

Escuela Académico-Profesional de Informática
Facultad de Ciencias Físicas y Matemáticas
Universidad Nacional de Trujillo

**Diseño de un Algoritmo de Segmentación de Imágenes
aplicando el Funcional de Mumford-Shah para mejorar
el desempeño de los Algoritmos Clásicos de
Segmentación**

Tesis para la obtención del título de Ingeniero Informático

Analí J. Alfaro Alfaro

Iván A. Sipurán Mendoza

Agradecimientos

Este trabajo lo dedico especialmente a Dios, que es el verdadero propulsor de nuestras vidas, y a mis padres, Vicente, Jesús y Soledad, por su continuo amor, paciencia y dedicación, y por ser el impulso que me da fuerzas para seguir adelante cada día. Para mis hermanas, por soportarme y hacerme la vida más feliz. Para todas las personas que amo y que siempre están conmigo en los buenos y los malos momentos, mi familia. Y para una persona especial de la cual he aprendido muchas cosas, Iván gracias por ser mi mejor amigo, el mejor compañero y la persona que amo.

Analí J. Alfaro Alfaro

Quiero dedicar este trabajo a mis padres. A mi papá por todo el apoyo y el esfuerzo puesto en mí. A mi mamá por enseñarme que todo lo que se quiere se puede y porque en su ausencia sigue dándome todas sus fuerzas. A mis hermanos, María y José, quienes nunca dejaron que me falte nada. A toda mi familia por el continuo apoyo. Mi dedicación especial es para la persona que siempre creyó en mí, mi compañera incondicional, la persona que amo, Analí. Agradezco a todos ellos y a Dios por ser mi guía.

Iván A. Sipirán Mendoza

También queremos dedicar este trabajo y este esfuerzo a nuestros profesores Violeta N. Chang C., José A. Rodríguez M. y sobretodo a nuestro asesor José M. Saavedra R., porque este trabajo es el producto de las motivaciones que día a día nos impartían en el salón de clases. Y queremos hacer un agradecimiento especial al profesor Nelson Aragonés, cuyas sugerencias y consejos nos encaminaron a encontrar la solución a nuestros problemas.

Finalmente queremos agradecer a Luminita Vese, Luigi Ambrossio, Jean-Michel Morel, Guy David y Antoine Chambolle, por su gran desprendimiento y por el valioso tiempo que nos dedicaron atendiendo nuestras inquietudes.

Índice general

1. Plan de Investigación	16
1.1. Antecedentes	16
1.2. Justificación del Problema	19
1.2.1. Científica	19
1.2.2. Académica	19
1.2.3. Organizacional	20
1.2.4. Económica	20
1.3. Problema	20
1.4. Hipótesis	20
1.5. Objetivos	21
1.5.1. Generales	21
1.5.2. Específicos	21
1.6. Diseño de la Contrastación	21
1.6.1. Material de Estudio	21
1.6.2. Métodos y Técnicas	22
2. Procesamiento Digital de Imágenes	23
2.1. Definición	25
2.2. Aplicaciones del Procesamiento Digital de Imágenes	26
2.2.1. Control de Calidad	26

2.2.2. Exploración del Espacio 3D	27
2.2.3. Clasificación de Objetos	27
2.3. Representación de Imágenes	29
2.3.1. Definición	29
3. Métodos Clásicos de Segmentación de Imágenes	33
3.1. Segmentación Multiumbral	35
3.1.1. Descripción	35
3.1.2. Proceso	37
3.1.3. Algoritmo	38
3.1.4. Resultados	40
3.2. Segmentación por Crecimiento de Regiones	42
3.2.1. Descripción	42
3.2.2. Proceso	43
3.2.3. Algoritmo	44
3.2.4. Resultados	46
3.3. Segmentación Split-Merge	46
3.3.1. Descripción	46
3.3.2. Proceso	48
3.3.3. Algoritmo	50
3.3.4. Resultados	52
4. El Funcional de Mumford-Shah	55
4.1. Problemas Inversos en Procesamiento de Imágenes	55
4.2. Origen del Enfoque Variacional para la Segmentación de Imágenes	57
4.2.1. Una breve descripción del enfoque de Geman y Geman .	57
4.3. El funcional de Mumford-Shah	59
4.3.1. Análisis del Comportamiento de la Solución del Funcional	61

4.3.2. Problema de Partición Mínimo	67
5. Método de Conjuntos de Nivel	69
5.1. Introducción a los Conjuntos de Nivel	72
5.2. Conceptos Previos	73
5.2.1. Representación de Conjuntos de Nivel	76
5.3. Formulación de los Conjuntos de Nivel	79
5.3.1. Otros aspectos de la Formulación de los Conjuntos de Nivel	82
6. Métodos de Conjuntos de Nivel para minimizar el Funcional de Mumford-Shah	84
6.1. Formulación del Funcional Mumford-Shah con Conjuntos de Nivel	85
6.2. Derivación de las Ecuaciones de Euler Lagrange	89
6.3. Implementación Numérica	97
6.4. Resultados	102
7. Análisis de Resultados	105
7.1. Otros resultados	117
8. Conclusiones	121
A. Aspectos Matemáticos	125
A.1. Cálculo Variacional	125
A.1.1. Definición de Funcional	125
A.1.2. Extremo de un Funcional	125
A.1.3. Teorema Fundamental del Cálculo Variacional	126
A.1.4. Ecuación de Euler-Lagrange	127
A.2. Fórmula de Green	131

ÍNDICE GENERAL **7**

A.3. Coloreamiento de Mapas 131

Bibliografía **133**

Índice de figuras

2.1. Partes industriales que son revisadas por un sistema de percepción visual para la correcta posición y diámetro de los agujeros.	27
2.2. Imágenes de Resonancia Magnética de una cabeza humana.	28
2.3. Tareas de inspección industrial. Izquierda: Reconocimiento Óptico de Caracteres. Derecha: Conectores.	28
2.4. Proceso de muestreo y cuantización.	30
2.5. A la izquierda imagen continua. A la derecha resultado de la imagen después del proceso de muestreo y cuantización.	31
2.6. Representación matricial de una imagen.	32
3.1. Imagen con su histograma, en donde se observa tres valles como posibles umbrales.	36
3.2. Segmentación multiumbral de una imagen general.	41
3.3. Segmentación multiumbral de una imagen cerebral.	41
3.4. Segmentación multiumbral de una imagen facial.	42
3.5. Segmentación multiumbral de una imagen sintética.	42
3.6. Segmentación por crecimiento de regiones de una imagen general. El valor umbral fue configurado en 20	46
3.7. Segmentación por crecimiento de regiones de una imagen cerebral.	47
3.8. Segmentación por crecimiento de regiones de una imagen facial.	47

3.9. Segmentación por crecimiento de regiones de una imagen sintética.	48
3.10. Diferentes niveles de descomposición representados por un quadtree.	49
3.11. Segmentación split-merge de una imagen general.	52
3.12. Segmentación split-merge de una imagen cerebral.	53
3.13. Segmentación split-merge de una imagen facial.	53
3.14. Segmentación split-merge de una imagen sintética.	53
5.1. Muestra la trayectoria del camino para mover un piano.	70
5.2. Figura que muestra la sustracción del texto sobre la imagen, para dejar la imagen pura.	71
5.3. Figura que muestra la simulación de humo.	71
5.4. Figura que muestra a la izquierda la simulación de la caída de una piedra en un depósito de agua. A la derecha dos troncos de madera quemándose. . .	72
5.5. Figura que muestra la velocidad como el movimiento perpendicular a la interfaz.	74
5.6. En la siguiente figura, se aprecia los cambios de la curvatura en las zonas convexas y cóncavas de la interfaz.	75
5.7. La figura muestra que el movimiento por curvatura termina por hacer co- lapsar cualquier curva simple cerrada a un único punto.	76
5.8. A la izquierda una curva simple cerrada. A la derecha está no es una curva simple cerrada, debido a que se intersecta a sí misma.	76
5.9. La superficie de Conjunto de Nivel(en rojo) dibuja la distancia de cada punto (x,y) a la interfaz (en azul).	78
5.10. La superficie de Conjunto de Nivel(en rojo)fue movida, produciendo una nueva interfaz(en azul).	79

6.1.	Dos curvas dadas por $\phi_1 = 0$ y $\phi_2 = 0$, particionan el dominio en cuatro regiones: $\{\phi_1 > 0, \phi_2 > 0\}$, $\{\phi_1 > 0, \phi_2 < 0\}$, $\{\phi_1 < 0, \phi_2 > 0\}$, $\{\phi_1 < 0, \phi_2 < 0\}$	88
6.2.	Muestra la gráfica de la función Heaviside junto con su regularización.	98
6.3.	Muestra la gráfica de la función Delta de Dirac regularizada.	98
6.4.	Segmentación de una imagen con la presencia de múltiples objetos de la misma clase. Tamaño 175×162 . $\mu = 0,01 \cdot 255^2$	102
6.5.	Muestra la segmentación de una imagen médica de la corteza cerebral. Tamaño 131×173 . $\mu = 0,001 \cdot 255^2$	103
6.6.	Imagen facial real. Tamaño 192×192 . $\mu = 0,01 \cdot 255^2$	103
6.7.	Imagen sintética. Tamaño 64×64 . $\mu = 0,01 \cdot 255^2$	104
7.1.	(a)Imagen Original.(b)Imagen segmentada usando Multiumbral. (c)Imagen segmentada usando Crecimiento de Regiones.(d)Imagen segmentada usando Split-Merge. (e)Imagen segmentada usando Mumford-Shah.	107
7.2.	(a)Imagen Original.(b)Imagen segmentada usando Multiumbral. (c)Imagen segmentada usando Crecimiento de Regiones.(d)Imagen segmentada usando Split-Merge. (e)Imagen segmentada usando Mumford-Shah.	109
7.3.	(a)Imagen Original.(b)Imagen segmentada usando Multiumbral. (c)Imagen segmentada usando Crecimiento de Regiones.(d)Imagen segmentada usando Split-Merge. (e)Imagen segmentada usando Mumford-Shah.	112
7.4.	(a)Imagen Original.(b)Imagen segmentada usando Multiumbral. (c)Imagen segmentada usando Crecimiento de Regiones.(d)Imagen segmentada usando Split-Merge. (e)Imagen segmentada usando Mumford-Shah.	114
7.5.	(a)Imagen Original.(b)Imagen segmentada usando Multiumbral. (c)Imagen segmentada usando Crecimiento de Regiones.(d)Imagen segmentada usando Split-Merge. (e)Imagen segmentada usando Mumford-Shah.	116

7.6. Imagen sintética. Tamaño 64×64 , $\mu = 0,01 \times 255 \times 255$. 6 iteraciones. 1s.
 $T = 10$ 118

7.7. Imagen cerebral. Tamaño 117×131 , $\mu = 0,01 \times 255 \times 255$. 13 iteraciones.
 8s. $T = 5$ 118

7.8. Imagen cerebral. Tamaño 129×154 , $\mu = 0,01 \times 255 \times 255$. 10 iteraciones.
 7s. $T = 5$ 119

7.9. Imagen microscópica de glóbulos rojos. Tamaño 196×140 , $\mu = 0,01 \times$
 255×255 . 7 iteraciones. 7s. $T = 10$ 119

7.10. Imagen facial. Tamaño 118×184 , $\mu = 0,01 \times 255 \times 255$. 30 iteraciones.
 28s. $T = 1$ 119

7.11. Imagen común de arroces. Tamaño 192×192 , $\mu = 0,01 \times 255 \times 255$. 37
 iteraciones. 55s. $T = 0,52$ 120

A.1. Se observa que la función $\epsilon h(x)$ es añadida como variación a la función $y(x)$
 con valores de bordes fijos. La función de desviación se denota como $y + \epsilon h$. 128

A.2. Figura que muestra un la coloración de un gráfico 4 -colorable. 132

Índice de cuadros

- 3.1. Esta tabla muestra el alto grado de complejidad del Algoritmo Multiumbral. 40
- 7.1. Tiempo de ejecución de los algoritmos clásicos y de Mumford-Shah. 117

Índice de Algoritmos

1.	Algoritmo de Segmentación Multiumbral	39
2.	Algoritmo de Segmentación por Crecimiento de Regiones	45
3.	Algoritmo de Segmentación Split-Merge	51
4.	Algoritmo de Segmentación Mumford-Shah	100

Resumen

En este trabajo, proponemos un algoritmo para la segmentación de imágenes digitales, basado en el problema de partición mínima como un caso límite del Funcional Mumford-Shah, el cual provee los criterios que caracterizan a una buena segmentación de imágenes. Compararemos los resultados de nuestro algoritmo con los obtenidos al aplicar los métodos clásicos de segmentación de imágenes como el método multiumbral, por crecimiento de regiones y el split-merge.

El método propuesto se basa en la minimización de la energía de un funcional aplicado sobre una imagen. Para la minimización de dicho funcional introducimos el uso de los conjuntos de nivel, ya que nos permiten representar correctamente este funcional puesto que depende de entidades geométricas, en este caso la curvatura del conjunto de bordes.

La aplicación del Procedimiento Variacional Euler-Lagrange al funcional propuesto, nos permite hallar el gradiente descendiente del mismo, en forma de una ecuación diferencial parcial que depende del tiempo. El resultado es un algoritmo de segmentación de imágenes multifase. Además mostramos que con tan sólo cuatro fases es posible describir cualquier imagen tomando como referencia el Teorema de los cuatro colores, de la Teoría de Grafos.

Nuestro método produce resultados satisfactorios como ilustraremos en varias imágenes reales y sintéticas. Además, mostramos como nuestro algoritmo produce mejores resultados que los alcanzados al aplicar métodos clásicos de segmentación.

Capítulo 1

Plan de Investigación

1.1. Antecedentes

Una de las áreas de la Ciencia de la Computación que en las últimas dos décadas ha ido ganando terreno es, sin duda, el Procesamiento Digital de Imágenes. Obviamente esto sucede debido a que cada día la información visual es más importante y abundante, y es por eso que se necesitan técnicas que manipulen adecuadamente esa información.

Es más, casi toda la información que procesamos los humanos está en forma de imágenes y de alguna manera, nosotros realizamos procesamiento de imágenes sin darnos cuenta, por ejemplo el simple hecho de convertir dos imágenes registradas por nuestros ojos al cerebro en una representación tridimensional nos da la posibilidad de comprender la profundidad de los objetos.

El campo del procesamiento digital de imágenes se refiere al estudio de técnicas que permitan de alguna forma mejorar una imagen, de manera que pueda ser utilizada en etapas posteriores de procesos de visión, como por ejemplo análisis de imágenes.

Por otro lado, el análisis de imágenes emplea técnicas que extraen información de las imágenes de interés. Obviamente, para poder extraer información de las imágenes, éstas deben tener una buena calidad, y es justamente allí donde entra a tallar el procesamiento de imágenes para mejorarlas. Desde este punto de vista, el procesamiento de imágenes comprende tareas como la eliminación de ruido, mejoramiento del contraste, segmentación, detección de bordes, etc; mientras el conteo de elementos, la extracción de descriptores de objetos, etc se consideran tareas del análisis de imágenes.

Ya que la tarea que nos interesa optimizar en este trabajo es la segmentación, nos ocuparemos de detallarla en adelante.

La segmentación de imágenes puede concebirse como el particionamiento de la imagen en grupos o segmentos de píxeles los cuales son homogéneos con respecto a algún criterio. Dicha homogeneidad puede basarse, por ejemplo, en la similitud de la intensidad de los tonos de gris de los píxeles en la imagen, textura, gradiente o profundidad relativa. Existen algoritmos clásicos de segmentación que se agrupan de acuerdo a su funcionamiento, tales como los basados en Amplitud, en Clustering y Crecimiento de Regiones [26] [13] [8] [29].

En general, los algoritmos basados en amplitud emplean umbrales para la partición en segmentos. La técnica *Multiumbral* utiliza como criterio para la segmentación la intensidad de los píxeles de la imagen, también es necesario extraer el histograma de la imagen y en base a un análisis se asocia cada valle en el histograma con un umbral, de manera que se obtiene diferentes regiones o segmentos. El inconveniente de ésta técnica es que no tiene en cuenta aspectos espaciales de los objetos, además de ser un proceso muy complejo.

El método denominado por *Crecimiento de Regiones* tiene como idea fundamental elegir puntos o semillas en la imagen para tomarlas como referencia para determinar una cierta región. Este criterio puede ser la conectividad o la diferencia de intensidades entre píxels vecinos. Una de las dificultades es justamente el criterio de parada, pues es difícil establecer el término de una región y el inicio de otra. Otro de los inconvenientes es el de determinar las semillas iniciales.

Está probado que los métodos de segmentación proveen buenos resultados para propósitos generales, pero son limitados con respecto a la realización de otras tareas como la extracción de características por ejemplo en un sistema de reconocimiento de rostros, pues este requiere mantener la forma de las características esenciales como cejas, ojos, nariz y boca; y es por ello que estos métodos no se ajustan a las necesidades de este proceso.

Existen otros enfoques para segmentación como los Probabilísticos que comprenden Técnicas con Relajación, Lógica Difusa y Clasificación con Modelos de Markov y Redes Neuronales [8]. Finalmente están los métodos variacionales como los Snakes o Contornos Activos y los que se basan en la Minimización de Energía.

Los *Snakes* (y los modelos deformables en general) han sido recientemente aceptados como una técnica estándar para segmentar diferentes características en imágenes faciales, dando una buena aproximación de la forma de cada característica, además de ser fáciles de computar. Sin embargo, por tratarse de una técnica basada en el cálculo variacional, el análisis numérico es significativamente tedioso [25].

