u_i}{\partial x_j} \right). \quad (\text{B.1})$$

Descomponiendo el gradiente de velocidad en parte simétrica y antisimétrica, se puede expresar de forma vectorial como

$$\nabla\mathbf{u} = \mathbf{D} + \mathbf{W} \quad (\text{B.2})$$

donde

$$\mathbf{D} = \frac{1}{2} (\nabla\mathbf{u} + \nabla\mathbf{u}^T) \quad (\text{B.3})$$

y

$$\mathbf{W} = \frac{1}{2} (\nabla\mathbf{u} - \nabla\mathbf{u}^T). \quad (\text{B.4})$$

\mathbf{D} se conoce como gradiente de rapidez de deformación, tensor de rapidez de deformación o simplemente tensor rapidez de deformación. \mathbf{W} se conoce como tensor de vorticidad o tensor de la razón de rotación y se puede interpretar físicamente como la rotación

de un cuerpo rígido (local) con una velocidad angular igual a $\frac{1}{2}|\omega|$ donde

$$\omega = \nabla \times \mathbf{u} \quad (\text{B.5})$$

y se conoce como vector de vorticidad. \mathbf{W} se relaciona con ω por medio de la siguiente ecuación:

$$2\nabla \cdot \mathbf{W} = -\nabla \times \omega. \quad (\text{B.6})$$

De lo anterior se puede concluir que para cualquier flujo en el que $\nabla \mathbf{u} \neq 0$ y $\mathbf{D} \neq 0$, puesto que el tensor \mathbf{D} es simétrico, este se puede diagonalizar obteniendo sus eigenvalores con los que se obtienen las deformaciones principales en dirección de los ejes principales a las que se les conoce como rapidez de elongación.

Estas razones se ilustran en la figura B.1. El sistema de coordenadas xy se ha rotado de tal forma que la dirección del flujo coincida con la dirección de los ejes. El primer eigenvalor α es positivo, por lo cual representa una elongación uniforme en la dirección x' . El segundo eigenvalor β tiene un valor negativo, lo cual se manifiesta físicamente como una compresión en la dirección y' . Durante un intervalo de tiempo infinitesimal dt , estos procesos deforman una esfera de radio ε en un elipsoide cuyos semiejes tienen las longitudes

$$\varepsilon(1 + \alpha dt), \quad \varepsilon(1 + \beta dt) \quad \text{y} \quad \varepsilon(1 + \gamma dt). \quad (\text{B.7})$$

Además de las componentes de deformación, la figura B.1 muestra una traslación \mathbf{u} y una rotación de la partícula M con una velocidad angular $\frac{1}{2}\omega$ perpendicular al plano xy [Larson, 1999].

Igualmente relevante es considerar la condición de compatibilidad que establece la consistencia de los datos obtenidos en un experimento de laboratorio a través de medidores

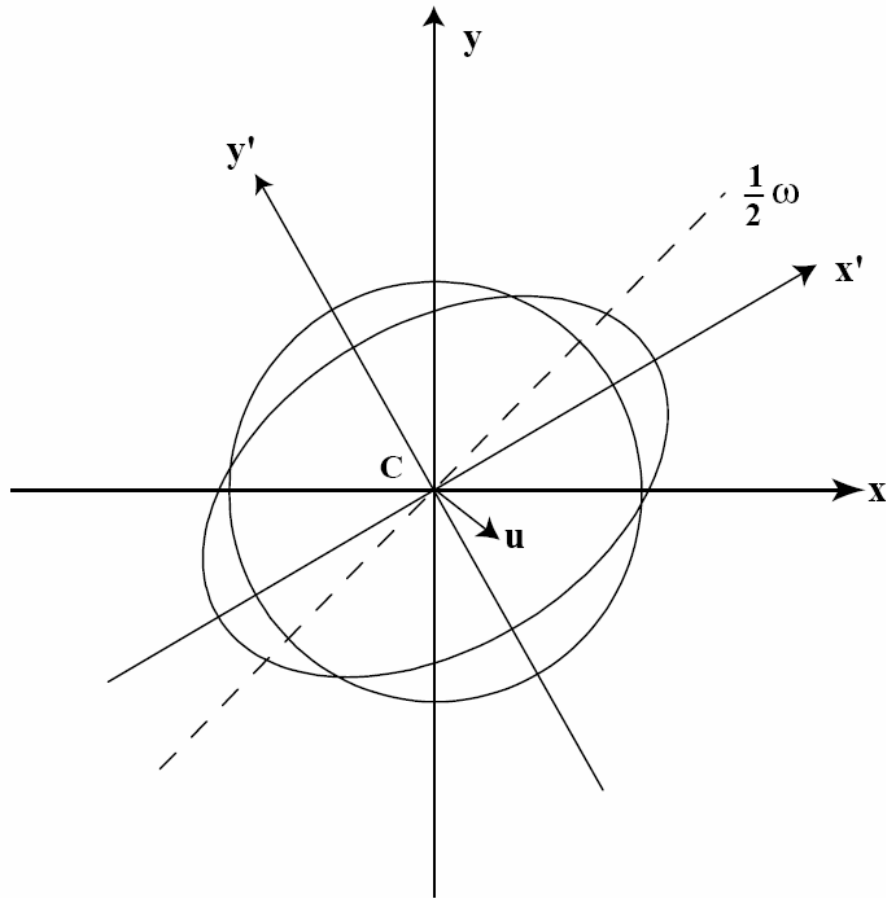


Figura B.1: Efecto del campo de velocidades sobre la partícula esférica \mathbf{M} , en sus tres componentes: (1) desplazamiento del centro de masa \mathbf{C} a la velocidad \mathbf{u} en el plano xy ; (2) rotación de M con una velocidad igual a $\frac{1}{2}\omega$; y (3) rapidez de deformación pura en dirección x' .

de deformación, equipo fotoelástico, holografía láser combinada con patrones de Moire, etc. Si los datos encontrados satisfacen esta condición entonces es posible obtener los desplazamientos \mathbf{u} y el campo de velocidad \mathbf{v} . Si establecemos un estado de deformación plano, la condición de compatibilidad toma la siguiente forma:

$$\frac{\partial^2 e_{xx}}{\partial y^2} + \frac{\partial^2 e_{yy}}{\partial x^2} = 2 \frac{\partial^2 e_{xy}}{\partial x \partial y}. \quad (\text{B.8})$$

Para el caso de un campo de velocidad bidimensional los componentes del vector de esfuerzo pueden ser medidos mediante la técnica de birrefringencia óptica. Para verificar la consistencia de los datos obtenidos se debe satisfacer la siguiente condición de compatibilidad:

$$\frac{\partial^2 V_{xx}}{\partial y^2} + \frac{\partial^2 V_{yy}}{\partial x^2} = 2 \frac{\partial^2 V_{xy}}{\partial x \partial y}, \quad (\text{B.9})$$

donde las componentes V_{ij} corresponden al tensor de esfuerzos. En la mecánica de fluidos esta condición se conoce como condición de integrabilidad.

B.3. Ecuación de Navier-Stokes

La ecuación de Navier-Stokes permite conocer las propiedades físicas para fluidos viscosos newtonianos. Se obtiene a partir de la ecuación de Cauchy y las relaciones constitutivas. De esta forma se obtiene

$$\rho \frac{Du_i}{Dt} = -\frac{\partial p}{\partial x_i} + \rho g_i + \frac{\partial}{\partial x_j} \left[2\mu e_{ij} - \frac{2}{3}\mu (\nabla \cdot \mathbf{u}) \delta_{ij} \right], \quad (\text{B.10})$$

donde $\left(\frac{\partial p}{\partial x_j} \right) \delta_{ij} = \frac{\partial p}{\partial x_i}$. La viscosidad μ es una variable que está en función de las condiciones termodinámicas. En el caso de los fluidos esta variable decrece generalmente con la

temperatura, mientras que para los gases aumenta. Sin embargo, si se considera que las variaciones de la temperatura del fluido son pequeñas, la ecuación B.10 se simplifica a

$$\rho \frac{Du_i}{Dt} = -\frac{\partial p}{\partial x_i} + \rho g_i + \frac{\partial}{\partial x_j} \left[\nabla^2 u_i - \frac{1}{3} \frac{\partial}{\partial x_i} (\nabla \cdot \mathbf{u}) \right]. \quad (\text{B.11})$$

Si además se considera que el fluido es incompresible, es decir $\nabla \cdot \mathbf{u} = 0$ y se utiliza notación vectorial, la ecuación de Navier-Stokes se puede expresar como

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \rho \mathbf{g} + \nabla \cdot \mathbf{T} \quad (\text{B.12})$$

donde \mathbf{T} es el vector de esfuerzos.

B.4. Tipos de flujo

Es posible clasificar a los flujos en base al tensor de gradiente de velocidad, siempre y cuando el flujo sea laminar y no presente fenómenos caóticos como turbulencias. Esta clasificación divide a los flujos en flujos viscométricos y flujos elongacionales [Reyes, 1997].

B.4.1. Flujos viscométricos

En este tipo de flujo cada elemento diferencial experimenta un esfuerzo cortante simple y un posible movimiento de cuerpo rígido de rotación y/o traslación. Todos los flujos viscométricos pueden entenderse como el movimiento relativo de un conjunto de superficies materiales paralelas e inextensibles. En muchas situaciones, las superficies deslizantes se mueven constantemente como cuerpos rígidos y cada superficie deslizante ocupa todo el tiempo la misma localización en el espacio. El movimiento relativo de dos cuerpos rígidos en contacto sobre un área se describe como la combinación de acciones deslizantes, giratorias

y helicoidales. Esto es útil cuando se piensa en movimientos viscométricos con superficies deslizantes rígidas, en donde las superficies pueden ser cilíndricas, superficies de revolución o helicoidales.

B.4.2. Flujos elongacionales

Los flujos elongacionales simples se caracterizan porque las partículas no giran, por lo tanto la vorticidad $\nabla \times \mathbf{u}$ es igual a cero. Entonces solamente sufren compresiones y tensiones a lo largo de sus direcciones perpendiculares con respecto a un sistema de coordenadas cartesiano. El tensor de deformación queda definido únicamente por las deformaciones en las direcciones principales, de la siguiente forma:

$$\nabla \mathbf{u} = \mathbf{D} = \begin{bmatrix} \dot{\epsilon}_1 & 0 & 0 \\ 0 & \dot{\epsilon}_2 & 0 \\ 0 & 0 & \dot{\epsilon}_3 \end{bmatrix}. \quad (\text{B.13})$$

Cuando se trata de un flujo incompresible la divergencia $\nabla \cdot \mathbf{u} = \dot{\epsilon}_1 + \dot{\epsilon}_2 + \dot{\epsilon}_3$ es igual a cero. En un flujo extensional simple e isocórico, los elementos $\dot{\epsilon}_1$, $\dot{\epsilon}_2$ y $\dot{\epsilon}_3$ son constantes para cualquier posición, aunque posiblemente dependan del tiempo. En este caso, todas las partículas sufren la misma deformación en todo tiempo por lo que se le conoce como deformación homogénea.

En el caso de flujo extensional se pueden describir varios casos de interés. Por ejemplo, el flujo uniaxial es aquel en el que el elemento de volumen se estira en una dirección a una velocidad $\dot{\epsilon}_i > 0$ y se comprime en las otras direcciones a una velocidad igual a la velocidad de elongación $\dot{\epsilon}_j = \frac{1}{2}\dot{\epsilon}_i$, $j = 2, 3$. El caso contrario en el que el elemento de volumen se comprime en una dirección y se expande en las otras dos se conoce como flujo

extensional biaxial. Otro caso de flujo extensional se genera a partir del flujo bidimensional en el que el elemento de volumen se estira en una dirección y se comprime en la otra a la misma velocidad. En este último caso $\dot{\epsilon}_3 \equiv 0$.