El método de Segmentación Variacional de Mumford-Shah consiste en minimizar el funcional¹:

$$E(f, \Gamma) = \nu \int_{\Omega \setminus \Gamma} (f - g)^2 + \lambda \int_{\Omega \setminus \Gamma} |\nabla f|^2 + \mu \cdot |\Gamma| \quad (1.1)$$

donde Ω constituye el dominio de la imagen. g es la imagen a segmentar. Γ es la unión de los bordes de cada región (curva segmentada), f es la imagen segmentada y $|\Gamma|$ es la longitud total de los arcos que comprende Γ . Finalmente, λ , μ y ν son pesos.

El objetivo de este trabajo de investigación es mostrar que el funcional Mumford-Shah permite formular un algoritmo de segmentación que mejore el desempeño de los algoritmos clásicos de segmentación.

1.2. Justificación del Problema

1.2.1. Científica

- Este trabajo servirá para futuras investigaciones en el campo del Procesamiento de Imágenes y Visión Computacional, permitiendo que estos campos se desarrollen en el país y se pueda adquirir un nivel científico considerable.

1.2.2. Académica

- Promover la investigación en el campo de la Visión Computacional en la Escuela Académico-Profesional de Informática y en las Universidades, dado que hasta el momento esta área es poco desarrollada a nivel local.

¹Ver Apéndice A.1.1

- Incentivar a los alumnos de Ciencia de la Computación a desarrollar investigación, con el fin de elevar el nivel académico de la Escuela de Informática.

1.2.3. Organizacional

- Motivar a las organizaciones a emplear alta tecnología en procesos que puedan ser automatizados mediante la Visión Computacional.

1.2.4. Económica

- Incentivar el desarrollo de aplicaciones para explotación y exportación, generando una fuente de ingresos económicos para el país.

1.3. Problema

¿Es posible desarrollar un algoritmo que mejore el desempeño de los algoritmos clásicos aplicados a la segmentación de imágenes?

1.4. Hipótesis

Aplicando la funcional de Mumford-Shah es posible desarrollar un algoritmo que mejore el desempeño de los algoritmos clásicos de segmentación de imágenes.

1.5. Objetivos

1.5.1. Generales

- Desarrollar un algoritmo que mejore el desempeño de los algoritmos clásicos de segmentación de imágenes aplicando el funcional de Mumford-Shah.

1.5.2. Específicos

- Lograr una segmentación que separe en forma correcta los objetos del fondo de la imagen.
- Lograr baja complejidad temporal en el proceso de segmentación de imágenes.
- Promover la investigación en el campo de visión computacional en la Escuela Académico-Profesional de Informática.

1.6. Diseño de la Contrastación

1.6.1. Material de Estudio

Para nuestros fines, el objeto de estudio básicamente está conformado por cualquier tipo de imágenes que se caracterizan por estar en escala de grises.

Estas imágenes tendrán una dimensión variable de píxels. El conjunto de imágenes de prueba serán extraídas de una base de datos aleatoria. Elegiremos, para efectos de nuestras pruebas en este documento: imágenes médicas, imágenes faciales, imágenes que describan escenas comunes e imágenes sintéticas.

1.6.2. Métodos y Técnicas

Para medir el desempeño del algoritmo propuesto fundamentalmente se hará uso de la inspección visual de las imágenes producto de la aplicación del algoritmo de segmentación, tomando en cuenta los siguientes indicadores:

- Separación adecuada de los objetos y el fondo de la imagen.
- Insensibilidad al ruido presente en la imagen, de modo que la segmentación de los objetos no sea afectada por la presencia de ruido.
- Segmentar objetos significativos, es decir, objetos de interés de acuerdo al contexto de la imagen, por ejemplo para el caso de rostros, donde el objetivo es segmentar las características faciales.
- Mantener la forma de los objetos.

Además para asegurar la eficiencia del algoritmo se estimará la complejidad temporal del mismo. Este término se define como sigue:

- **Complejidad Temporal:** Es el número de pasos que le tomará al algoritmo para obtener un resultado, para ello usaremos la Notación Asintótica.

Capítulo 2

Procesamiento Digital de Imágenes

Las imágenes están en todos lados y casi toda la información que procesamos está en forma de imágenes, por ejemplo cuando miramos una fotografía, vemos televisión, admiramos una pintura o leemos un libro. Más aún, nuestra visión es el más eficiente de nuestros sentidos [?].

Nosotros realizamos una gran cantidad de tareas de procesamiento de imágenes. Por ejemplo, cuando miramos algo, la primera imagen que nuestros ojos envían al cerebro está posiblemente fuera de foco. El cerebro intenta corregir esto ajustando los lentes oculares, entonces una nueva imagen es enviada de los ojos al cerebro. Este proceso de retroalimentación es tan rápido que no se puede percibir ni sentir. Otro ejemplo es la estereovisión, en donde nuestros ojos envían dos imágenes bidimensionales al cerebro y éste es capaz de fusionarlas en una imagen tridimensional, todo esto de manera instantánea.

El procesamiento de imágenes combina esta forma natural de cómo los humanos usan las imágenes con la matemática. Esto produce una mezcla única,

ya que las imágenes y el procesamiento de imágenes son descritos con rigor matemático pero sin perder el carácter intuitivo [21].

Pero en realidad, el procesamiento de imágenes tiene una naturaleza interdisciplinaria, porque antes de que nosotros podamos procesar una imagen, necesitamos conocer como la señal digital está relacionada a las características de los objetos en la imagen. La comprensión de estos procesos recae en la física. Luego, un sensor convierte la irradiación incidente en una forma de señal eléctrica para luego convertir esa señal en números digitales y ser procesada por una computadora digital para extraer información relevante. En esta cadena de procesos, intervienen muchas áreas: Física, Ciencia de la Computación y Matemática [17].

Más aún, las tareas del procesamiento de imágenes pueden ser parcialmente vistas como un problema de medida, el cual es parte de la ciencia conocida como metrología. Asimismo, las tareas de reconocimiento de patrones son muchas veces incorporadas dentro del procesamiento de imágenes. Existen otras disciplinas con conexiones relacionadas como: las Redes Neuronales, Inteligencia Artificial y la Percepción Visual.

El segundo aspecto importante de la naturaleza interdisciplinaria del procesamiento de imágenes es su extenso campo de aplicación. No existe campo en las ciencias naturales o disciplinas técnicas en donde el procesamiento de imágenes no sea aplicado. Esta es una de las causas por las que éste campo ha ganado terreno y se hace notorio su rápido progreso.

El interés en el procesamiento digital de imágenes proviene de dos principales áreas de aplicación: el mejoramiento de información pictórica para interpretación humana y el procesamiento de imágenes de datos para almacenar, transmitir y representar información en la percepción de máquinas autónomas.

2.1. Definición

El campo del Procesamiento Digital de Imágenes se refiere al estudio de técnicas que permitan de alguna manera mejorar una imagen, de modo que pueda ser utilizada en etapas posteriores de procesos de visión, como por ejemplo análisis de imágenes.

Por otro lado, el análisis de imágenes emplea técnicas que extraen información de las imágenes. Obviamente, para poder extraer información de las imágenes, éstas deben tener una buena calidad, y es justamente allí donde encaja el Procesamiento Digital de Imágenes para mejorar dichas imágenes.

Desde este punto de vista, el procesamiento de imágenes involucra tareas como eliminación de ruido, mejoramiento del contraste, segmentación (y la binarización, la cual es una segmentación particular), detección de bordes, etc. y el análisis de imágenes involucra tareas como conteo de elementos, extracción de descriptores de objetos, etc.

En realidad, no existe un acuerdo común entre los autores sobre los límites del procesamiento de imágenes y otras áreas relacionadas, tales como análisis de imágenes y visión computacional. Algunas veces se hace la distinción del procesamiento de imágenes como una disciplina en la cual tanto la entrada como la salida de un proceso son imágenes. Existen otros campos tales como la visión computacional cuya meta es usar el computador para emular la visión humana, incluyendo aprendizaje y la capacidad de hacer inferencias y tomar acciones basadas en entradas visuales. Es notorio que este campo es un subcampo de la Inteligencia Artificial, el cual intenta emular la inteligencia humana.

Sin embargo, un paradigma muy útil es considerar tres tipos de procesos: procesos de bajo, medio y alto nivel. Los procesos de bajo nivel involucran operaciones primitivas tales como preprocesamiento de imágenes para reducir ruido, mejorar el contraste y hacer más pronunciadas las imágenes. Un proceso de bajo nivel es caracterizado porque tanto la entrada y salidas son imágenes. Procesamiento de medio nivel sobre imágenes involucra tareas como segmentación (partir una imagen en regiones u objetos), la descripción de esos objetos para reducirlos a una forma manejable por computadora, y clasificación (o reconocimiento) de objetos individuales. Un proceso de medio nivel es caracterizado por el hecho de que las entradas generalmente son imágenes, pero las salidas son atributos extraídos de esas imágenes (por ejemplo: la identidad de un objeto). Finalmente, los procesos de alto nivel involucran realizar funciones cognitivas normalmente asociadas con visión [15].

2.2. Aplicaciones del Procesamiento Digital de Imágenes

En esta sección mostraremos algunas de las aplicaciones en donde interviene el Procesamiento Digital de Imágenes en diversos campos. Estas aplicaciones fueron extraídas de [17].

2.2.1. Control de Calidad

En este tipo de aplicaciones interesa controlar la calidad de los productos por medio de un sistema de visión, el cual en su primera fase, contempla tareas de mejoramiento de las imágenes a analizar después, por ejemplo una segmentación para separar los objetos del fondo, o tal vez una detección de

bordes para encontrar donde se posicionan los agujeros. En la figura 2.1 se muestran imágenes de piezas industriales que deben ser procesadas y analizadas.

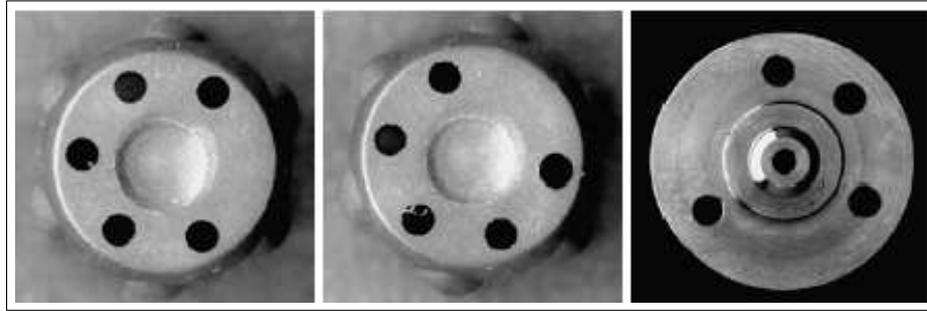


Figura 2.1: Partes industriales que son revisadas por un sistema de percepción visual para la correcta posición y diámetro de los agujeros.

2.2.2. Exploración del Espacio 3D

Las imágenes son proyecciones 2D de escenas 3D. Entonces la información de profundidad se pierde y son requeridas las técnicas para recuperar la topografía de superficies o imágenes volumétricas. Las imágenes de Resonancia Magnética es un ejemplo de una técnica de imágenes volumétricas moderna, las cuales se pueden usar para observar en el interior de un objeto 3D. Las Imágenes de Resonancia Magnética son una técnica muy flexible y dependiendo de los parámetros usados, diferentes propiedades pueden ser visualizadas(Fig. 2.2).

2.2.3. Clasificación de Objetos

Otra aplicación importante es la clasificación de objetos observados en imágenes. El ejemplo clásico de clasificación es el reconocimiento de caracteres(reconocimiento óptico de caracteres u OCR). La figura 2.3 a la izquier-

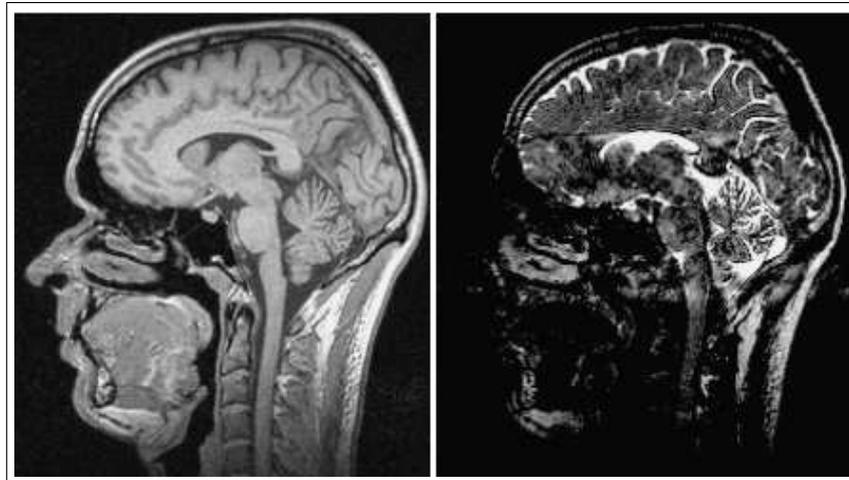


Figura 2.2: Imágenes de Resonancia Magnética de una cabeza humana.

da, muestra el reconocimiento de una etiqueta sobre un circuito integrado. La clasificación de objetos incluye también el reconocimiento de objetos en diferentes posiciones. En la figura 2.3 a la derecha, los conectores están localizados en orientaciones aleatorias. En este tipo de aplicaciones interesa eliminar todo el ruido posible, ya que este provoca que los caracteres sean poco observables. Una tarea de mejoramiento de contraste puede también ser útil.

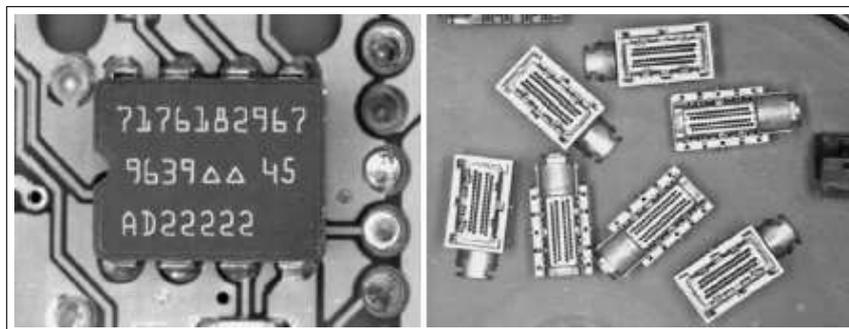


Figura 2.3: Tareas de inspección industrial. Izquierda: Reconocimiento Óptico de Caracteres. Derecha: Conectores.

2.3. Representación de Imágenes

La información contenida en las imágenes puede ser representada de diferentes maneras. Nosotros veremos la representación espacial, dejando de lado representaciones también muy útiles como la representación en número de ondas (se obtiene al aplicar la Transformada de Fourier a una representación espacial). Obviamente nos interesa también saber cómo esa representación puede ser manejada por una computadora.

2.3.1. Definición

Una imagen constituye una distribución espacial de la irradiación en un plano. Matemáticamente hablando, la distribución de irradiación espacial puede ser descrita como una función continua de dos variables espaciales.

De manera general, la imagen se define como:

$$f : \Omega \rightarrow \mathbf{C} \tag{2.1}$$

donde $\Omega \subset R^2$ y \mathbf{C} es llamado el espacio característico [6]. El Espacio Característico puede ser:

- Un intervalo, por ejemplo $[0, 255]$ o $[0, \infty]$, para imágenes en escala de grises.
- Un subconjunto de R^3 , por ejemplo $[0, 1]^3$, para imágenes a color en RGB.

Para imágenes en escala de grises, la función f puede ser vista como una superficie tridimensional.

Obviamente las computadoras no pueden manejar imágenes continuas, sino solo números o arreglos de ellos. Es por eso que se necesita representar las

imágenes como arreglos bidimensionales de puntos. El proceso de convertir una imagen continua en una digital se conoce como: muestreo y cuantización [15].

El muestreo se refiere a digitalizar los valores de las coordenadas, y la cuantización a digitalizar los valores de la amplitud, respectivamente.

En la figura 2.4 se observa una imagen continua, la cual ha sido cortada a lo largo del segmento AB, y se ha extraído una función unidimensional (arriba a la derecha). Luego se realiza el proceso de muestreo (discretización de las coordenadas) y cuantización (digitalizar las amplitudes que definirán los tonos de gris, en el caso de imágenes en escala de grises).

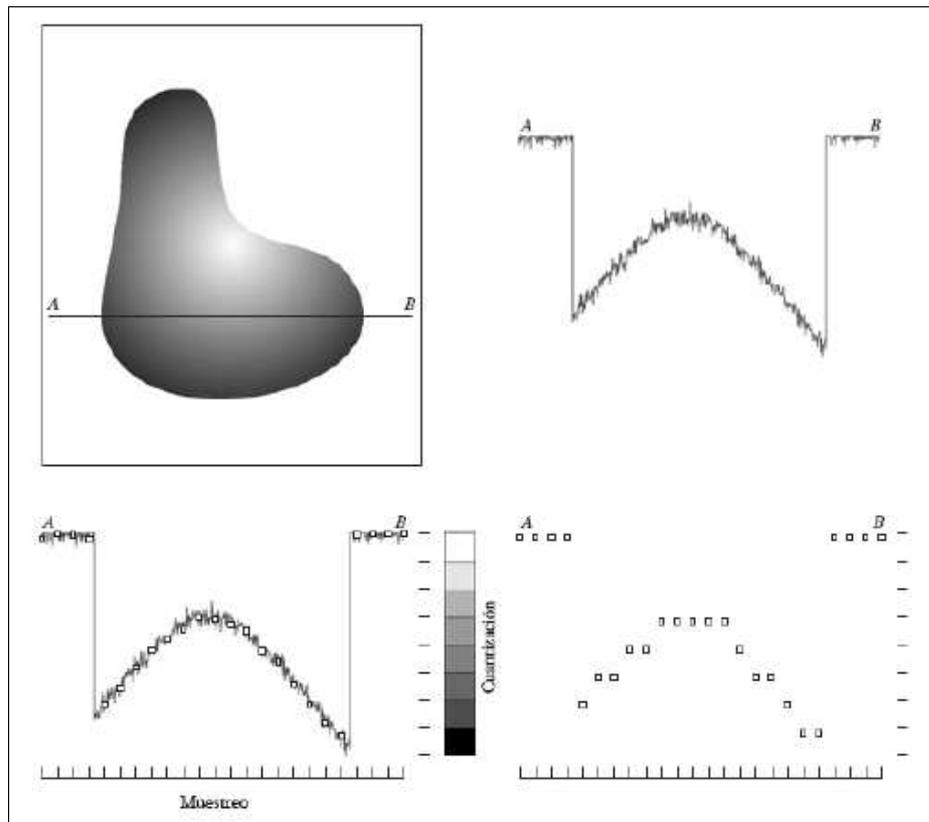


Figura 2.4: Proceso de muestreo y cuantización.