B.4.3. Flujos fuertes

Los flujos fuertes son aquellos flujos bidimensionales en los que la magnitud de la rapidez de deformación es mayor que la magnitud de la vorticidad. Por lo tanto, los flujos extensionales no son más que un caso particular de los flujos fuertes, en los que la vorticidad se anula.

Otro de los criterios para clasificar a un flujo como fuerte es el criterio de Astarita. Este criterio tiene las siguientes propiedades: (a) es un criterio local, pues es posible calcular su valor en un punto sin que se requiera tener información de todo el campo; (b) es objetivo, no depende del marco de referencia; (c) es generalmente aplicable, puesto que su cálculo es aplicable a la mayoría de los casos de interés. El criterio de Astarita propone que la razón de rotación puede expresarse como

$$R = \left(\frac{1 - \lambda}{1 + \lambda} \right)^2 \quad (\text{B.14})$$

en donde $\lambda^2 \leq 1$ y se extiende a través de todo el intervalo de interés, desde el movimiento de un cuerpo rígido ($\lambda = -1$) pasando por un flujo viscométrico ($\lambda = 0$) hasta el flujo elongacional ($\lambda = 1$). Por lo tanto, R es una medida de que tan rápido un material se protege por medio de la rotación contra los estiramientos severos, cuyo valor es infinito para rotaciones de cuerpo rígido y decrece monotónicamente a la unidad en el caso de los flujos viscométricos y es cero para flujos elongacionales. En este último caso, no hay rotación

que alivie los esfuerzos. Los flujos fuertes se encuentran en el intervalo entre $0 < \lambda < 1$ ($0 \leq R < 1$), y el rango de flujos débiles está entre $-1 \leq \lambda \leq 0$ ($1 \leq R < \infty$).

B.5. Importancia de los flujos fuertes

Los flujos fuertes son de gran relevancia en muchos de los procesos tecnológicos de la fabricación de materiales. Son además, los que se encuentran en la naturaleza con mayor frecuencia. En varios de estos procesos ocurren flujos extensionales con propiedades que al menos localmente corresponden con las características de un flujo elongacional simple. Dos ejemplos de la aplicación de los flujos fuertes en la industria de materiales son la extrusión para la fabricación de fibras y la producción de láminas delgadas de plástico.

En el proceso de extrusión de monofilamentos (fibras de acero, vidrio, materiales poliméricos, etc.) el material fundido se extruye a través de una boquilla y se estira aplicándole una tensión, mientras que simultáneamente se enfría. Así la fibra se adelgaza y solidifica después de viajar una cierta distancia. Los parámetros de importancia de este proceso son la longitud de la zona fundida antes de la solidificación, la fuerza de tensión, la razón de reducción de la sección transversal de la fibra, la velocidad a la que la fibra se extruye y el flujo másico en la boquilla. El proceso de extrusión tiene varias características que son el resultado del tipo de material utilizado, es decir, las características del filamento a la salida de la boquilla dependen del grado de elasticidad que tenga el material.

El segundo ejemplo de la aplicación de flujos fuertes es el *soplado de películas* usado en la fabricación de láminas delgadas. En este caso el fluido sale de una boquilla anular y forma un tubo cerrado simétrico que se enfría y solidifica a una distancia determinada de la

boquilla, mientras que una presión diferencial manométrica baja dentro del tubo contribuye a darle estabilidad durante su expansión. Para dar al extruido su espesor deseado este puede ser estirado con una extensión biaxial. Entonces, el extruido se estira en la zona de formación del tubo y no solo en la dirección anular, sino también longitudinalmente.

Debido a la importancia de las aplicaciones de los flujos fuertes en la fabricación y estudio de las propiedades de muchos materiales de aplicación industrial, resulta de gran valor estudiar, diseñar y mejorar dispositivos que generen flujos fuertes que permitan conocer y cuantificar las desviaciones de los modelos teóricos de los experimentos para poder evaluar así tales aproximaciones. Es claro que matemáticamente es posible definir condiciones de flujo que semejen a los flujos que se dan en procesos industriales, sin embargo, en el laboratorio es difícil reproducir flujos que permitan caracterizar la respuesta de un material a una deformación de tipo elongacional que sea equivalente. La caracterización de la respuesta de un material a flujos elongacionales es todavía un problema científico abierto de gran interés tanto académico como tecnológico. Por lo cual el molino de dos rodillos desarrollado en el Instituto de Investigaciones en Materiales es un avance importante en nuestro país para el estudio de los flujos fuertes.

B.6. Una gota en un flujo generado por un molino de dos rodillos

En primer lugar se estudia el comportamiento y las características de un flujo generado por un molino de dos rodillos cuyas características geométricas y cinemáticas están perfectamente definidas. Posteriormente se procede a estudiar el comportamiento de

una gota inmersa en dicho fluido.

Se puede representar matemáticamente el modelo para un molino de dos rodillos. Y entonces a partir de dicho modelo se procede a encontrar las soluciones de flujo, para los parámetros deseados.

B.6.1. Solución analítica

De la representación de la ecuación de Stokes bidimensional (flujo lento) en términos de la función de corriente, se tiene que el biarmónico de esta última es cero. Entonces, se tiene una ecuación diferencial parcial que se resuelve por el método de variables separables. Una vez obtenida la expresión para la función de corriente, se introducen las condiciones de frontera para resolver casos específicos. Aplicando condiciones de frontera en el infinito se tiene una solución analítica para el flujo generado por un molino de dos rodillos no acotado, es decir el flujo se ubica en una area infinita, en donde lo único que no es fluido son los rodillos.

En el trabajo desarrollado por Reyes [Reyes, 2005] se obtiene una solución en series de potencias para el problema de Stokes (flujos con bajo número de Reynolds) de un flujo bidimensional no acotado.

En la Figura B.2 se pueden observar las líneas de flujo generadas por un molino de dos rodillos para varias geometrías simétricas. Los resultados muestran que se pueden obtener campos de flujo complejos que no han sido estudiados y son especialmente útiles para una variedad de experimentos sobre gotas colocadas en el punto de estancamiento en campos de flujo bien caracterizados.

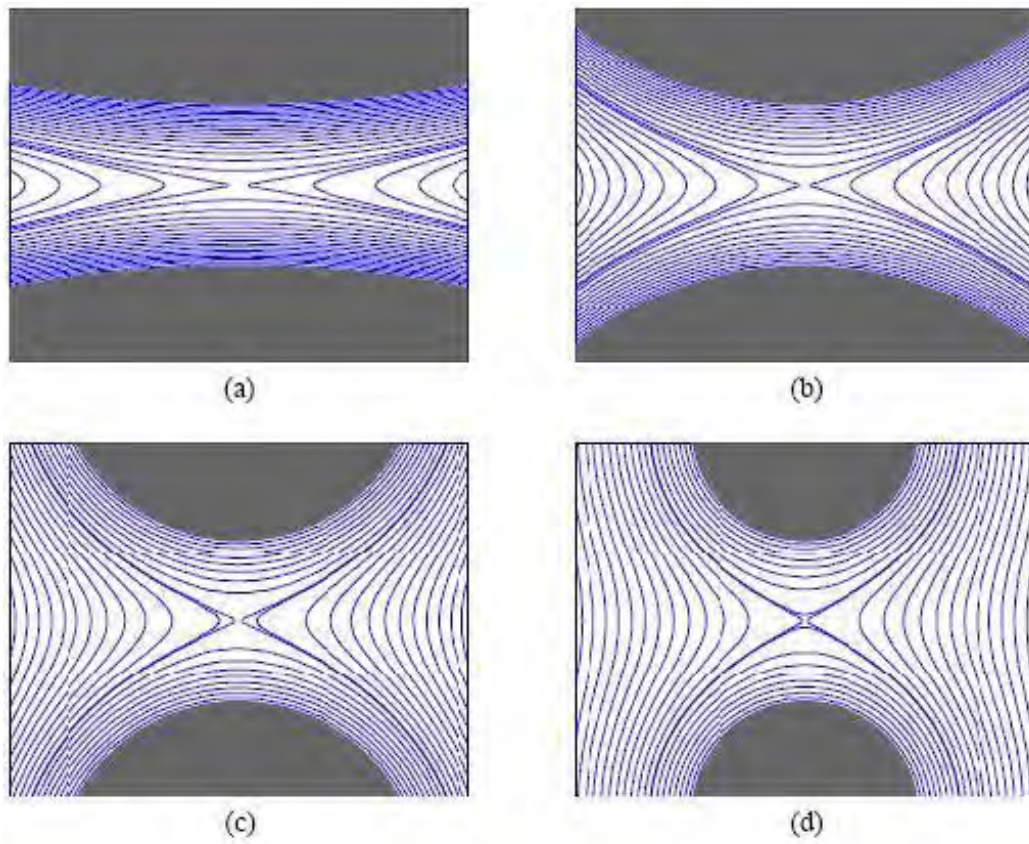


Figura B.2: Líneas de corriente para 4 distintas configuraciones simétricas en un molino de dos rodillos. (a) $r_g = 0,057, \lambda = 0,03$; (b) $r_g = 0,176, \lambda = 0,1$; (c) $r_g = 0,303, \lambda = 0,2$; (d) $r_g = 0,412, \lambda = 0,3$.

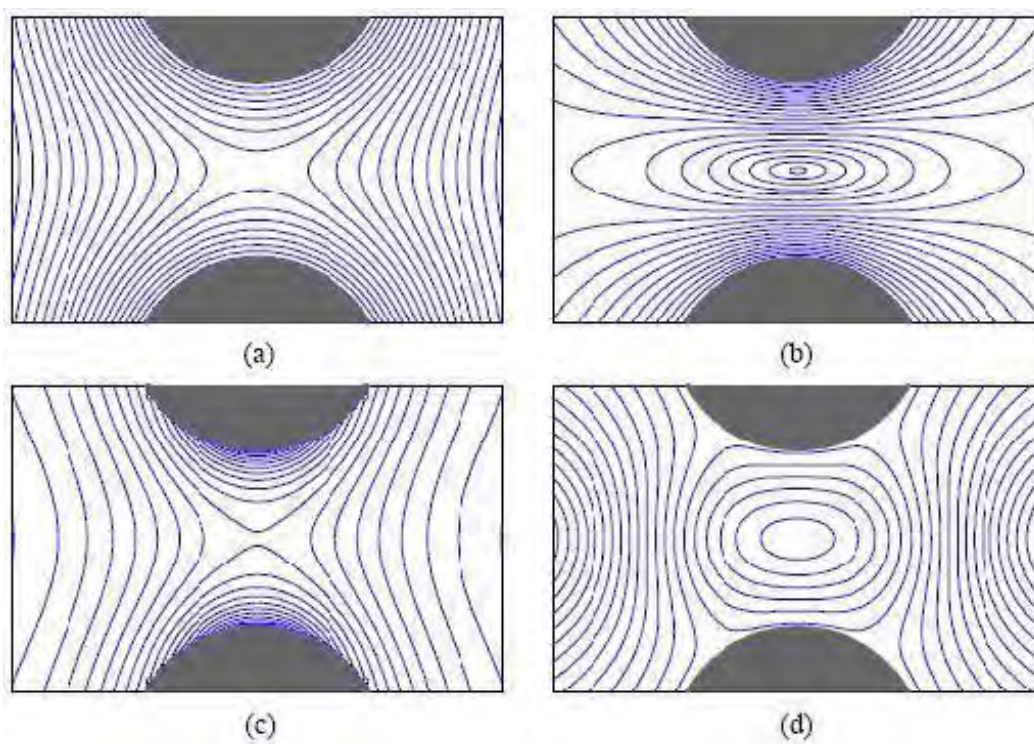


Figura B.3: Varias propiedades del flujo generado por un molino de dos rodillos con radios iguales y velocidades angulares iguales. (a) líneas de flujo (b) isopleths para la magnitud de la velocidad (c) isopleths para la magnitud del gradiente del campo de velocidad $\dot{\gamma}$ con valores máximos en la superficie de los rodillos (d) isopleths del parámetro de tipo de flujo λ con sus máximos valores en el centro del campo.