En la figura 2.5, se observa como resulta una imagen digital después de

aplicar el proceso de muestreo y cuantización.

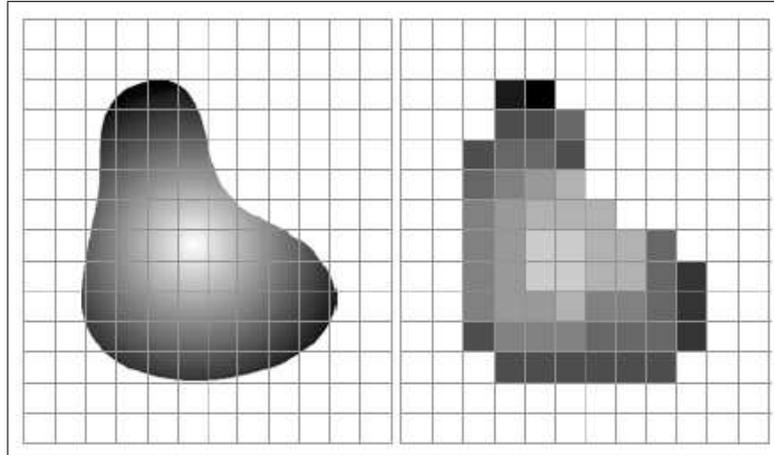


Figura 2.5: A la izquierda imagen continua. A la derecha resultado de la imagen después del proceso de muestreo y cuantización.

Es claro que la imagen digital es una aproximación de la imagen continua. Esta aproximación será mejor dependiendo de la cantidad de unidades de muestreo y cuantización que se tomaron en cuenta.

Se puede observar que las unidades de muestreo definen la resolución de una imagen y las unidades de cuantización definen el tamaño del espacio característico.

A cada punto sobre la malla bidimensional resultante se le conoce como píxel(abreviatura del término en inglés picture element). A toda la malla de píxels se le puede tratar como una matriz numérica de N filas y M columnas(Ver figura 2.6).

Evidentemente los índices de la matriz dependen del ambiente de programación en el que se trabaje, pero eso escapa al propósito de este marco teórico, por lo convenimos en usar la notación de la figura 2.6.

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

Figura 2.6: Representación matricial de una imagen.

Capítulo 3

Métodos Clásicos de

Segmentación de Imágenes

La segmentación de imágenes consiste en dividir la imagen en estructuras con significado, por ejemplo los objetos contenidos en una imagen, y de asociar cada píxel de la imagen como perteneciente a un sólo objeto de la imagen.

Una segmentación perfecta puede ser concebida como la asignación de todos los píxels al objeto correcto. Obviamente ésta es una tarea muy complicada debido a que para realizarla, muchas veces es necesario contar con información a priori de los objetos, además de la información local.

Después de realizada una segmentación, se conocen las regiones y las discontinuidades entre regiones. Luego esas regiones son empleadas para extraer información relevante sobre los objetos contenidos en la imagen.

Existen muchos métodos de Segmentación que han surgido durante este tiempo, es por eso que ha sido necesario clasificarlos de acuerdo a sus propiedades.

Aunque muchas veces, por ejemplo, dos algoritmos diferentes pueden compartir propiedades, esto no necesariamente implica que ambos pertenezcan a la misma categoría, pues podrían ser propiedades no tan relevantes.

Si consideramos técnicas que obedecen a un enfoque clásico de segmentación, la clasificación sería la siguiente [33]:

- **Basados en Umbrales**, se caracterizan por trabajar con umbrales para segmentar la imagen. Los umbrales actúan como separadores que permitirán decidir que conjunto de tonos de gris pertenece a una determinada región. Estas técnicas son aplicadas sobre una imagen completa, y también pueden combinarse con otras durante el pre-procesamiento o post-procesamiento de la imagen, de manera que se obtengan mejores resultados.
- **Basados en Clustering**, como su nombre lo indica estas técnicas tratan de agrupar un conjunto de píxeles que son similares bajo algún criterio.
- **Basados en Crecimiento de Regiones**, estas técnicas intentan segmentar una imagen partiendo desde el centro de un objeto y creciendo hacia el exterior del mismo hasta encontrar los bordes que lo limitan, este proceso es repetitivo para cada objeto dentro de la imagen.
- **Basados en Bordes**, estas permiten encontrar los bordes en una imagen, los cuales en realidad determinan los límites de cada segmento en la imagen y así poder identificar un objeto.
- **Basados en Matching**, se trata de indentificar determinados objetos en una imagen, entonces a partir de este conocimiento es posible ubicarlos en la imagen. A esto se denomina un enfoque matching.

Existen otros enfoques para segmentar como los métodos Probabilísticos, Variacionales y de Minimización de Energía.

Los Enfoques Probabilísticos consideran asignar una distribución de probabilidad a los segmentos para cada píxel. Aunque éstas hacen más complejo el proceso de segmentar y la posterior implementación [21, 33].

3.1. Segmentación Multiumbral

3.1.1. Descripción

Este método de segmentación es uno de los más usados debido a la forma como interpreta la segmentación, y a su sencilla implementación. Esta técnica es una generalización del uso de un solo umbral para segmentar una imagen, en donde el objetivo es sólo separar los objetos del fondo (binarizar), pero si hablamos de otras tareas más elaboradas que requieren distinguir entre cada objeto y el fondo, tales como la detección y el reconocimiento de objetos, será necesaria una generalización que maneje varios umbrales para llevar a cabo una correcta segmentación.

La idea de la segmentación Multiumbral o Multitresholding trata de una operación de reasignación g de los valores de grises de los pixels v_{ij} comparados con respecto a un solo valor umbral t , definido como:

$$g(x, y) = \begin{cases} 1 & \text{Si } v(x, y) < t, \\ 0 & \text{Si } v(x, y) \geq t. \end{cases} \quad (3.1)$$

Debido a que contamos con un único umbral, la imagen resultante estará binarizada, es decir obtendremos píxels con dos tonos de gris, que constituyen

dos segmentos. Si nuestro objetivo es separar los objetos del fondo, con este proceso será suficiente, pero si necesitamos realizar otro tipo de tareas es necesario trabajar con un conjunto de umbrales; esto se conoce como la técnica de Multiumbrales. Una forma de escoger los umbrales adecuados es utilizando el histograma de la imagen, donde se observan picos y valles. A cada valle del histograma se asocia un valor umbral, entonces tendremos un conjunto de umbrales (Ver figura 3.1). Todos los píxeles con un valor menor al de un umbral t_i son asignados al segmento s_i . Formalmente, esto queda expresado como sigue:

$$g(x, y) = \begin{cases} 0 & \text{Si } v(x, y) < t_1, \\ 1 & \text{Si } t_1 \leq v(x, y) < t_2, \\ 2 & \text{Si } t_2 \leq v(x, y) < t_3, \\ \vdots & \vdots \\ n & \text{Si } t_n \leq v(x, y). \end{cases} \quad (3.2)$$

donde cada segmento s_i corresponde a una región en la imagen.

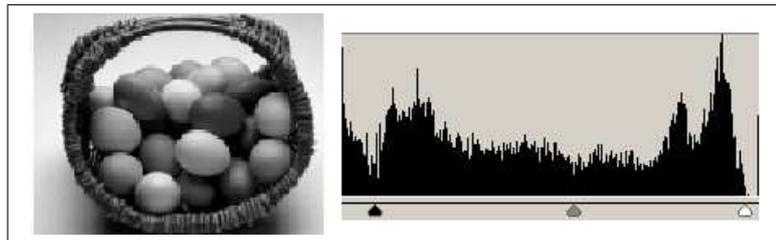


Figura 3.1: Imagen con su histograma, en donde se observa tres valles como posibles umbrales.

La técnica de Multiumbrales, es una de las más frecuentemente usadas pero a la vez es un proceso complejo, además no toma en cuenta aspectos espaciales de los objetos que componen la imagen [21].

3.1.2. Proceso

En general, es obvio que la segmentación de imágenes requiere de múltiples umbrales para obtener una correcta separación de los objetos y el fondo de la imagen. Entonces surge la interrogante, ¿Cómo seleccionar los umbrales adecuados para segmentar?. Una estrategia consiste en calcular el histograma de la imagen, donde se pueden observar picos y valles, entonces por cada valle en el histograma se asocia un umbral distinto, con lo cual tenemos un conjunto de umbrales que representan cada segmento o región en la imagen.

Un método común para hallar los umbrales que definen la segmentación de una imagen dada es usar información estadística sobre los tonos de gris en la imagen y plantear un problema de optimización con la finalidad de hallar los umbrales.

Definimos los valores admisibles de tonos de gris $\lambda_1, \lambda_2, \dots, \lambda_L$. La probabilidad discreta de cada tono de gris en una imagen puede determinarse por su histograma de frecuencias relativas. Sea I_{ij} el tono de gris en el punto (i, j) de la imagen. La probabilidad P_v (es representada por el histograma normalizado de la imagen) de que el punto (i, j) tenga el tono de gris λ_v , es igual a la frecuencia relativa:

$$P(I_{ij} = \lambda_v) = P_v = \frac{\text{Nro. de puntos de la imagen con tono } \lambda_v}{\text{Nro. total de puntos en la imagen}} \quad (3.3)$$

En base a esta probabilidad, cada valor umbral induce distribuciones, pudiendo entonces calcular también la incertidumbre que encierra cada una de esas distribuciones de probabilidad (entropía). La entropía de un conjunto de umbrales se define como sigue:

$$\begin{aligned}
\Psi(l_1, l_2, \dots, l_k) = & \log_2 \sum_{v=1}^{l_1} P_v + \log_2 \sum_{v=l_1+1}^{l_2} P_v + \dots + \log_2 \sum_{v=l_k+1}^L P_v \\
& - \frac{\sum_{v=1}^{l_1} P_v \log_2 P_v}{\sum_{v=1}^{l_1} P_v} - \frac{\sum_{v=l_1+1}^{l_2} P_v \log_2 P_v}{\sum_{v=l_1+1}^{l_2} P_v} - \dots \\
& - \frac{\sum_{v=l_k+1}^L P_v \log_2 P_v}{\sum_{v=l_k+1}^L P_v}
\end{aligned} \tag{3.4}$$

La finalidad es encontrar el conjunto l_1, l_2, \dots, l_k de umbrales que maximicen $\Psi(l_1, l_2, \dots, l_k)$.

Algunas de las limitaciones del uso de este método son que es difícil identificar correctamente los mínimos (mínimo valor en un valle) en el histograma. Se tienen problemas cuando las regiones varían suavemente su nivel (sombreado, por ejemplo) y se aplica sólo cuando hay pocas regiones. No se pueden distinguir regiones separadas de niveles similares de gris (conectividad). Además es un proceso complejo puesto que se tiene que hacer una búsqueda exhaustiva para encontrar el mejor conjunto de valores de gris que servirán de umbrales.

3.1.3. Algoritmo

En esta sección se muestra el algoritmo para calcular un conjunto de umbrales para la segmentación multiumbral (Ver algoritmo 1).

Este algoritmo es muy complejo, debido a que hace una búsqueda exhaustiva del mejor conjunto de umbrales que definan las regiones de una imagen.

El parámetro sobre el cual se define la complejidad computacional de este método es el número k de umbrales que se buscan. En el peor de los casos (cuando $ini = 0$ y $fin = 255$), el algoritmo debe generar todas las posibles

Algoritmo 1 Algoritmo de Segmentación Multiumbral

Entrada: Imagen I , Nro. de umbrales k **Salida:** Imagen segmentada I_S

```

1: Procedimiento SEGMENTACIÓNMULTIUMBRAL( $I,k$ )
2:   Sea  $ini$  el primer tono de gris con  $histograma \neq 0$ 
3:   Sea  $fin$  el último tono de gris con  $histograma \neq 0$ 
4:   Sea  $l$  el conjunto de  $k$  umbrales

5:   Iniciar  $l$  con los  $k$  primeros tono de gris a partir de  $ini$ 
6:    $sum \leftarrow 0$ 
7:    $may \leftarrow -10000$ 

8:   Mientras Verdadero Hacer
9:      $sum \leftarrow Entropia(l)$ 
10:    Si  $sum > may$  Entonces
11:       $may \leftarrow sum$ 
12:       $lMayor \leftarrow l$ 
13:    Fin Si

14:     $p \leftarrow k$ 
15:     $q \leftarrow fin$ 

16:    Mientras  $l(p) < fin$  Hacer
17:       $l(p) \leftarrow l(p) + 1$ 
18:       $sum \leftarrow Entropia(l)$ 
19:      Si  $sum > may$  Entonces
20:         $may \leftarrow sum$ 
21:         $lMayor \leftarrow l$ 
22:      Fin Si
23:    Fin Mientras

24:    Si  $l(1) = fin - k + 1$  Entonces
25:      Salir
26:    Fin Si

27:    Mientras  $l(p) = q$  Hacer
28:       $p \leftarrow p - 1$ 
29:       $q \leftarrow q - 1$ 
30:    Fin Mientras

```

```

31:       $l(p) \leftarrow l(p) + 1$ 

32:      Mientras  $p < k$  Hacer
33:           $l(p + 1) \leftarrow l(p) + 1$ 
34:           $p \leftarrow p + 1$ 
35:      Fin Mientras
36:  Fin Mientras

37:  Calcular la nueva imagen  $I_S$  en base a  $lMayor$ 
38:  Retornar  $I_S$  ▷ La imagen segmentada
39: Fin Procedimiento

```

combinaciones de 256 tonos de gris en grupos de k umbrales, por lo que la regla de correspondencia para la complejidad de este algoritmo es:

$$T(n) = \frac{256!}{n!(256 - n)!}, \quad (3.5)$$

la cual corresponde con el número de combinaciones posibles de 256 tonos de gris en conjuntos de n elementos.

La siguiente tabla muestra el alto grado de complejidad de este algoritmo, y justifica porque es poco eficiente para segmentar muchas regiones.

n	T(n)
1	256
2	16320
3	2763520
⋮	⋮

Cuadro 3.1: Esta tabla muestra el alto grado de complejidad del Algoritmo Multiumbral.

3.1.4. Resultados

En esta sección mostramos los resultados obtenidos al aplicar la segmentación multiumbral sobre imágenes sintéticas y reales. Todas las imágenes mostradas aquí

fueron segmentadas con $k = 2$, es decir solo encontrará 3 regiones.



Figura 3.2: Segmentación multiumbral de una imagen general.

En la figura 3.2 vemos como el emplear sólo 3 regiones a implicado que el algoritmo resalte las sombras de los números, que originalmente son casi imperceptibles.



Figura 3.3: Segmentación multiumbral de una imagen cerebral.

El hecho de emplear información global de la imagen(en este caso el histograma) para calcular los umbrales que generarán las regiones, y obviando información espacial de los objetos en la escena, hace que el algoritmo genere objetos que no están en la escena original. Así es el caso de las figuras 3.4 y

3.5, en donde la imagen segmentada muestra pequeños puntos aislados que no pertenecen a la imagen original.



Figura 3.4: Segmentación multiumbral de una imagen facial.

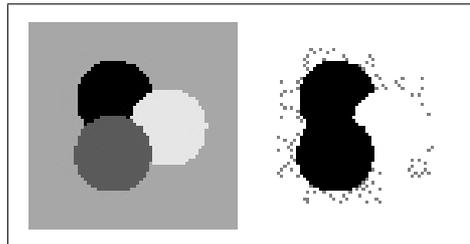


Figura 3.5: Segmentación multiumbral de una imagen sintética.

3.2. Segmentación por Crecimiento de Regiones

3.2.1. Descripción

Ya que la idea intuitiva de segmentar una imagen es separar las regiones que en ellas aparecen, la segmentación por crecimiento de regiones intenta hallar esas regiones directamente de la imagen y no por medio de estructuras globales como el histograma. Además ahora se tendrá en cuenta la disposición espacial de las regiones (las cuales pertenecen a los objetos) en la imagen.

Si representamos la región de la imagen entera como R , podemos ver la segmentación como el proceso que particione R en n subregiones R_1, R_2, \dots, R_n tal que:

1. $\bigcup_{i=1}^n R_i = R$.
2. R_i es una región conexa, $i = 1, 2, \dots, n$.
3. $R_i \cap R_j = \emptyset, \forall i, j, i \neq j$.
4. $P(R_i) = VERDADERO$
5. $P(R_i \cup R_j) = FALSO, \forall i, j, i \neq j$.

donde $P(R_i)$ es un predicado lógico definido sobre los puntos en el conjunto R_i y \emptyset denota el conjunto vacío.