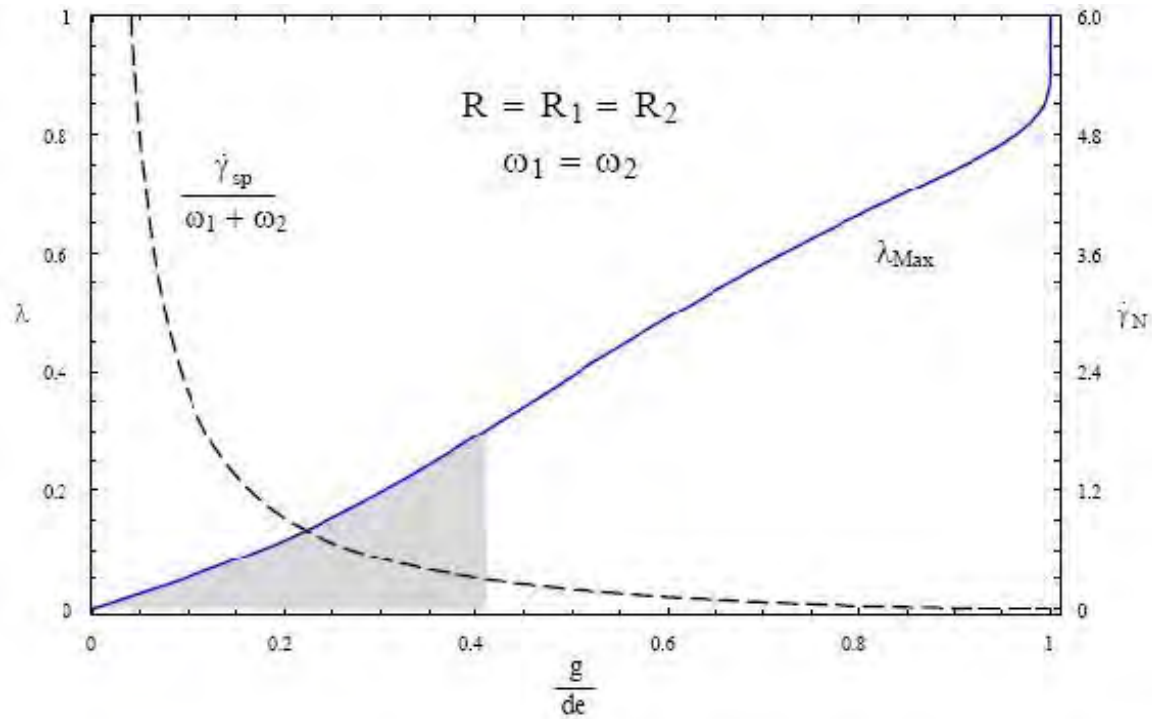


Figura B.4: Valores del parámetro de tipo de flujo λ y razón de cizallamiento normalizada $\dot{\gamma}_N$ en la región del punto de estancamiento en un molino de dos rodillos de radios iguales y velocidades angulares iguales. El área sombreada representa los valores accesibles de λ para el experimento propuesto.

En la Figura B.3 se pueden observar varias propiedades del flujo generado por un molino de dos rodillos con geometría simétrica y velocidades angulares iguales.

Para geometrías simétricas el parámetro de tipo de flujo en el punto de estancamiento tiene un valor máximo cuando la velocidad angular es igual en cada rodillo como se observa en la Figura B.4 en donde se muestran los valores del parámetro de tipo de flujo λ y el área sombreada indica la región factible de valores para la implementación del experimento.

Cabe mencionar que para valores de λ mayores, $\lambda > 4$ se vuelve más complicado de analizar en un sistema de dos rodillos, sin embargo para estos casos se adecua mejor

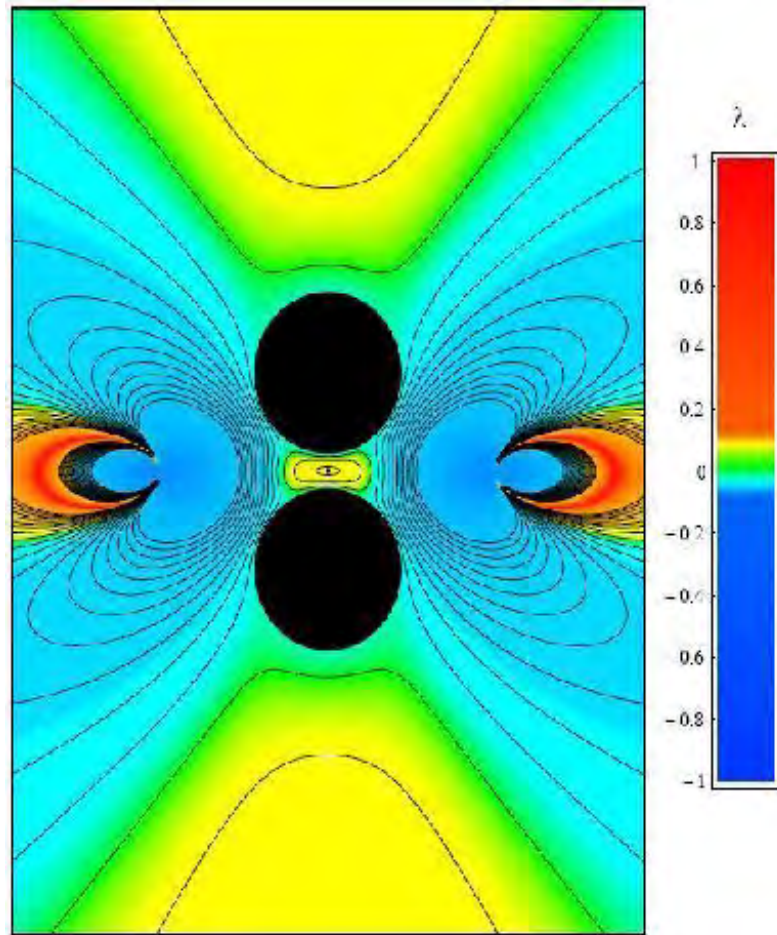


Figura B.5: Campo del parámetro de tipo de flujo λ para el caso de rodillos de radios iguales $R_1 = R_2 = 14,00mm$, y $\omega_1/\omega_2 = 1,0$.