La primera afirmación implica que todo píxel debe pertenecer a una región. La segunda afirmación conlleva a que toda región en la imagen debe ser conexa, es decir se asume que los píxels de una región están todos conectados de alguna forma, obviamente porque pertenecen al mismo objeto o región. La tercera afirmación asegura que las regiones deban ser totalmente disjuntas, es decir que un píxel pertenezca a una sola región a la vez.

3.2.2. Proceso

La segmentación por crecimiento de regiones intenta agrupar píxels o subregiones basado en un criterio de similitud predefinido que corresponde al predicado lógico P visto anteriormente.

Una estrategia común es empezar con un cierto conjunto de semillas, en base a las cuales se van agrupando píxels vecinos que tengan propiedades

similares. Entre los principales criterios tomados en cuenta tenemos: la similitud en la intensidad de los píxels y la varianza, considerando que la selección de este criterio depende del tipo de imagen disponible.

Existen desventajas que hacen que este método sea poco práctico en un nivel general. Uno de los inconvenientes es seleccionar el criterio adecuado, ya que esto depende de que clase de imágenes queremos segmentar. Por ejemplo, una imagen en la que existen objetos diferentes que tienen intensidades similares, el criterio de similaridad por intensidad no serviría, debido a que agruparía dos objetos que en realidad son diferentes.

Otro de los principales inconvenientes de este método es la selección inicial de las semillas. Fácilmente se puede notar que sin información a priori de la ubicación aproximada de los objetos en la escena, es prácticamente imposible llegar a formular un conjunto inicial de semillas que nos asegure resultados satisfactorios.

Un inconveniente que también vale la pena señalar es el criterio de parada, es decir en qué momento de la ejecución, el agrupamiento de píxels no debe agregar más píxels a una región. Una idea intuitiva es la de parar cuando ya no hayan más píxels que satisfagan el criterio de similaridad.

3.2.3. Algoritmo

En esta sección se muestra el Algoritmo de Segmentación por Crecimiento de Regiones (Ver algoritmo 2).

En el algoritmo anterior, la forma del predicado P , dependerá del criterio que se emplee. Por lo general representará un número de tolerancia para la similitud en intensidad de píxels o la tolerancia en la varianza de los píxels.

Algoritmo 2 Algoritmo de Segmentación por Crecimiento de Regiones

Entrada: Imagen I , Predicado P

Salida: Imagen segmentada I_S

```

1: Procedimiento SEGMENTACIÓNCRECIMIENTOREGIONES( $I, P$ )
2:   Generar el conjunto  $S$  de semillas
3:   Sea  $I_S$  una imagen, tal que  $I_S = 0$ 

4:   Para todo  $s \in S$  Hacer
5:     Sean  $v_i$  los 4-vecinos sin marca de  $s$  en  $I$ 
6:     Si  $P(s \cup v_i)$  Entonces
7:       Marcar  $s$  y  $v_i$  con su marca de región en  $I_S$ 
8:       Apilar  $v_i$  en  $S$ 
9:     Fin Si
10:  Fin Para

11:  Retornar  $I_S$                                 ▷ La imagen segmentada
12: Fin Procedimiento

```

Por lo que el predicado se convierte en una comparación entre esta tolerancia y la medida de similitud entre los píxels.

Para hallar la complejidad de este algoritmo en el peor de los casos, supondremos que el conjunto inicial de semillas está compuesto por un solo elemento y que además la región a segmentar a partir de la única semilla es la imagen completa. Así, el tamaño de los datos es el número de píxels de la imagen, al cual denotaremos como n .

Al marcar los píxels que ya fueron evaluados, el algoritmo sólo evaluará un píxel una sola vez en todo el proceso, y además tomando en cuenta que la región la compone toda la imagen I , eso quiere decir que se evaluarán todos los píxels de la imagen una sola vez, por lo que podemos decir que el algoritmo de segmentación por crecimiento de regiones es $O(n)$ (ó de orden lineal).

3.2.4. Resultados

Para las pruebas realizadas con este algoritmo, hemos empleado como criterio de homogeneidad la similitud de tonos de gris, es decir la diferencia entre los tonos de gris de un píxel y sus vecinos no debe ser mayor que un valor umbral. El valor umbral fue configurado en 20.



Figura 3.6: Segmentación por crecimiento de regiones de una imagen general. El valor umbral fue configurado en 20

En la figura 3.6 se puede observar cómo el algoritmo de segmentación por crecimiento de regiones separa bien el fondo de los objetos presentes, pero ignora la parte del fondo que se encuentra encerrado por un objeto.

En las figuras 3.7, 3.8 y 3.9 se puede apreciar resultados un tanto más aceptables sobre regiones con menos variación entre los tonos de gris de sus píxels.

3.3. Segmentación Split-Merge

3.3.1. Descripción

El algoritmo de segmentación split-merge pretende resolver los inconvenientes que se tenían en la segmentación de crecimiento de regiones, en donde

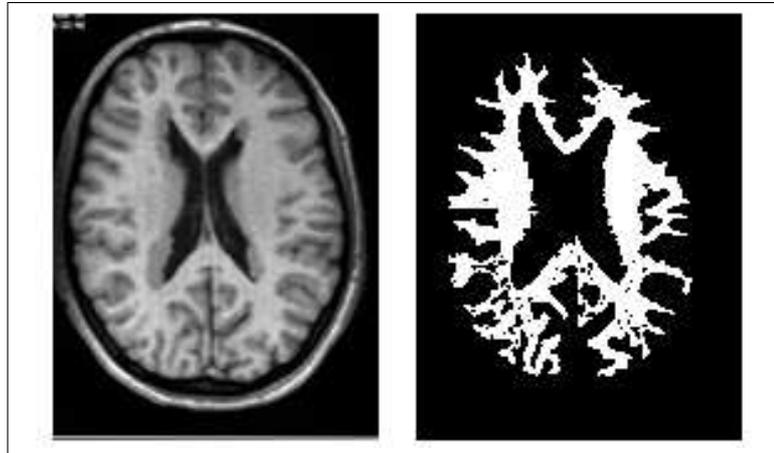


Figura 3.7: Segmentación por crecimiento de regiones de una imagen cerebral.



Figura 3.8: Segmentación por crecimiento de regiones de una imagen facial.

un simple criterio(predicado P) proponía hallar la región de interés, basándose en un conjunto inicial de puntos llamados semillas.

Este algoritmo usa una descomposición recursiva de la imagen tomando en cuenta el predicado que mide la homogeneidad de una región. Además se debe hacer uso de una estructura de datos adecuada para almacenar la información de la descomposición. La estructura de datos más empleada es el árbol cuaternario(también conocido como *quadtree*). Un árbol cuaternario permitirá llevar control de la descomposición de la imagen en cuadrantes(Fig. 3.10).

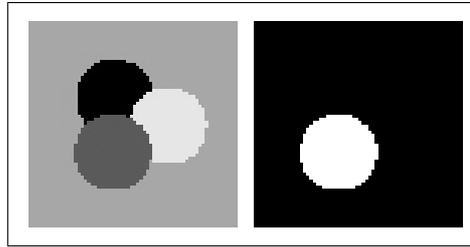


Figura 3.9: Segmentación por crecimiento de regiones de una imagen sintética.

Una vez que se ha descompuesto la imagen y se tiene una representación jerárquica de las regiones homogéneas de la imagen será necesario emplear un algoritmo de crecimiento de regiones para fusionar las regiones homogéneas vecinas que al fusionarlas sigan formando una región homogénea. Claramente se puede observar, que en todo momento se hará uso del predicado de homogeneidad P .

Es importante tener en cuenta que para que este método resulte efectivo, las imágenes sobre las que se aplica deben tener dimensiones equivalentes a cualquier potencia de 2. Así se puede controlar que el algoritmo realice las subdivisiones de regiones en cuadrantes del mismo tamaño.

3.3.2. Proceso

La forma como actúa este algoritmo se divide en dos fases. Primero se debe descomponer la imagen en regiones homogéneas que satisfagan el predicado de homogeneidad. Esto se realiza por medio de un algoritmo de descomposición cuyo resultado es un árbol cuaternario que representa la jerarquía de regiones homogéneas. El algoritmo de descomposición inicia considerando a la imagen completa como una región, a partir de la que se verifica si cumple con el predicado P . Cada vez que una región no cumpla con el predicado P , se divide en cuatro cuadrantes del mismo tamaño y se actualiza la información del árbol

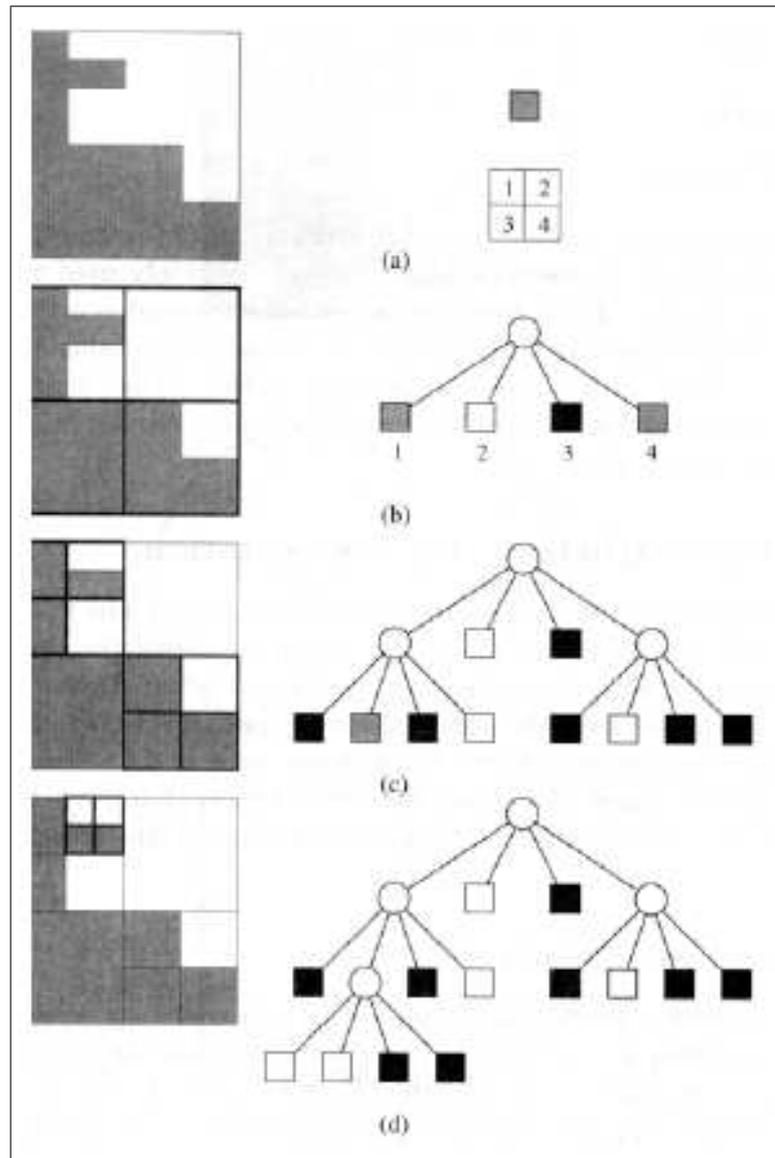


Figura 3.10: Diferentes niveles de descomposición representados por un quadtree.

cuaternario. Ahora, el algoritmo se ejecuta recursivamente sobre los nuevos nodos del árbol, sucesivamente hasta que no haya más regiones que evaluar.

Para completar el proceso de segmentación, la descomposición debe ser seguida por una fase de fusión. El problema radica en encontrar las regiones vecinas adyacentes a una región dada (nodo hoja del árbol cuaternario).

Para el proceso de fusión, se evalúa el criterio de homogeneidad para cada región junto con sus regiones adyacentes. Cuando dos regiones cumplen con el criterio, ellas se fusionan en una sola y se actualiza la información del árbol cuaternario.

3.3.3. Algoritmo

En esta sección se muestra el algoritmo de segmentación Split-Merge (Ver algoritmo 3).

Para hallar la complejidad de este algoritmo podemos identificar el tamaño de los datos como el número de filas que contiene la imagen, al fin y al cabo la otra dimensión es igual, ya que para que el algoritmo funcione las imágenes deben ser cuadradas.

En el peor de los casos, la imagen se subdividirá hasta que cada región quede representada por un solo píxel. Antes de que una imagen se divida, se evalúa el criterio de homogeneidad; el cual, generalmente, toma un tiempo lineal $O(n)$. Esto unido con el comportamiento de divisiones binarias que se realiza sobre las regiones hace que el algoritmo de descomposición sea $O(n \log n)$.

Para la fase de fusión tomaremos en cuenta el tamaño de los datos como el número de regiones que se pretenden fusionar. Podemos darnos cuenta que

Algoritmo 3 Algoritmo de Segmentación Split-Merge

Entrada: Imagen I , Predicado P **Salida:** Imagen segmentada I_S

- 1: **Procedimiento** SEGMENTACIÓNSPITMERGE(I, P)
 - 2: Iniciar el árbol con I como nodo raíz

 - 3: **Para cada** Región R_i iterativamente **Hacer**
 - 4: **Si** $P(R_i) = FALSO$ **Entonces**
 - 5: Descomponer R_i en 4 cuadrantes
 - 6: Actualizar el árbol cuaternario
 - 7: **Fin Si**
 - 8: **Fin Para**

 - 9: **Para cada** Regiones R_i, R_j adyacentes **Hacer**
 - 10: **Si** $P(R_i \cup R_j) = VERDADERO$ **Entonces**
 - 11: Fusionar R_i y R_j
 - 12: Actualizar el árbol cuaternario
 - 13: Repetir hasta que no hayan mas regiones que fusionar
 - 14: **Fin Si**
 - 15: **Fin Para**

 - 16: Reconstruir I_S desde el árbol cuaternario.
 - 17: **Retornar** I_S ▷ La imagen segmentada
 - 18: **Fin Procedimiento**
-

en el peor de los casos, cada región será un solo píxel, por lo que el tamaño de los datos de las dos fases es la misma, aunque no representen lo mismo.

Una manera óptima de llevar control de la adyacencia de las regiones, es construir un grafo de adyacencia representado por una matriz de adyacencia. Manipular la matriz para detectar que regiones pueden fusionarse está en el orden de $O(n^2)$, por lo que finalmente el algoritmo de segmentación split-merge está en el orden de $O(n^2)$.

3.3.4. Resultados

El algoritmo split-merge hace uso de dos umbrales para el criterio de homogeneidad, uno para cada fase del algoritmo. En los resultados mostrados aquí, hicimos uso de un valor umbral de 50 para la fase de descomposición y un umbral de 25 para la fase de fusión de regiones. La finalidad de emplear diferentes umbrales es que para tener un mejor nivel de detalle de las imágenes segmentadas, se tienen que generar la mayor cantidad de regiones descompuestas y hacer una fase de fusión restrictiva, así tendremos regiones con mejores detalles. Si se desea obtener imágenes con menos detalle, tendríamos que aplicar el proceso inverso.



Figura 3.11: Segmentación split-merge de una imagen general.

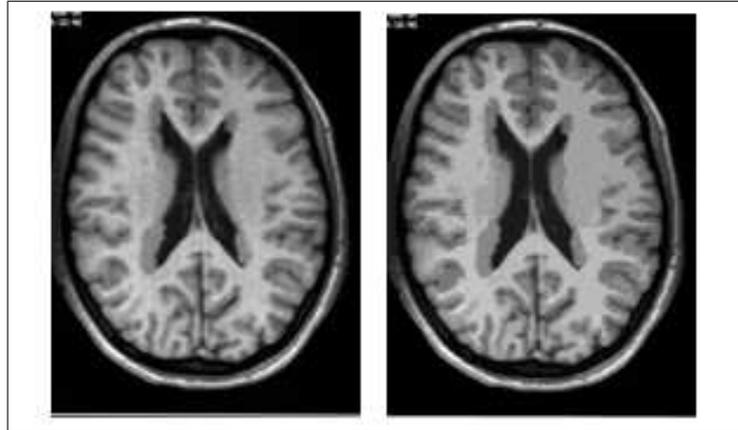


Figura 3.12: Segmentación split-merge de una imagen cerebral.

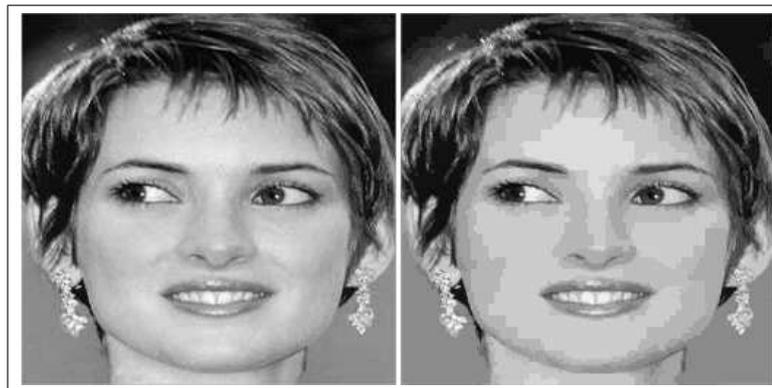


Figura 3.13: Segmentación split-merge de una imagen facial.

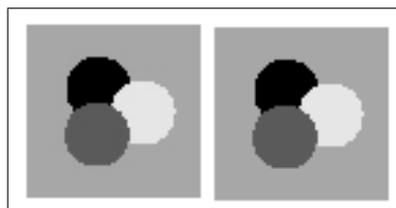


Figura 3.14: Segmentación split-merge de una imagen sintética.