un sistema con cuatro rodillos como ya se ha estudiado en [Bentley and Leal,1986]. En la Figura B.5 se muestra el parámetro de tipo de flujo para un molino con geometría simétrica. La región central está dominada por un comportamiento elongacional y por lo tanto valores positivos del parámetro de tipo de flujo. También se observan dos vórtices a la derecha e izquierda de los rodillos y a lo largo de la línea horizontal de simetría. En dichas regiones la velocidad de campo es débil, pero el parámetro de tipo de flujo es muy cercano a -1.

La solución analítica proporciona tres ventajas importantes:

- 1) Es de forma cerrada y puede ser aplicada a una geometría completamente asimétrica para un molino de dos rodillos.
- 2) Puede ser utilizada para calcular todos los parámetros de flujo.
- 3) Es válida para una región central extendida.

B.6.2. Solución numérica

El molino de dos rodillos exige encontrar una solución cuando las condiciones de frontera no son en el infinito pues el flujo es delimitado por una celda de flujo como se muestra en la Figura B.6. Nuevamente en el trabajo de Reyes [Reyes, 2005] se desarrolla la solución numérica para dicho flujo utilizando el método de elementos de frontera. Con base en esta solución se calcula la influencia que tiene la frontera sobre el campo de flujo, haciendo énfasis en los efectos sobre la región central del campo. Comparando los resultados analíticos y numéricos se evalúa la validez de utilizar la solución analítica no acotada para modelar la región central del flujo.

El error numérico calculado para las geometrías propuestas de manera que se pueda discernir de la contribución de las fronteras externas fue menor del 0.05 % en el punto de estancamiento.

En la Figura B.6 se muestran las dos geometrías para el trabajo experimental, una etiquetada como TRM-A y la otra como TRM-B. En donde la distancia entre los centros de los rodillos es de 34.00mm y 52.00mm respectivamente. El contorno también es diferente entre las dos configuraciones y se calculó para un acoplamiento con las líneas de flujo generadas por cada configuración de los rodillos obtenidas en la solución analítica. Para TRM-B se encontró el mayor error el cual es en el peor caso menor del 1.2 % como se

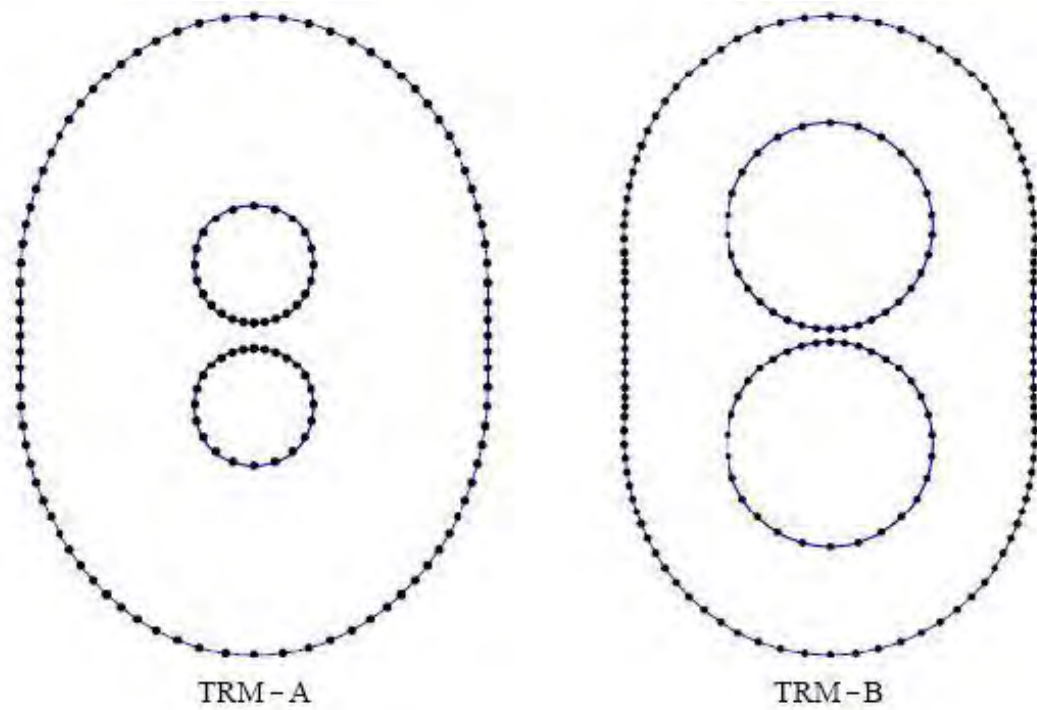


Figura B.6: Discretización de la geometría del molino de dos rodillos y su celda de flujo. Note que hay mayor concentración de elementos en la región del punto de estancamiento.