De manera general, podemos apreciar que el algoritmo split-merge realiza una buena segmentación de las imágenes, tanto de las reales como para la imagen sintética mostrada. Sin embargo, tiende a producir regiones inexistentes debido en gran medida a factores como la iluminación y las sombras, como podemos apreciarlo en las figuras 3.13 y 3.14, es decir la imagen resultante se vuelve más complicada cuando lo que buscamos en realidad es obtener imágenes más simples, como bosquejos en donde cada región identifica un objeto.

Capítulo 4

El Funcional de Mumford-Shah

En este capítulo, haremos un análisis más profundo del funcional propuesto por Mumford y Shah en 1989. Para nuestro propósito será necesario citar otros problemas y definiciones que llevaron a los autores a formular dicha teoría y que más tarde serán útiles para comprender los problemas inherentes a este funcional de manera que podamos plantear una solución práctica.

Empezaremos por describir e interpretar los problemas originales que dieron pie a la formulación de Mumford y Shah, después nos avocaremos a analizar dicha formulación identificando los problemas que presenta y finalmente identificaremos el comportamiento de la solución del funcional para luego hacer uso de las herramientas necesarias para calcular su solución.

4.1. Problemas Inversos en Procesamiento de Imágenes

Una tarea importante en el procesamiento de imágenes es la reconstrucción de una imagen, en donde dada una imagen corrupta se pretende encontrar una imagen limpia y clara. Algunas de las tareas de reconstrucción

de imágenes más conocidas son la eliminación de ruido y la restauración de imágenes borrosas, todas estas pertenecen a la clase de problemas conocidos como *Problemas Inversos*. Esto significa que el proceso mediante el cual los datos son obtenidos desde las características físicas de la escena observada corresponden a transformaciones que son bien comprendidas y pueden ser más o menos modeladas matemáticamente, pero que el proceso inverso no es conocido o no puede ser calculado por métodos directos, lo que hace que la escena sea difícil de reconstruir.

Un modelo estándar de adquisición de imágenes está dado por:

$$g = Af + n, \quad (4.1)$$

donde g es la imagen corrupta, f es la imagen "perfecta", n es el ruido aditivo, el cual se asume que es aleatorio y con valor medio conocido y varianza σ^2 , A es un operador lineal que representa la influencia del sistema óptico. Típicamente, el efecto de A es suavizar, puede por ejemplo considerarse a A como un operador gaussiano.

Entonces la idea es encontrar f teniendo g y una estimación de A y σ^2 . La solución más obvia sería computar $A^{-1}g = f + A^{-1}n$, sin embargo esto no es factible en la práctica; ya que el operador A muchas veces no es invertible o su inversa es imposible de computar.

Un mejor enfoque para este tipo de problema es el siguiente: Intentar encontrar la "mejor" función f que satisfaga

$$\begin{cases} \int_{\Omega} Af - g = 0 \\ \int_{\Omega} |Af - g|^2 = \sigma^2 \end{cases} \quad (4.2)$$

Existen diversos enfoques para encontrar un buen criterio que caracterice a la "mejor" función f sujeto al cumplimiento de la ecuación anterior. Entre los enfoques más clásicos se encuentra el de Tichonov¹ y el criterio de la Variación Total² [7].

4.2. Origen del Enfoque Variacional para la Segmentación de Imágenes

Para hablar del enfoque variacional de Mumford y Shah es necesario remontarnos al enfoque estadístico propuesto por S. Geman y D. Geman y que trata acerca de regularizar correctamente el problema inverso en un ambiente discreto y restaurar correctamente los bordes de una imagen, todo ello aplicado al problema de la eliminación de ruido. Es así como el trabajo de Mumford y Shah trata de reformular este enfoque llevándolo a un ambiente continuo pero aplicado al problema de la segmentación de imágenes, es por eso que puede considerarse que la eliminación de ruido y la segmentación de imágenes tienen un origen común.

4.2.1. Una breve descripción del enfoque de Geman y Geman

La idea fundamental del enfoque de S.Geman y D.Geman [7, 14] es considerar la imagen observada G como una matriz $G = (g_{i,j})_{1 \leq i,j \leq n}$ con valores en tonos de gris en $[0, 1]$ y que es la combinación de una imagen desconocida F ,

¹El enfoque clásico de Tichonov consiste en minimizar alguna norma cuadrática tal como $\int_{\Omega} |f|^2$ o $\int_{\Omega} |\nabla f|^2$ bajo las restricciones expuestas.

²Rudin, Osher y Fatemi propusieron encontrar una función f que minimizara el funcional $\int_{\Omega} |\nabla f|$ sujeto a las restricciones expuestas.

donde $F = (f_{i,j})_{1 \leq i,j \leq n}$, y además un ruido aditivo gaussiano $N = (n_{i,j})_{1 \leq i,j \leq n}$. Cada $n_{i,j}$ son independientes y tiene una media de 0 y varianza σ^2 .

Ellos consideraron que la mayoría de imágenes son suaves por piezas³ y con posibles discontinuidades(bordes). Es por ello que introdujeron un conjunto de bordes o *line-process* L , donde $L = (l_{i+\frac{1}{2},j})_{1 \leq i < n, 1 \leq j \leq n}, (l_{i,j+\frac{1}{2}})_{1 \leq i \leq n, 1 \leq j < n}$ y la variable $l_{\alpha,\beta}$ es 0 o 1, dependiendo del cumplimiento de los siguientes criterios:

$$l_{i+\frac{1}{2},j} = \begin{cases} 1 & \text{Si existe una discontinuidad entre } (i, j) \text{ y } (i+1, j) \\ 0 & \text{Si U es suave entre } (i, j) \text{ y } (i+1, j) \end{cases} \quad (4.3)$$

$$l_{i,j+\frac{1}{2}} = \begin{cases} 1 & \text{Si existe una discontinuidad entre } (i, j) \text{ y } (i, j+1) \\ 0 & \text{Si U es suave entre } (i, j) \text{ y } (i, j+1) \end{cases} \quad (4.4)$$

Tomando en cuenta lo anterior, propusieron la siguiente ley de probabilidad para F,L:

$$P(F, L) = \frac{1}{Z} \exp\left\{- \sum_{i,j} (\lambda(1 - l_{i+\frac{1}{2},j})(f_{i+1,j} - f_{i,j})^2 + \mu l_{i+\frac{1}{2},j} + \lambda(1 - l_{i,j+\frac{1}{2}})(f_{i,j+1} - f_{i,j})^2 + \mu l_{i,j+\frac{1}{2}})\right\}, \quad (4.5)$$

donde los parámetros λ y μ son dos pesos positivos y Z es calculado para obtener $\sum_{F,L} P(F, L) = 1$.

³Una función(imagen) es suave por piezas o "piecewise" si para cada intervalo abierto disjunto, donde cada intervalo determina una región, la función es constante

Ahora, nos encontramos con un problema y es saber de entre todas las posibles imágenes F y L line-process existentes, cuales son aquellas que tienen la más alta probabilidad $P(F, L|G)$.

$$P(F, L|G) \sim e^{-E(F,L,G)}, \quad (4.6)$$

donde la energía libre $E(F, L, G)$ está dada por:

$$\begin{aligned} E(F, L, G) = & \sum_{i,j} \lambda((1 - l_{i+\frac{1}{2},j})(f_{i+1,j} - f_{i,j})^2 + (1 - l_{i,j+\frac{1}{2}})(f_{i,j+1} - f_{i,j})^2) \\ & + \mu(l_{i+\frac{1}{2},j} + l_{i,j+\frac{1}{2}}) \\ & + \frac{1}{2\sigma^2}(g_{i,j} - f_{i,j})^2, \end{aligned} \quad (4.7)$$

donde G corresponde a los datos dados y por consiguiente puede ser obviado en la notación quedando sólo la energía en función de F y L , es decir $E(F, L)$.

4.3. El funcional de Mumford-Shah

La propuesta de Mumford y Shah [22] consiste en reformular la ecuación (4.7) en un ambiente continuo. Entonces ellos consideraron una imagen observada $g(x, y) \in [0, 1]$ donde $(x, y) \in \Omega$, siendo Ω un subconjunto abierto acotado de R^2 , es decir Ω constituye el dominio de la imagen.

También consideraron descomponer el dominio Ω en diversas regiones R_i y establecer a Γ como el conjunto de posibles discontinuidades entre las regiones R_i .

$$\Omega = R_1 \cup R_2 \cup R_3 \dots \cup R_n \cup \Gamma, \quad (4.8)$$

Además ellos notaron que la variable L (enfoque de Geman y Geman) describe las discontinuidades o conjunto salto $\Gamma \subset \Omega$ de una función regular por piezas $f(x, y)$ donde $(x, y) \in \Omega$, mientras las diferencias finitas entre $f_{i+1,j} - f_{i,j}$ y $f_{i,j+1} - f_{i,j}$ son aproximaciones de las derivadas parciales $\frac{\partial f}{\partial x}(x, y)$ y $\frac{\partial f}{\partial y}(x, y)$ respectivamente [7]. Entonces la energía que ellos escribieron usa la notación estándar $\nabla f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$. De esta manera Mumford y Shah propusieron el siguiente funcional:

$$E(f, \Gamma) = \nu \int_{\Omega \setminus \Gamma} (f - g)^2 + \lambda \int_{\Omega \setminus \Gamma} |\nabla f|^2 + \mu \cdot |\Gamma| \quad (4.9)$$

donde Ω es el dominio, g es una imagen original en escala de grises, Γ denota el conjunto de bordes, f es la imagen suavizada pero discontinua a lo largo de Γ , $|\Gamma|$ es la longitud del conjunto de bordes y finalmente ν , λ y μ son pesos que varían de acuerdo a la fuerza que se quiera imprimir a cada término dependiendo de la aplicación que se desea dar al funcional.

La propuesta de Mumford-Shah radica en la minimización del funcional $E(f, \Gamma)$ de la ecuación(4.9), que está compuesto de tres términos que pueden interpretarse de la siguiente manera [22, 32, 6]:

- El término $\int_{\Omega \setminus \Gamma} (f - g)^2$, indica el nivel de fidelidad que controla que tanto se aproxima la imagen suave f a la imagen original g .
- El término de suavidad $\int_{\Omega \setminus \Gamma} |\nabla f|^2$, el cual debe ser pequeño si f cambia lentamente en las regiones.
- Finalmente el término $|\Gamma|$, la longitud del conjunto de bordes y que debe ser pequeño para prevenir que los bordes ocupen toda la imagen.

Cada uno de estos tres términos está relacionado a aspectos importantes que deben considerarse al segmentar una imagen, si eliminamos cualquiera de los términos del funcional entonces $\inf E = 0$.

- Si suprimimos el primer término, el $\inf E = 0$ se cumple tomando $f = 0$ y $\Gamma = \emptyset$, entonces la imagen segmentada sería como una imagen completamente gris.
- Si suprimimos el segundo término, el $\inf E = 0$ se logra haciendo $f = g$ y $\Gamma = \emptyset$.
- Si eliminamos el tercer término, cuando se toma Γ (bordes) como una malla fina de N líneas verticales y horizontales, toda la imagen podría estar compuesta de bordes.

Mumford y Shah conjeturaron que existe un minimizador para E , tal que los bordes son la unión de un conjunto finito de curvas $C^{1,1}$.

4.3.1. Análisis del Comportamiento de la Solución del Funcional

Es necesario realizar el análisis del comportamiento de la solución del funcional debido a que nos permitirá decidir la mejor manera de solucionarlo numéricamente. Primero reemplazaremos en el funcional original (4.9), el término de longitud de bordes por una integral equivalente a lo largo del conjunto Γ . Además por motivos de simplicidad en las demostraciones, obviaremos los parámetros ν , λ y μ .

$$E(f, \Gamma) = \int_{\Omega \setminus \Gamma} (f - g)^2 + \int_{\Omega \setminus \Gamma} |\nabla f|^2 + \int_{\Gamma} d\sigma \quad (4.10)$$

Supongamos que existe un par (f, Γ) , una solución de (4.10). Más aún, suponemos que (f, Γ) satisface las conjeturas de Mumford y Shah:

- (C_1) Γ consiste de un número finito de curvas $C^{1,1}$ γ_i , que llegan a $\partial\Omega$ y que llegan a otras sólo en sus puntos finales.
- (C_2) f es C^1 sobre cada componente conexa de $\Omega - \Gamma$.

Teorema 4.3.1. *Sea (f, Γ) una solución del problema (4.10) que satisface (C_1) y (C_2) . Entonces*

$$\Delta f = f - g \text{ sobre } \Omega, \quad (4.11)$$

$$\frac{\partial f}{\partial N} = 0 \text{ sobre } \partial\Omega \text{ y a ambos lados } \gamma_i^\pm \text{ de cada } \gamma_i, \quad (4.12)$$

$$e(f^+) - e(f^-) + \text{curv}\gamma_i = 0 \text{ sobre } \gamma_i. \quad (4.13)$$

donde $e(f) = (f - g)^2 + |\nabla f|^2$, f^+ y f^- son las trazas de f a cada lado de Γ (cada lado de γ_i), $\text{curv}\gamma_i$ es la curvatura de γ_i .

Demostración. La prueba de (4.11) y (4.12) es estándar. Primero veremos la variación de E con respecto a f . En (4.14), nosotros escogemos $\Gamma' = \Gamma$ y $v = f + \theta\varphi$ con $\theta \in R$ y φ es una función de prueba con soporte compacto. Entonces

$$\begin{aligned} 0 \leq E(f + \theta\varphi, \Gamma) - E(f, \Gamma) &= \theta^2 \int_{\Omega \setminus \Gamma} (\varphi^2 + |\nabla\varphi|^2) dx dy \\ &\quad + 2\theta \int_{\Omega \setminus \Gamma} (\varphi(f - g) + \nabla\varphi \cdot \nabla f) dx dy. \end{aligned} \quad (4.14)$$

Dividiendo (4.14) entre $\theta > 0$ y haciendo $\theta \rightarrow 0^+$ obtenemos

$$0 = \int_{\Omega \setminus \Gamma} \varphi(f - g) dx dy + \int_{\Omega \setminus \Gamma} \nabla\varphi \cdot \nabla f dx dy \quad (4.15)$$

Escogiendo φ con soporte compacto en $\Omega \setminus \Gamma$ y aplicando la fórmula de Green en (4.15) obtenemos

$$0 = \int_{\Omega \setminus \Gamma} \varphi(f - g - \Delta f) dx dy$$

es decir

$$f - g - \Delta f = 0, \text{ sobre } \Omega \setminus \Gamma$$

La prueba de (4.12) es consecuencia directa de la aplicación de la fórmula de Green en la prueba anterior. A (4.12) también se le conoce como Condición de Frontera de Neumann.

Para probar (4.13), la idea es evaluar la variación de E con respecto a Γ . Nosotros proponemos una prueba diferente a la dada por Mumford y Shah en su teoría original. Los argumentos que vamos a usar serán de mucha utilidad para la generación de una solución en los capítulos posteriores.

Sea Ω_{int} el conjunto abierto encerrado por Γ y $\Omega_{ext} = \Omega - \Omega_{int} - \Gamma$. Nuestro objetivo es considerar variaciones de Γ de acuerdo al flujo $\frac{d(x,y)}{dt} = v(x, y, t)$, donde v es una velocidad arbitraria. Nosotros denotamos como $\Gamma(t)$ tal variación, $t \geq 0$, con $\Gamma(0) = \Gamma$. Desde que f varía con el movimiento de Γ , nosotros denotamos por $f(x, y, t)$ la solución única de $\inf_f E(f, \Gamma(t))$ y $f_{int}(x, y, t) = f(x, y, t)|_{\Omega_{int}}$, $f_{ext}(x, y, t) = f(x, y, t)|_{\Omega_{ext}}$.

Tenemos entonces

$$J(t) = \int_{\Omega - \Gamma(t)} [(f(x, y, t) - g(x, y))^2 + |\nabla f(x, y, t)|^2] dx dy + \int_{\Gamma(t)} d\sigma.$$

Escribiendo $\Omega = \Omega_{int}(t) \cup \Omega_{ext}(t) \cup \Gamma(t)$, tenemos

$$J(t) = \int_{\Omega_{int}-\Gamma(t)} [(f_{int}(x, y, t) - g(x, y))^2 + |\nabla f_{int}(x, y, t)|^2] dx dy \\ + \int_{\Omega_{ext}-\Gamma(t)} [(f_{ext}(x, y, t) - g(x, y))^2 + |\nabla f_{ext}(x, y, t)|^2] dx dy + \int_{\Gamma(t)} d\sigma$$

Podemos apreciar que en la expresion de $J(t)$ tanto el dominio de integración como los integrandos dependen de t . Como estamos interesados en la primera variación de E , necesitamos estimar $J'(t)$.

- Para los primeros dos integrales necesitamos usar un resultado clásico sobre la *derivativa sobre un dominio integral*: Si $l(x, y, t)$ es una función regular definida sobre un dominio regular acotado $w(t)$ de R^N y si tenemos

$$g(t) = \int_{w(t)} l(x, y, t) dx dy,$$

entonces

$$g'(t) = \int_{w(t)} \frac{\partial l}{\partial t}(x, y, t) dx dy + \int_{\partial w(t)} l(x, y, t) v \cdot N d\sigma,$$

donde $\partial w(t)$ es la frontera de $w(t)$, N es la normal unitaria hacia afuera a $\partial w(t)$, y v es la velocidad de $\partial w(t)$.

- Considerando el último término, necesitamos conocer como estimar la derivativa de la longitud. Podemos mostrar que

$$\frac{d}{dt} \left(\int_{\Gamma(t)} d\sigma \right) = \int_{\Gamma(t)} \text{curv} \Gamma(t) v \cdot N d\sigma.$$

Aplicando los resultados de arriba tenemos

$$\begin{aligned}
J'(t) &= 2 \int_{\Omega_{int}(t)} (f_{int} - g) \frac{\partial f_{int}}{\partial t} dx dy + \int_{\Gamma(t)} (f_{int} - g)^2 v \cdot N d\sigma \\
&+ 2 \int_{\Omega_{int}(t)} \nabla f_{int} \cdot \nabla \left(\frac{\partial f_{int}}{\partial t} \right) dx dy + \int_{\Omega(t)} |\nabla f_{int}|^2 v \cdot N d\sigma \\
&+ 2 \int_{\Omega_{ext}(t)} (f_{ext} - g) \frac{\partial f_{ext}}{\partial t} dx dy - \int_{\Gamma(t)} (f_{ext} - g)^2 v \cdot N d\sigma \\
&+ 2 \int_{\Omega_{ext}(t)} \nabla f_{ext} \cdot \nabla \left(\frac{\partial f_{ext}}{\partial t} \right) dx dy - \int_{\Omega(t)} |\nabla f_{ext}|^2 v \cdot N d\sigma \\
&+ \int_{\Gamma(t)} \text{curv} \Gamma(t) v \cdot N d\sigma.
\end{aligned} \tag{4.16}$$

Gracias a la fórmula de Green tenemos

$$\int_{\Omega_{int}(t)} \nabla f_{int} \cdot \nabla \left(\frac{\partial f_{int}}{\partial t} \right) dx dy = - \int_{\Omega_{int}} \Delta f_{int} \frac{\partial f_{int}}{\partial t} dx dy + \int_{\Gamma(t)} \frac{\partial f_{int}}{\partial t} \frac{\partial f_{int}}{\partial N} d\sigma,$$

pero $f_{int}(t, x)$ es la solución de

$$\begin{cases} \Delta f_{int}(x, y, t) = f_{int}(x, y, t) - g(x, y) \text{ en } \Omega_{int}(t), \\ \frac{\partial f_{int}}{\partial N} = 0 \text{ sobre } \Gamma(t). \end{cases}$$

Entonces

$$\int_{\Omega_{int}(t)} \nabla f_{int} \cdot \nabla \left(\frac{\partial f_{int}}{\partial t} \right) dx dy = - \int_{\Omega_{int}(t)} (f_{int}(x, y, t) - g(x, y)) \frac{\partial f_{int}}{\partial t}(x, y, t) dx dy.$$

Lo mismo sucede para

$$\int_{\Omega_{ext}(t)} \nabla f_{ext} \cdot \nabla \left(\frac{\partial f_{ext}}{\partial t} \right) dx dy = - \int_{\Omega_{ext}(t)} (f_{ext}(x, y, t) - g(x, y)) \frac{\partial f_{ext}}{\partial t}(x, y, t) dx dy.$$

Por lo tanto, reemplazando estas últimas expresiones en (4.16) obtenemos

$$\begin{aligned} J'(t) &= \int_{\Gamma(t)} ((f_{int} - g)^2 + |\nabla f_{int}|^2)v \cdot N d\sigma \\ &\quad - \int_{\Gamma(t)} ((f_{ext} - g)^2 + |\nabla f_{ext}|^2)v \cdot N d\sigma \\ &\quad + \int_{\Gamma(t)} \text{curv}\Gamma(t)v \cdot N d\sigma \end{aligned}$$

o, con la notación del Teorema 4.3.1,

$$J'(t) = \int_{\Gamma(t)} (e(f_{int}) - e(f_{ext}) + \text{curv}\Gamma(t))v \cdot N d\sigma.$$

Hasta ahora no hemos especificado la variación de $\Gamma(t)$. Haremos que Γ se mueva a lo largo de su normal hacia afuera de acuerdo a la siguiente ecuación diferencial:

$$\frac{\partial(x, y)}{\partial t} = v(x, y, t) = V(x(t), y(t))N,$$

$$(x, y)(0) = \Gamma,$$

donde V es una velocidad arbitraria. Desde que $|N|^2 = 1$, $J'(t)$ puede ser escrita como

$$J'(t) = \int_{\Gamma(t)} (e(f_{int}(x, y, t)) - e(f_{ext}(x, y, t)) + \text{curv}\Gamma(t))V(x(t), y(t))d\sigma.$$

Si (f, Γ) es un minimizador del funcional de Mumford-Shah, necesariamente tenemos que $J'(0) = 0$, es decir,

$$0 = \int_{\Gamma} (e(f_{int}(x, y)) - e(f_{ext}(x, y)) + \text{curv}\Gamma(t))V(x(t), y(t))d\sigma,$$

y como V es arbitrario, obtenemos,

$$e(f_{int}) - e(f_{ext}) + \text{curv}\Gamma = 0 \text{ sobre } \Gamma.$$

□

El teorema anterior nos ha permitido descubrir una característica importante del funcional Mumford-Shah. El funcional Mumford-Shah involucra la medición de la curvatura como cantidad geométrica, hecho que nos permitirá, posteriormente, representar este funcional adecuadamente por conjuntos de niveles.

4.3.2. Problema de Partición Mínimo

El funcional (4.9) propuesto por Mumford y Shah considera que f es una aproximación suave por piezas de la imagen dada g , es decir, f varía suavemente dentro de las regiones de los objetos. Sin embargo, Mumford y Shah [22] también propusieron un caso límite del funcional (4.9):

$$E(f, \Gamma) = \nu \int_{\Omega \setminus \Gamma} (f - g)^2 + \mu \cdot |\Gamma|, \quad (4.17)$$

donde se considera a f una aproximación constante por piezas de g , es decir, $f = \text{constante } a_i$ sobre cada conjunto abierto R_i . En otras palabras, f ya no es suave dentro de cada región que compone a un objeto, sino que toda la región tiene un valor de intensidad constante. Es más, Mumford y Shah llegaron a deducir que $a_i = \text{media}_{R_i} g$.

Mumford y Shah denominaron *Problema de Partición Mínimo* a este caso restrictivo del funcional original.

Nosotros emplearemos el *Problema de Partición Mínimo* para plantear un algoritmo para segmentación de imágenes.

Capítulo 5

Método de Conjuntos de Nivel

En este capítulo ponemos a disposición del lector la teoría de fondo que acompaña a uno de los esquemas numéricos más recientes y robustos que existen, ya que a superado con creces las expectativas con las que fue ideado por S.Osher y J.Sethian en 1987, tanto así que ya en Mayo del 2002 realizando una tarea de búsqueda sobre Google, esta máquina de búsqueda arrojó 2800 respuestas acerca del tema y el artículo original de su teoría ha sido citado más de 530 veces.

Los conjuntos de nivel, proponen controlar el movimiento de *interfaces* o *curvas* evolutivas (también llamados *fronts*) para detectar los bordes correspondientes a alguna geometría. Puede tratarse de bordes dinámicos como las olas rompiéndose en el océano, llamas de fuego, las ondas hechas por la leche en una taza o bordes estáticos si se trata de imágenes médicas acerca de tumores, o si se busca encontrar la figura de un dalmata en una alfombra de puntos. Este método incluso puede aplicarse para dar solución a problemas cotidianos como tratar de trasladar un piano a través de un departamento estrecho y con obstáculos o el de intentar encontrar el camino más corto por una cordillera. Ver figura 5.1.

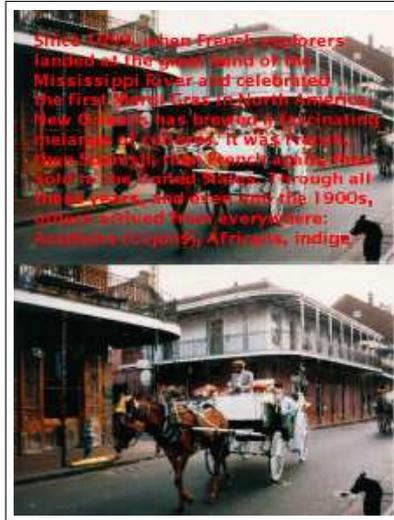


Figura 5.2: Figura que muestra la sustracción del texto sobre la imagen, para dejar la imagen pura.

de ondas, simulación de humo, logrados con gran realismo y que resultan sumamente útiles en la industria de la producción cinematográfica, como se aprecia en las figuras 5.3 y 5.4.

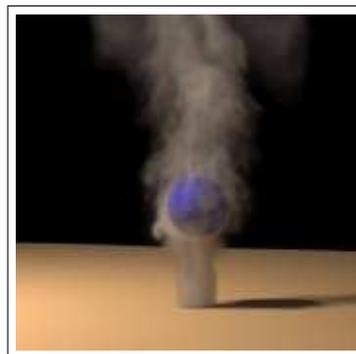


Figura 5.3: Figura que muestra la simulación de humo.

En las siguientes líneas hemos querido sintetizar toda la información útil para poder comprender el funcionamiento de este método y describirlo desde un punto de vista analítico, pero usando al mismo tiempo un lenguaje didáctico, con gráficos y ejemplos.



Figura 5.4: Figura que muestra a la izquierda la simulación de la caída de una piedra en un depósito de agua. A la derecha dos troncos de madera quemándose.

5.1. Introducción a los Conjuntos de Nivel

El rol de los conjuntos de nivel en el procesamiento de imágenes se relaciona frecuentemente a las ecuaciones diferenciales parciales. Usualmente los conjuntos de nivel utilizan algún método EDP, el cual debe cumplir con algunas características: 1) Asegurar la Regularidad de las soluciones, 2) Representar adecuadamente las fronteras y 3) Ofrecer desarrollos numéricos para los conjuntos de nivel.

Los conjuntos de nivel sirven para particionar el dominio de la imagen en diferentes regiones. Las interfaces que separan dichas regiones representan el conjunto de nivel cero. Estas interfaces son representadas por funciones adecuadas, por lo menos una función Lipschitz continua, usualmente la función de distancia con signo, cuando se trata de curvas bidimensionales como es el caso de las imágenes.

Mientras el movimiento de estas interfaces es controlado por EDP's, en otros casos, el movimiento de las interfaces depende de la posición, el tiempo, la geometría de la interfaz(su curvatura o la dirección de su normal) y fenómenos físicos externos, por tanto podemos decir que este método es conveniente para

la representación de funcionales que involucran cantidades geométricas tales como la curvatura.

Este enfoque actúa trasladando el problema a solucionar a una dimensión más alta, es decir en el caso de las imágenes que son bidimensionales, la solución es trabajada en el espacio tridimensional en donde las interfaces son superficies cerradas, desafortunadamente la alternativa tradicional conocida como una parametrización intrínseca podría no ser conveniente si se desea soportar esquinas afinadas y cambios topológicos (por ejemplo cuando se queman dos llamas de fuego que terminan por fundirse en una), por este motivo para nuestros propósitos computacionales se debe reformar el problema en un espacio dimensional más alto, donde paradójicamente un sistema de coordenadas ofrece la solución, por eso alguna vez Hermann Weyl, un importante matemático dijo *"La introducción de un sistema de coordenadas para una geometría es un acto de violencia.."* [30].

Finalmente, podemos concluir diciendo que el método de conjuntos de nivel ayuda a analizar y manipular los conjuntos de niveles de una función continua que podría ser o no una imagen.

5.2. Conceptos Previos

El método de conjuntos de nivel puede definirse como una herramienta matemática y computacional para seguir el movimiento de interfaces o curvas evolutivas, capaz de soportar la formación de esquinas y cúspides, cambios topológicos y complicaciones tridimensionales.

Dicho formalmente, dada una interfaz Γ_t en R^n acotando una región abierta Ω en un tiempo t , es posible analizar el movimiento de Γ en un campo de velocidad $\mathbf{v} = (v_1, v_2, \dots, v_n)$.

La velocidad v no es más que un movimiento en dirección perpendicular (Ver figura 5.5) a la interfaz dependiente además de una serie de factores o propiedades de la interfaz de tal forma que puede formalizarse en una función como esta $v = v(L, G, I)$, donde:

- L = Propiedades Locales, están determinadas por información geométrica, como la curvatura y la dirección de la normal.
- G = Propiedades Globales, depende de la forma y posición de la interfaz.
- I = Propiedades Independientes, precisamente son independientes de la forma y tienen que ver por ejemplo con la velocidad de fluido que sirve para transportar la interfaz [31].

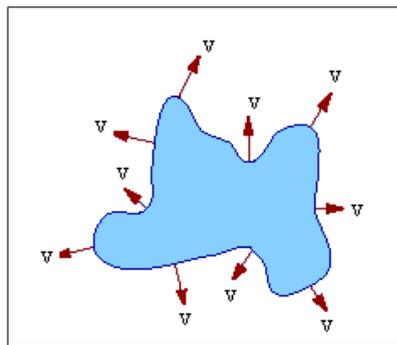


Figura 5.5: Figura que muestra la velocidad como el movimiento perpendicular a la interfaz.

Otra definición básica es la de la curvatura k que se define como la rapidez con la que una interfaz Γ se dobla en cualquiera de sus puntos.

Por ejemplo un círculo tiene una curvatura constante ya que siempre se dobla en una misma proporción sobre cada uno de sus puntos, pero si suponemos un círculo más pequeño, éste tiende a tener una curvatura también constante pero mayor, pues se dobla más rápido aún.

Algunos puntos a considerar son por ejemplo restringir el movimiento de cada parte de una curva siguiendo sólo la dirección de su normal con una velocidad v (proporcional a su curvatura k) e ignorando el movimiento de la interfaz en su dirección tangencial. Sabiendo que la curvatura k puede ser positiva o negativa (si lleva sentido horario o antihorario, respectivamente), obviamente algunas partes de la curva pueden moverse hacia adentro y otras hacia afuera, ya que se mueven siguiendo a su normal (Ver figura 5.6).

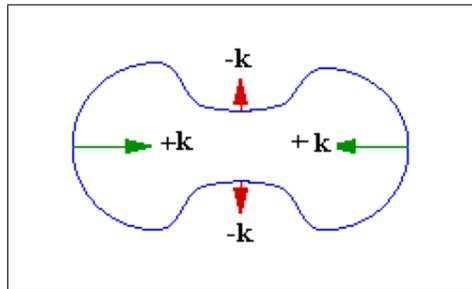


Figura 5.6: En la siguiente figura, se aprecia los cambios de la curvatura en las zonas convexas y cóncavas de la interfaz.

También podemos deducir, según la figura, que las flechas más largas indican una magnitud más alta que las flechas cortas, por tanto la curvatura es más alta en las zonas convexas.

Más aún, si seguimos tratando de deducir más aspectos, deberíamos preguntarnos ¿Qué sucede si la curva sigue moviéndose de acuerdo a su curvatura?. Si la curva inicial es un círculo y sus puntos se mueven hacia el centro, entonces

el círculo tiende a decrecer y a colapsar como un único punto. Pero, si evaluamos una curva con forma relajada y suave, también tiende a hacerse circular, como lo podemos apreciar en la figura 5.7. Entonces podemos generalizar este efecto para cualquier curva simple cerrada (Ver figura 5.8) [30].

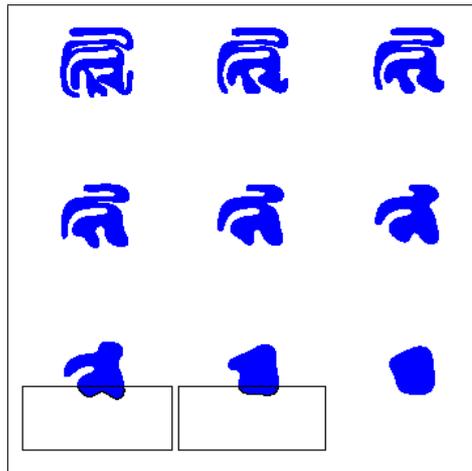


Figura 5.7: La figura muestra que el movimiento por curvatura termina por hacer colapsar cualquier curva simple cerrada a un único punto.

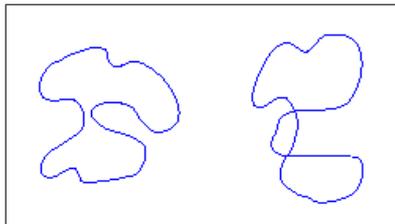


Figura 5.8: A la izquierda una curva simple cerrada. A la derecha está no es una curva simple cerrada, debido a que se interseca a sí misma.

5.2.1. Representación de Conjuntos de Nivel

Dado que, las representaciones clásicas que existen (funcional y paramétrica) han sido desechadas por considerarse no convenientes para construir un modelo computacional como el de los conjuntos de nivel, primero porque no

permiten fusionar las interfaces; y segundo porque limitan la traslación del problema a una dimensión mayor (introducción sistema de coordenadas), es necesario plantear una representación adecuada.

Estando en condiciones de introducir un sistema de coordenadas, usando el plano xy el cual contiene a la interfaz, añadiremos una coordenada más z que mide la altura.

Sea una función $z = \phi(x, y, t_0)$, donde al punto (x, y) le asignamos una altura z . Inicialmente si consideramos que la distancia desde el punto (x, y) a la interfaz en un tiempo t_0 , dicho valor corresponde a la altura respectiva z .

El propósito de construir esta superficie, es lograr intersectar el plano xy con la interfaz. En la figura 5.9, observamos una superficie, la cual es llamada función de conjunto de nivel, ésta acepta como entrada un punto en el plano xy y arroja como salida una altura z . La interfaz plana, es llamada el conjunto de nivel cero, porque ésta es una colección de todos los puntos que tienen como altura cero, $z = 0$.