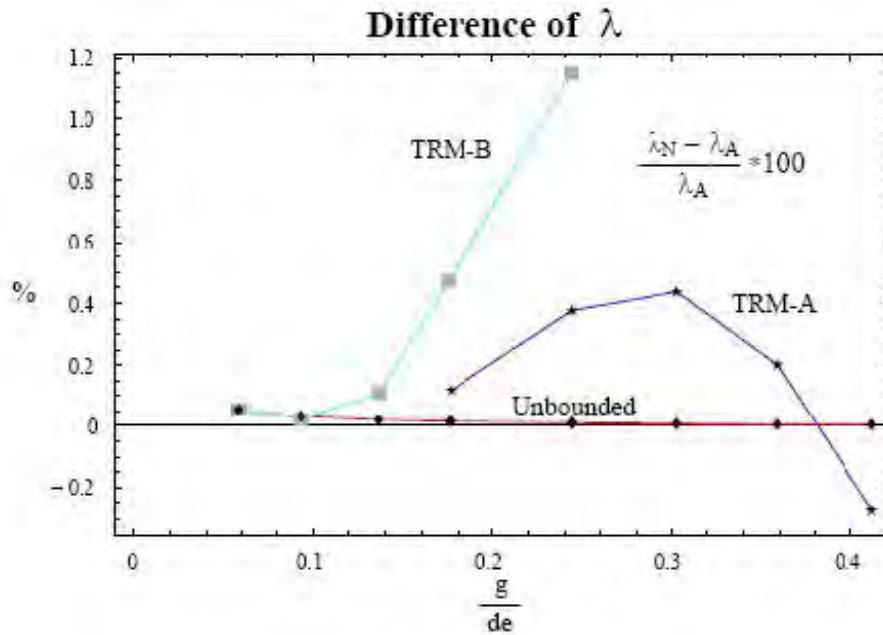


Figura B.7: Diferencia del parámetro de tipo de flujo λ entre las configuraciones TRM-A, TRMB y la solución analítica (con frontera en el infinito)

puede observar en la Figura B.7 en donde se toma como referencia el valor obtenido para la solución analítica.

B.6.3. La gota en el flujo

Habiendo caracterizado el flujo generado por un molino de dos rodillos, se introduce la gota al sistema y se enfoca la atención en el comportamiento de la gota dentro del molino. Se considera que la gota es colocada en la región de estancamiento como se muestra en la Figura B.8 . Para continuar con el estudio se hacen las siguientes consideraciones:

- Flujo lento: bajo número de Reynolds (flujo de Stokes).
- La gota no altera el flujo externo

- La velocidad de la partícula es equivalente a la de un elemento de fluido localizado en la posición de la partícula.
- No se toman en cuenta los efectos transitorios.
- La relación entre las velocidades angulares de los cilindros se calcula de acuerdo con:

$$\omega_1 R_1 + \omega_2 R_2 = p(\dot{\gamma})$$

Para la gota se hacen las siguientes aproximaciones:

- Flujo bidimensional lento
- Gota bidimensional embebida en un flujo fuerte lineal.
- Gotas bidimensionales en un flujo fuerte y lineal que considera también la presencia de fronteras cercanas.
- finalmente, las propiedades que determinan la respuesta de una gota a un flujo son:
 - Tamaño y forma inicial
 - Densidades
 - Viscosidades
 - Tensión superficial
 - Presencia de surfactantes
 - La evolución de flujo
 - Interacciones con paredes u otras gotas.

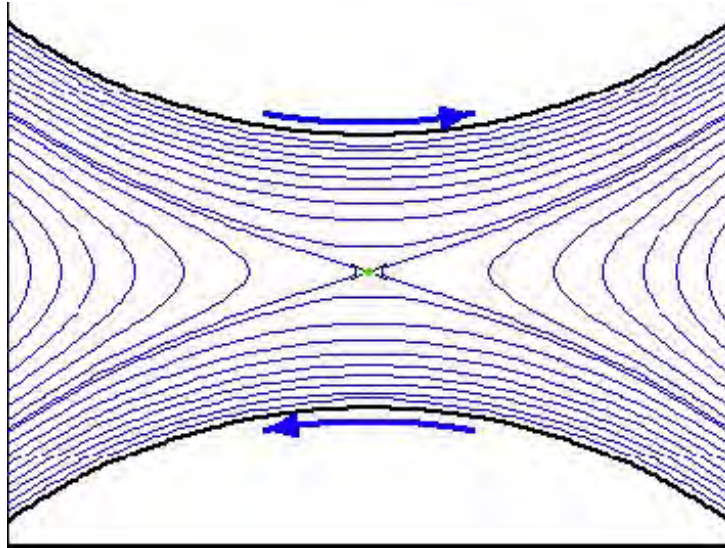


Figura B.8: La gota (partícula en verde) en el punto de estancamiento.

En la segunda parte del trabajo desarrollado por [Reyes, 2005] se lleva a cabo una simulación numérica para gotas viscosas bidimensionales inmersas en flujos fuertes generados por molinos de dos rodillos. En dicho trabajo se hace énfasis en la clase de flujos no hayan sido estudiados en trabajos anteriores. La simulación se desarrolla para varios valores de parámetro de tipo de flujo y algunos números de capilaridad. En la Figura B.9 podemos ver el efecto provocado en una gota para diferentes valores del parámetro de tipo de flujo λ . El estudio muestra que dentro del intervalo seleccionado para el parámetro de tipo de flujo y los valores de capilaridad sucede la principal transición de la respuesta de la gota al flujo externo. Se estudia entonces la evolución de la gota desde que tiene forma esférica hasta que alcanza una forma estacionaria o incluso se rompe.