La superficie descrita es una superficie de nivel, que puede ser entendida por ejemplo, si hacemos un corte extrayendo una lámina de dicha superficie, y la dejamos caer sobre el plano xy . Sin embargo, la lámina posee un nivel (Ver figura 5.9).

- Si cortamos la superficie del conjunto de nivel a una altura cero arriba del plano xy , la lámina caería sobre el plano xy que podría ser la interfaz original.
- Si cortamos la superficie del conjunto de nivel a otra altura, la lámina caería bajo el plano xy , produciendo cualquiera de las otras curvas .

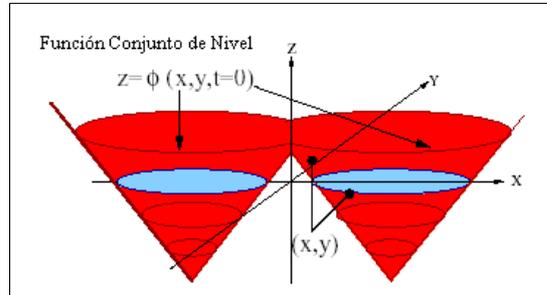


Figura 5.9: La superficie de Conjunto de Nivel (en rojo) dibuja la distancia de cada punto (x, y) a la interfaz (en azul).

Otra forma más sencilla de ver y entender una función de conjunto de nivel ϕ es imaginarla como un mapa topográfico con superficies elevadas afuera y con una laguna dentro, cuyo nivel corresponde al borde(s) de nuestra interfaz. Pasado un tiempo la altura de la superficie $\phi(x, y, t)$ cambia (Ver figura 5.10), lo cual representa justamente la evolución de la interfaz, después que la función termine de evolucionar, para obtener una función de conjunto de nivel basta con cortar la superficie a la altura cero (u otra), en otras palabras mostrar el contorno cero (u otro).

Seguramente, el hecho de emular el mismo proceso de mover una curva para mover una superficie, puede sonar extraño, fácilmente podemos darnos cuenta que problemas de dimensiones más altas, significan más trabajo.

Finalmente, la razón de la dimensión extra, es que estamos en condiciones de posicionar cada punto (x, y) y ajustar la altura de la función del conjunto de nivel. Esto justifica, el soporte de los problemas debido a cambios topológicos, tales como la fusión o separación de dos interaces [30].

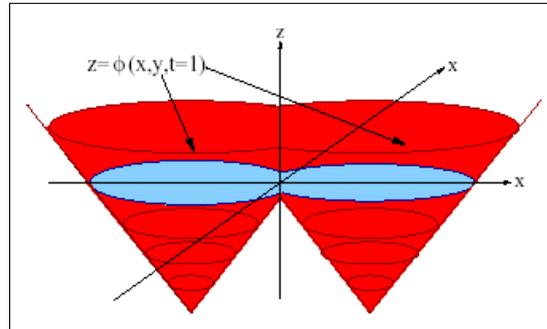


Figura 5.10: La superficie de Conjunto de Nivel(en rojo)fue movida, produciendo una nueva interfaz(en azul).

5.3. Formulación de los Conjuntos de Nivel

Habiendo revisado ya algunas definiciones básicas y sabiendo que la curvatura es una componente de muchos fenómenos físicos, el modelo computacional que a continuación describimos se sustenta considerando lo que sucede en una interfaz evolutiva moviéndose con respecto a su curvatura.

Dada una hipersuperficie $(N - 1)$ -dimensional cerrada Γ_0 , produciremos una formulación Euleriana para controlar el movimiento de la hipersuperficie Γ propagándose a lo largo de su normal con una velocidad v .

Dicha hipersuperficie es la interfaz y está dada por una función suave $\phi(\mathbf{x}, t)$ representando la interfaz como un conjunto donde $\phi(\mathbf{x}, t) = 0$ (con un nivel cero que representa el borde de la interfaz) y $\mathbf{x} = (x_1, x_2, \dots, x_n) \in R^n$. Es decir, $\phi(\mathbf{x}, t)$ es la función de conjunto de nivel concebida como curvas cerradas en dos o más dimensiones y que sirven para dividir un dominio en regiones [31] [24].

Entonces expresando la normal \mathbf{N} de la interfaz Γ_t asumida como una función de conjuntos de nivel, tendremos:

$$\mathbf{N} = \frac{\nabla\phi}{|\nabla\phi|}, \quad (5.1)$$

Si bien se dice que la hipersuperficie Γ se mueve siguiendo la dirección de la normal de cada uno de los puntos que la componen, debemos considerar que es la curvatura la que realmente determina como se mueven estos puntos, es decir, si lo hacen hacia afuera o hacia adentro de la interfaz, como ya lo hemos explicado en una sección anterior. Entonces es bueno aquí hacer una definición más formal de la curvatura, la cual se expresa como un vector de puntos en dirección a la normal de la función de conjuntos de nivel.

$$k = -\nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} \quad (5.2)$$

Además sea la región Ω_t abierta acotada por $\Gamma_t : \{\mathbf{x} | \phi(\mathbf{x}, t) < 0\}$ define la parte interior de la interfaz, y $\Gamma_t : \{\mathbf{x} | \phi(\mathbf{x}, t) > 0\}$ la parte exterior.

La función de conjunto de nivel ϕ tiene las siguientes propiedades [23]:

- $\phi(\mathbf{x}, t) < 0$ para $\mathbf{x} \in \Omega$
- $\phi(\mathbf{x}, t) > 0$ para $\mathbf{x} \notin \bar{\Omega}$
- $\phi(\mathbf{x}, t) = 0$ para $\mathbf{x} \in \partial\Omega = \Gamma(t)$,

sabiendo que la idea es hacer pasar la interfaz de propagación como una función de conjunto de nivel cero. La función ϕ puede ser perfectamente una función de distancia con signo, como esta:

$$\phi(\mathbf{x}, t_0) = \bar{d}(\mathbf{x}, \Gamma_0), \quad (5.3)$$

donde \bar{d} es la distancia de \mathbf{x} a la interfaz Γ_0 (distancia Euclídeana) y el signo \pm depende de si el punto \mathbf{x} se encuentra afuera o adentro, respectivamente de la hipersuperficie Γ_0 [4].

$$\bar{d} = \begin{cases} +d(\mathbf{x}, \Gamma_0) & \text{Si } \mathbf{x} \text{ está fuera de } \Gamma_0, \\ -d(\mathbf{x}, \Gamma_0) & \text{Si } \mathbf{x} \text{ está dentro de } \Gamma_0. \end{cases} \quad (5.4)$$

Ahora tenemos que $\phi(\mathbf{x}, t_0) : R^2 \longrightarrow R$ con la propiedad de que:

$$\Gamma_0 = [\mathbf{x} | \phi(\mathbf{x}, t_0) = 0], \quad (5.5)$$

Lo siguiente es introducir una ecuación evolutiva $\phi(\mathbf{x}, t)$ que incluya el movimiento de Γ_t cuando el conjunto de nivel $\phi = 0$. Para lograr esto, necesitamos que la distancia entre \mathbf{x}_t y la interfaz de propagación Γ_t sea cero. Es decir, \mathbf{x}_0 es un punto sobre la interfaz inicial Γ_0 , y además $\nu_{\mathbf{x}_t} = N$, donde N naturalmente es la normal.

Además sabiendo que, la función del conjunto de nivel cero debe corresponderse con la interfaz, eso significa que:

$$\phi(\mathbf{x}_t, t) = 0, \quad (5.6)$$

Aplicando la regla de la cadena para la ecuación anterior, tenemos:

$$\phi_t + \nabla\phi(\mathbf{x}_t, t) \cdot \mathbf{x}'_t = 0, \quad (5.7)$$

Desde que v indica la velocidad en dirección de la normal hacia afuera, entonces $v = N \cdot \mathbf{x}'_t$, donde $N = \frac{\nabla\phi}{|\nabla\phi|}$ y esto produce una ecuación de evolución

para ϕ , esta es:

$$\phi_t + v|\nabla\phi| = 0, \quad (5.8)$$

dado un $\phi(\mathbf{x}, t_0)$ [31] [24].

Pero la ecuación de arriba no es el modelo final. Consideremos un $t \geq 0$ y $\mathbf{x} \in \Omega$ en tanto v este definido fuera de la interfaz, en el espacio completo y tomando en cuenta estas condiciones:

- Una condición de bordes: una elección general es que la derivativa en la dirección normal N de desvanesca en los bordes. Así $\frac{\partial\phi}{\partial N} = 0$ sobre $\partial\Omega$.
- Una condición inicial en un tiempo $t = 0$. Una buena candidata es la función de distancia con signo, como hemos visto.

Entonces el modelo final realmente queda así:

$$\begin{cases} \phi_t(\mathbf{x}, t) = v|\nabla\phi(\mathbf{x}, t)| & \text{para } (\mathbf{x}, t) \in \Omega \times]0, +\infty[, \\ \phi(\mathbf{x}, 0) = \bar{d}(\mathbf{x}, \Gamma_0) & , \\ \frac{\partial\phi}{\partial N} = 0 & \text{para } (\mathbf{x}, t) \in \partial\Omega \times]0, +\infty[. \end{cases} \quad (5.9)$$

Esta es la ecuación introducida por S.Osher y J.Sethian, nótese que para una cierta función de velocidad v se obtiene una ecuación clásica de Hamilton-Jacobi [4].

5.3.1. Otros aspectos de la Formulación de los Conjuntos de Nivel

El hecho de trabajar bajo una formulación Euleriana (Hamilton-Jacobi), trae consigo algunas aspectos deseables [4], como:

- Primero, la función evolutiva $\phi(\mathbf{x}, t)$ siempre permanece siendo una función tanto como v sea suave. Sin embargo, la superficie de nivel $\phi = 0$ (y por ende la hipersuperficie Γ_t), podría cambiar su topología, romperse, fusionarse y formar esquinas. Esta es una de las principales ventajas de esta representación, pues no necesita considerar los cambios topológicos numéricamente.
- Segundo, concerniente a las aproximaciones numéricas: Se puede usar una malla discreta fija en el dominio espacial y elegir la aproximación por diferencias finitas para las derivativas espaciales y temporales.
- Otra ventaja es que permite la representación intrínseca de los elementos geométricos de la interfaz, como la normal y la curvatura, los cuales pueden ser fácilmente expresados con respecto a ϕ . Esto es una condición necesaria para que cualquier representación resulte útil.
- Por último, la formulación de los Conjunto de Nivel puede ser extendida y aplicada a cualquier dimensión.

Capítulo 6

Métodos de Conjuntos de Nivel para minimizar el Funcional de Mumford-Shah

En el capítulo 4 se expusieron las razones de porque el funcional Mumford-Shah no podía ser trabajado directamente, por lo que es necesario buscar un funcional equivalente que sea más manipulable y que permita plantear una solución para la segmentación de imágenes.

Una característica esencial del funcional Mumford-Shah es que depende de la curvatura del conjunto de bordes Γ .

Como vimos en el capítulo 5, los conjuntos de nivel ofrecen propiedades convenientes que permiten representar funcionales que involucran la curvatura, por lo que en este capítulo formulamos un funcional equivalente al de Mumford-Shah, el cual nos permite plantear la solución al problema de la segmentación de imágenes. Otras formulaciones del funcional Mumford-Shah pueden ser vistas en [6, 5, 2, 3].

6.1. Formulación del Funcional Mumford-Shah con Conjuntos de Nivel

Ya que el problema que nos interesa aquí es lograr minimizar el funcional (4.17), para facilitar nuestro desarrollo, debemos asumir un caso simple en donde la imagen f pueda asumirse como una función de piezas constantes, es decir que se cumpla que $f = constante(c_i)$ dentro de cada región R_i , donde $c_i = media(f)$ sobre cada componente conexa R_i de $\Omega \setminus \mathcal{C}$. Esto es frecuentemente conocido como el *Problema de Partición Mínima*.

Dada la curva $\mathcal{C} = \partial R$. Asociaremos una función de conjunto de nivel ϕ_i con cada fase¹ R_i . Entonces una fase R_i queda definida por:

$$R_i = \{(x, y) \in \Omega : \phi_i(x, y) > 0\},$$

y los bordes de cada fase son definidos por la unión de los conjuntos de nivel cero de ϕ_i . La función del conjunto de nivel mueve cada conjunto de nivel hacia el objeto, la rapidez de este movimiento depende de la proximidad entre la interfaz evolutiva y el objeto. Pero algunos problemas pueden surgir como vacíos o superposición, para prevenir esto, es necesario establecer que cada R_i es disjunta y $\bigcup_i R_i = \Omega$, esto se logra satisfaciendo esta condición :

$$\sum_{i=1}^n H(\phi_i) = 1, \forall (x, y) \in \Omega, \tag{6.1}$$

¹Se entiende como fase a un conjunto de regiones que tienen un mismo tono de gris.

donde H es la función Heaviside unidimensional definida por:

$$H(z) = \begin{cases} 1 & \text{Si } z \geq 0, \\ 0 & \text{Si } z < 0. \end{cases}$$

también será necesario definir la medida Delta Dirac unidimensional δ , que será útil más adelante, cuando describamos nuestra solución.

$$\delta(z) = \frac{d}{dz}H(z)$$

La condición(6.1) debe considerarse como un término adicional de la energía que se busca minimizar con respecto a ϕ_i , relacionado a un multiplicador de Lagrange, de la forma:

$$\int_{\Omega} \left(\sum_{i=1} H(\phi_i) - 1 \right) dx dy,$$

Este término puede ayudar a resolver problemas de transición de fases, pero para un número pequeño de regiones, sin embargo para efectos de la segmentación en donde por lo general existen varias fases, esto debe ser mejorado.

Para mejorar entonces el problema anterior, consideraremos para cualquier imagen que es necesario sólo $\log n$ funciones de conjuntos de nivel para representar n fases, incluyendo la solución a problemas de vacíos y superposiciones.

Sea $m = \log n$ funciones de conjuntos de nivel $\phi_i : \Omega \rightarrow R$. Los bordes de la imagen segmentada están dados por los conjuntos de nivel cero de ϕ_i . Además introduciremos el uso de un vector de funciones de conjuntos de nivel $\Phi = (\phi_1, \dots, \phi_m)$ y también un vector de funciones Heaviside $H(\Phi) = (H(\phi_1), \dots, H(\phi_m))$ cuyas componentes son sólo 1 o 0. Además definire-

mos las fases en el dominio Ω , así dos píxels (x_1, y_1) y (x_2, y_2) pueden pertenecer a la misma fase, si y sólo si $H(\Phi(x_1, y_1)) = H(\Phi(x_2, y_2))$, es decir, cada fase está determinada por la función de conjuntos de nivel $H(\Phi)$.

Podemos deducir además, que existen $n = 2^m$ posibles fases o clases en el dominio Ω . De esta manera, las fases forman una descomposición disjunta y a su vez, su unión provee el dominio completo.

Si etiquetamos cada fase con I , con $1 \leq I \leq 2^m = n$. Seguidamente introducimos un vector constante de promedios $c = (c_1, c_2, \dots, c_n)$, donde $c_I = \text{media}(f)$ en la clase I , y considerando una función característica χ_I para cada clase I , donde $\chi_I = H(\phi_i)$. Entonces, el funcional puede ser escrito así:

$$F_n(c, \Phi) = \sum_{1 \leq I \leq n=2^m} \nu \int_{\Omega} |f - c_I|^2 \chi_I dx dy + \frac{1}{2} \sum_{1 \leq i \leq m} \mu \int_{\Omega} |\nabla \chi_I|, \quad (6.2)$$

donde el término de longitud está dado por:

$$|\mathcal{C}| = \frac{1}{2} \sum_I \int_{\Omega} |\nabla \chi_I|,$$

donde \mathcal{C} denota el conjunto de bordes. Y aún podríamos simplificar este término de longitud, haciéndolo:

$$|\mathcal{C}| \approx \sum_i \int_{\Omega} |\nabla H(\phi_i)|,$$

esto expresa la suma de los conjuntos de nivel cero que corresponden a los bordes. Algunas bordes podrían tener diferentes pesos en el término total de longitud, debido a su espesor.

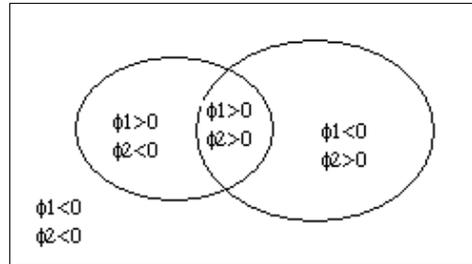


Figura 6.1: Dos curvas dadas por $\phi_1 = 0$ y $\phi_2 = 0$, particionan el dominio en cuatro regiones: $\{\phi_1 > 0, \phi_2 > 0\}$, $\{\phi_1 > 0, \phi_2 < 0\}$, $\{\phi_1 < 0, \phi_2 > 0\}$, $\{\phi_1 < 0, \phi_2 < 0\}$.

Por tanto la energía que minimizaremos está dada por:

$$F_n(c, \Phi) = \sum_{1 \leq I \leq n=2m} \nu \int_{\Omega} |f - c_I|^2 \chi_I dx dy + \sum_{1 \leq i \leq m} \mu \int_{\Omega} |\nabla H(\phi_i)|. \quad (6.3)$$

Obviamente aquí el conjunto de bordes \mathcal{C} está representado por la unión de los conjuntos de nivel cero.