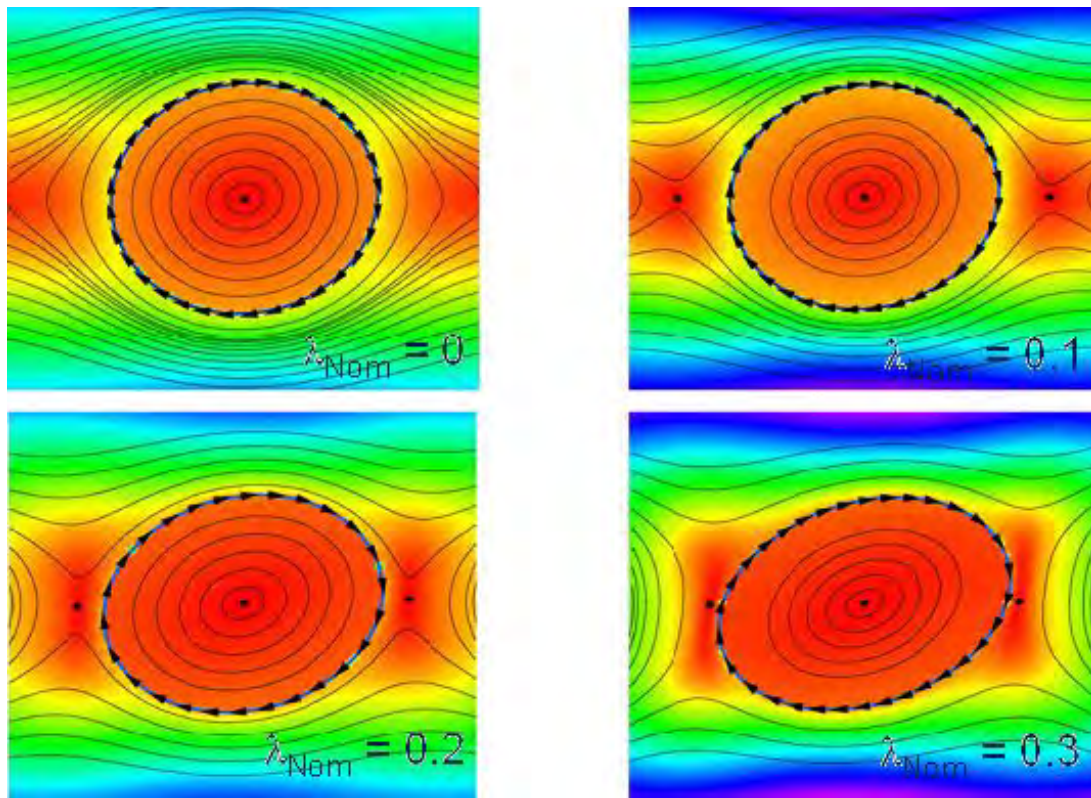


Figura B.9: Efecto de diferentes valores de λ . $C_a = 0,23$, $\lambda_u = 25,8$

Bibliografía

- [Alzarahni, 1997] Alzahrani, F. and Chen, T. (1997). A real-time edge detector: Algorithm and VLSI architecture. *Real-Time Imaging*, 3:363-378.
- [Bentley and Leal,1986] B. J. Bentley and L. G. Leal. A computer-controlled four-roll mill for investigations of particle and drop dynamics in two-dimensional linear shear flows. *Journal of Fluid Mechanics*, 167:219, 1986.
- [Canny, 1983] Canny, J. (1983). Finding edge and lines. Technical Report 720, M.I.T., Cambridge. Tech.
- [Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, PAMI-8:679-698.
- [Edmund Optics] Best of Edmund Optics Application Notes 2006. Edmund Optics.
- [Geffroy, 2000a] Reyes, M. and Geffroy, E. (2000). A corotating two-roll mill for studies of two-dimensional elongational flows with vorticity. *Physics of Fluids*, 12:2372-2376.

- [Geffroy, 2000b] Reyes, M. and Geffroy, E. (2000). Study of low Reynolds number hydrodynamics generated by symmetric corotating two-roll mills. *Revista Mexicana de Física*, 46:135-147.
- [Geffroy, 2005a] Reyes, M. and Geffroy, E. (2005). Experimental flow cell for rupture and coalescence of drops in elongational flows. 1. Analytical solution for the flow field. Submitted to *Physics of Fluids*.
- [Geffroy, 2005b] Reyes, M. and Geffroy, E. (2005). Experimental flow cell for rupture and coalescence of drops in elongational flows. 2. Numerical solution for experimental device. Submitted to *Physics of Fluids*.
- [Geffroy, 2006] Reyes, M. and Geffroy, E. (2005). Experimental flow cell for rupture and coalescence of drops in elongational flows. 3. Characteristics and control of a two-roll mill. Submitted to *Physics of Fluids*.
- [Gonzalez, 2002] Gonzalez, R. and Woods, R. (2002) *Digital image processing*. Prentice Hall.
- [Hetch, 1974] Hetch, E. y Zajac, A. (1974). *Optica*. Addison Wesley.
- [Ishii, 1996] Ishii, I. Nakabo, Y. and Ishikawa, M. (1996). Target Tracking Algorithm for 1ms Visual Feedback System Using Massively Parallel Processing, *IEEE Int. Conf. Robotics and Automation (Minneapolis, 1996.4.25)/Proc. IEEE Int. Conf. Robotics and Automation*, pp.2309-2314

- [Kovesi, 2002] Kovesi, P. Edges Are Not Just Steps. (2002). Proceedings of ACCV2002 The Fifth Asian Conference on Computer Vision, Melbourne Jan 22-25, 2002. pp 822-827.
- [Leal, 1996] Singh P, Leal LG (1996). Computational studies of the FENE dumbbell model with conformation-dependent friction in a co-rotating two-roll mill. *J Non-Newton Fluid Mech* 67:137–178.
- [Larson, 1999] Larson, R. (1999). The structure and rheology of complex fluids. Oxford University.
- [Papoulis, 1968] Papoulis, A. (1968) Systems and transforms with applications in optics. McGraw-Hill.
- [Proakis, 1999] Proakis, J. y Manolakis, D. (1999). Tratamiento digital de señales. Prentice Hall.
- [Reyes, 1997] Reyes, M. (1997). Estudio y solución analítica de un flujo fuerte generado por un molino de dos rodillos. Tesis de Licenciatura, Universidad Nacional Autónoma de México.
- [Reyes, 2005] Reyes, M. (2005). Hydrodynamics of deformable objects in creeping flows. Tesis doctoral, Universidad Nacional Autónoma de México.
- [Rishi, 2004] Rishi R. Rakesh, Probal Chaudhuri, C. A. Murthy. (2004). Thresholding in edge detection: a statistical approach. *IEEE Transactions on Image Processing* 13(7): 927-936.

- [Robinson, 2002] Robinson. J.A. (2002). Optimal Detection of Blurred Edges. 16th International Conference on Pattern Recognition, Quebec City, 11-15 August, 2002.
- [Sidney, 1999] Sidney, F. (1999). Scientific photography and applied imaging. Focal press.
- [Zvérev, 1978] Zvérev, V. (1978). Radioóptica: Transformación de señales en radio y óptica. Editorial Mir Moscú.