Es fácil darse cuenta, por ejemplo si deseamos realizar una segmentación sencilla en una imagen que contenga un solo objeto, ya que existen $n = 2$ fases (objeto y fondo), sólo sería necesario $m = 1$ funciones de conjuntos de nivel. Pero, si consideramos por ejemplo $n = 4$ fases entonces requerimos de $m = 2$ funciones de nivel. Llegado este punto y basándonos en el *Teorema de los cuatro colores*, de la Teoría de Grafos, podemos decir que es posible segmentar una imagen multifase óptimamente, coloreando todas las regiones en una partición usando sólo cuatro colores, tal que un par de regiones adyacentes tienen distintos colores. Por lo tanto, con sólo $n = 2$ funciones de conjuntos de nivel, podemos identificar las cuatro fases análogas a los cuatro colores, formados por funciones de nivel disjuntas $\{\phi_1 > 0, \phi_2 > 0\}$, $\{\phi_1 > 0, \phi_2 < 0\}$, $\{\phi_1 < 0, \phi_2 > 0\}$, $\{\phi_1 < 0, \phi_2 < 0\}$. En la figura 6.1 podemos observar las regiones que se forman por el particionamiento del dominio.

Además considerando que son los bordes de las regiones los que forman la partición $\{\phi_1 = 0\} \cup \{\phi_2 = 0\}$, lo cual conforma el conjunto de bordes \mathcal{C} .

Según lo mencionado, veamos a continuación cómo podemos plantear el funcional (6.3), considerando dos funciones de conjuntos de nivel, para lograr cuatro fases.

$$\begin{aligned}
F_4(c, \Phi) = & \nu_1 \int_{\Omega} |g(x, y) - c_{11}|^2 H(\phi_1) H(\phi_2) dx dy \\
& + \nu_2 \int_{\Omega} |g(x, y) - c_{10}|^2 H(\phi_1) (1 - H(\phi_2)) dx dy \\
& + \nu_3 \int_{\Omega} |g(x, y) - c_{01}|^2 (1 - H(\phi_1)) H(\phi_2) dx dy \\
& + \nu_4 \int_{\Omega} |g(x, y) - c_{00}|^2 (1 - H(\phi_1)) (1 - H(\phi_2)) dx dy \\
& + \mu_1 \int_{\Omega} |\nabla H(\phi_1)| \\
& + \mu_2 \int_{\Omega} |\nabla H(\phi_2)|,
\end{aligned} \tag{6.4}$$

donde $c = (c_{11}, c_{10}, c_{01}, c_{00})$ es un vector de constantes y $\Phi = (\phi_1, \phi_2)$.

Para efectos de simplicidad en nuestras derivaciones, obviaremos trabajar con los parámetros $\nu_1, \nu_2, \nu_3, \nu_4, \mu_1$ y μ_2 , que serán luego considerados en la solución final.

6.2. Derivación de las Ecuaciones de Euler Lagrange

Para derivar las ecuaciones Euler Lagrange² asociadas al funcional (6.4), consideraremos por razones de simplicidad derivar estas ecuaciones para el

²Ver Apéndice A.1.4

término general:

$$\hat{F}(\phi) = \int_{\Omega} |\nabla H(\phi)|,$$

En los resultados obtenidos sólo bastará con reemplazar ϕ por sus respectivos ϕ_1 y ϕ_2 .

Teorema 6.2.1. *Sea $\hat{F}(\phi) = \int_{\Omega} |\nabla H(\phi)|$, y $\phi = \phi(x, t)$, $x \in R^2$, entonces la ecuación Euler-Lagrange asociada es:*

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right), \quad (6.5)$$

con la condición de frontera: $\frac{\partial \phi}{\partial N} \Big|_{\partial \Omega} = 0$, donde N es el vector unitario normal a $\partial \Omega$ hacia afuera.

Demostración. Para obtener la ecuación de Euler-Lagrange, tomaremos la derivada de \hat{F} de acuerdo al tiempo t , entonces determinaremos la derivada de ϕ con respecto a t de manera que decremente al funcional \hat{F} .

$$\begin{aligned} \hat{F}(\phi) &= \int_{\Omega} |\nabla H(\phi)| \\ &= \int_{\Omega} \delta(\phi) |\nabla \phi| \end{aligned}$$

Derivando con respecto al tiempo, tenemos

$$\begin{aligned} \frac{\partial \hat{F}(\phi)}{\partial t} &= \int_{\Omega} \delta'(\phi) \phi_t |\nabla \phi| + |\nabla \phi|_t \delta(\phi) \\ &= \int_{\Omega} \delta'(\phi) \phi_t |\nabla \phi| + \left(\frac{\nabla \phi}{|\nabla \phi|} \delta(\phi) \right) \cdot \nabla \phi_t \end{aligned}$$

Ahora, aplicando la fórmula de Green³ al segundo término, tenemos

$$\frac{\partial \hat{F}(\phi)}{\partial t} = \int_{\Omega} \delta'(\phi) \phi_t |\nabla \phi| - \phi_t \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \delta(\phi) \right) + \int_{\partial \Omega} \phi_t \frac{\delta(\phi)}{|\nabla \phi|} \frac{\partial \phi}{\partial N}$$

³Ver Apéndice A.2

donde $\frac{\partial \phi}{\partial N} \Big|_{\partial \Omega} = 0$ según la condición de frontera de Neumann.

Desarrollando el segundo término tenemos

$$\begin{aligned} \frac{\partial \hat{F}(\phi)}{\partial t} &= \int_{\Omega} \delta'(\phi) \phi_t |\nabla \phi| - \phi_t \left[\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \delta(\phi) + \delta'(\phi) \nabla \phi \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right] \\ &= \int_{\Omega} \delta'(\phi) \phi_t |\nabla \phi| - \phi_t \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \delta(\phi) - \delta'(\phi) \phi_t |\nabla \phi| \end{aligned}$$

Eliminando el primer y tercer término, obtenemos

$$\frac{\partial \hat{F}(\phi)}{\partial t} = \int_{\Omega} \phi_t \left[-\delta(\phi) \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right]$$

Por tanto, para que el funcional decrezca, hacemos

$$\phi_t = \delta(\phi) \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right)$$

Usando la notación del teorema

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right). \quad (6.6)$$

□

Teorema 6.2.2. *Sea*

$$\begin{aligned} \check{F}(\phi_1) &= \int_{\Omega} |g - c_{11}|^2 H(\phi_1) H(\phi_2) + \int_{\Omega} |g - c_{10}|^2 H(\phi_1) (1 - H(\phi_2)) \\ &\quad + \int_{\Omega} |g - c_{01}|^2 (1 - H(\phi_1)) H(\phi_2) + \int_{\Omega} |g - c_{00}|^2 (1 - H(\phi_1)) (1 - H(\phi_2)) \end{aligned} \quad (6.7)$$

y $g = g(x)$, $x \in \mathbb{R}^2$, entonces la ecuación Euler-Lagrange asociada es:

$$\frac{\partial \phi_1}{\partial t} = -\delta(\phi_1) \left[[(g-c_{11})^2 - (g-c_{01})^2] [H(\phi_2)] + [(g-c_{10})^2 - (g-c_{00})^2] [1-H(\phi_2)] \right] \quad (6.8)$$

Demostración. Para obtener la ecuación de Euler-Lagrange, tomaremos la derivada de $\check{F}(\phi_1)$ de acuerdo al tiempo t , entonces determinaremos la derivada de ϕ_1 para hacer que decremente el funcional $\check{F}(\phi_1)$.

$$\begin{aligned} \check{F}(\phi_1) &= \int_{\Omega} |g - c_{11}|^2 H(\phi_1) H(\phi_2) + \int_{\Omega} |g - c_{10}|^2 H(\phi_1) (1 - H(\phi_2)) \\ &\quad + \int_{\Omega} |g - c_{01}|^2 (1 - H(\phi_1)) H(\phi_2) + \int_{\Omega} |g - c_{00}|^2 (1 - H(\phi_1)) (1 - H(\phi_2)) \end{aligned}$$

Derivando con respecto al tiempo, tenemos

$$\begin{aligned} \frac{\partial \check{F}(\phi_1)}{\partial t} &= \int_{\Omega} \delta(\phi_1) \phi_{1_t} |g - c_{11}|^2 H(\phi_2) + \delta(\phi_1) \phi_{1_t} |g - c_{10}|^2 (1 - H(\phi_2)) \\ &\quad - \delta(\phi_1) \phi_{1_t} |g - c_{01}|^2 H(\phi_2) - \delta(\phi_1) \phi_{1_t} |g - c_{00}|^2 (1 - H(\phi_2)) \end{aligned}$$

Factorizando y tomando ϕ_{1_t} de manera que $\check{F}(\phi_1)$ decrezca tenemos

$$\frac{\partial \phi_1}{\partial t} = -\delta(\phi_1) \left[[(g-c_{11})^2 - (g-c_{01})^2] [H(\phi_2)] + [(g-c_{10})^2 - (g-c_{00})^2] [1-H(\phi_2)] \right] \quad (6.9)$$

□

Teorema 6.2.3. *Sea*

$$\begin{aligned}\check{F}(\phi_2) &= \int_{\Omega} |g - c_{11}|^2 H(\phi_1) H(\phi_2) + \int_{\Omega} |g - c_{10}|^2 H(\phi_1) (1 - H(\phi_2)) \\ &\quad + \int_{\Omega} |g - c_{01}|^2 (1 - H(\phi_1)) H(\phi_2) + \int_{\Omega} |g - c_{00}|^2 (1 - H(\phi_1)) (1 - H(\phi_2))\end{aligned}\tag{6.10}$$

y $g = g(x)$, $x \in \mathbb{R}^2$, entonces la ecuación Euler-Lagrange asociada es:

$$\frac{\partial \phi_2}{\partial t} = -\delta(\phi_2) \left[[(g - c_{11})^2 - (g - c_{10})^2] [H(\phi_1)] + [(g - c_{01})^2 - (g - c_{00})^2] [1 - H(\phi_1)] \right]\tag{6.11}$$

Demostración. Para obtener la ecuación de Euler-Lagrange, tomaremos la derivada de $\check{F}(\phi_2)$ de acuerdo al tiempo t , entonces determinaremos la derivada de ϕ_2 para hacer que decremente el funcional $\check{F}(\phi_2)$.

$$\begin{aligned}\check{F}(\phi_2) &= \int_{\Omega} |g - c_{11}|^2 H(\phi_1) H(\phi_2) + \int_{\Omega} |g - c_{10}|^2 H(\phi_1) (1 - H(\phi_2)) \\ &\quad + \int_{\Omega} |g - c_{01}|^2 (1 - H(\phi_1)) H(\phi_2) + \int_{\Omega} |g - c_{00}|^2 (1 - H(\phi_1)) (1 - H(\phi_2))\end{aligned}$$

Derivando con respecto al tiempo, tenemos

$$\begin{aligned}\frac{\partial \check{F}(\phi_2)}{\partial t} &= \int_{\Omega} \delta(\phi_2) \phi_{2t} |g - c_{11}|^2 H(\phi_1) - \delta(\phi_2) \phi_{2t} |g - c_{10}|^2 H(\phi_1) \\ &\quad + \delta(\phi_2) \phi_{2t} |g - c_{01}|^2 (1 - H(\phi_1)) - \delta(\phi_2) \phi_{2t} |g - c_{00}|^2 (1 - H(\phi_1))\end{aligned}$$

Factorizando y tomando ϕ_{2t} de manera que $\check{F}(\phi_2)$ decrezca tenemos

$$\frac{\partial \phi_2}{\partial t} = -\delta(\phi_2) \left[[(g - c_{11})^2 - (g - c_{10})^2] [H(\phi_1)] + [(g - c_{01})^2 - (g - c_{00})^2] [1 - H(\phi_1)] \right]\tag{6.12}$$

□

En las ecuaciones derivadas anteriormente, los funcionales se derivaron en función de ϕ , manteniendo fijas las constantes $c_{11}, c_{10}, c_{01}, c_{00}$. Ahora, mantendremos fija a ϕ y minimizaremos la energía de nuestro funcional con respecto a estas constantes.

Teorema 6.2.4. *Sea $g = g(x)$, $x \in R^2$, entonces*

$$\begin{cases} c_{11} = \text{media}(g) & \text{en } \{\phi_1 > 0, \phi_2 > 0\}, \\ c_{10} = \text{media}(g) & \text{en } \{\phi_1 > 0, \phi_2 < 0\}, \\ c_{01} = \text{media}(g) & \text{en } \{\phi_1 < 0, \phi_2 > 0\}, \\ c_{00} = \text{media}(g) & \text{en } \{\phi_1 < 0, \phi_2 < 0\}. \end{cases} \quad (6.13)$$

Demostración. Probaremos los cuatro casos y tomaremos en cuenta el funcional \tilde{F} .

- **Constante c_{11} :** Derivamos $\tilde{F}(c_{11})$ con respecto al tiempo.

$$\frac{\partial \tilde{F}(c_{11})}{\partial t} = \int_{\Omega} 2|g - c_{11}| |g - c_{11}|_t H(\phi_1) H(\phi_2)$$

Conociendo que la derivada de c_{11} en el tiempo es cero, tenemos

$$\begin{aligned} 0 &= \int_{\Omega} g H(\phi_1) H(\phi_2) - \int_{\Omega} c_{11} H(\phi_1) H(\phi_2) \\ \int_{\Omega} g H(\phi_1) H(\phi_2) &= c_{11} \int_{\Omega} H(\phi_1) H(\phi_2) \end{aligned}$$

Despejando c_{11}

$$c_{11}(\phi_1, \phi_2) = \frac{\int_{\Omega} g H(\phi_1) H(\phi_2)}{\int_{\Omega} H(\phi_1) H(\phi_2)} \quad (6.14)$$

- **Constante c_{10} :** Derivamos $\check{F}(c_{10})$ con respecto al tiempo.

$$\frac{\partial \check{F}(c_{10})}{\partial t} = \int_{\Omega} 2|g - c_{10}| |g - c_{10}|_t H(\phi_1)(1 - H(\phi_2))$$

Conociendo que la derivada de c_{10} en el tiempo es cero, tenemos

$$0 = \int_{\Omega} gH(\phi_1)(1 - H(\phi_2)) - \int_{\Omega} c_{10}H(\phi_1)(1 - H(\phi_2))$$

$$\int_{\Omega} gH(\phi_1)(1 - H(\phi_2)) = c_{10} \int_{\Omega} H(\phi_1)(1 - H(\phi_2))$$

Despejando c_{10}

$$c_{10}(\phi_1, \phi_2) = \frac{\int_{\Omega} gH(\phi_1)(1 - H(\phi_2))}{\int_{\Omega} H(\phi_1)(1 - H(\phi_2))} \quad (6.15)$$

- **Constante c_{01} :** Derivamos $\check{F}(c_{01})$ con respecto al tiempo.

$$\frac{\partial \check{F}(c_{01})}{\partial t} = \int_{\Omega} 2|g - c_{01}| |g - c_{01}|_t (1 - H(\phi_1))H(\phi_2)$$

Conociendo que la derivada de c_{01} en el tiempo es cero, tenemos

$$0 = \int_{\Omega} g(1 - H(\phi_1))H(\phi_2) - \int_{\Omega} c_{01}(1 - H(\phi_1))H(\phi_2)$$

$$\int_{\Omega} g(1 - H(\phi_1))H(\phi_2) = c_{01} \int_{\Omega} (1 - H(\phi_1))H(\phi_2)$$

Despejando c_{01}

$$c_{01}(\phi_1, \phi_2) = \frac{\int_{\Omega} g(1 - H(\phi_1))H(\phi_2)}{\int_{\Omega} (1 - H(\phi_1))H(\phi_2)} \quad (6.16)$$

- **Constante c_{00} :** Derivamos $\check{F}(c_{00})$ con respecto al tiempo.

$$\frac{\partial \check{F}(c_{00})}{\partial t} = \int_{\Omega} 2|g - c_{00}| |g - c_{00}|_t (1 - H(\phi_1))(1 - H(\phi_2))$$

Conociendo que la derivada de c_{00} en el tiempo es cero, tenemos

$$0 = \int_{\Omega} g(1 - H(\phi_1))(1 - H(\phi_2)) - \int_{\Omega} c_{00}(1 - H(\phi_1))(1 - H(\phi_2))$$

$$\int_{\Omega} g(1 - H(\phi_1))(1 - H(\phi_2)) = c_{00} \int_{\Omega} (1 - H(\phi_1))(1 - H(\phi_2))$$

Despejando c_{00}

$$c_{00}(\phi_1, \phi_2) = \frac{\int_{\Omega} g(1 - H(\phi_1))(1 - H(\phi_2))}{\int_{\Omega} (1 - H(\phi_1))(1 - H(\phi_2))} \quad (6.17)$$

□

Después de haber derivado las ecuaciones Euler Lagrange de los funcionales \hat{F} y \check{F} , procederemos a fusionar los resultados obtenidos para obtener las ecuaciones diferenciales que rigen el comportamiento de ϕ_1 y ϕ_2 .

La ecuación diferencial para ϕ_1 , según lo obtenido en (6.6) y (6.9), tenemos

$$\frac{\partial \phi_1}{\partial t} = \delta(\phi_1) \left\{ \mu_1 \nabla \cdot \left(\frac{\nabla \phi_1}{|\nabla \phi_1|} \right) - \left[(\nu_1(g - c_{11})^2 - \nu_3(g - c_{01})^2) H(\phi_2) \right. \right. \\ \left. \left. + (\nu_2(g - c_{10})^2 - \nu_4(g - c_{00})^2)(1 - H(\phi_2)) \right] \right\} \text{ en } \Omega \times (0, \infty)$$

bajo las condiciones

$$\begin{cases} \phi_1(x, 0) = \phi_1^0(x) & \text{en } \Omega, \\ \frac{\delta(\phi_1)}{|\nabla \phi_1|} \frac{\partial \phi_1}{\partial N} = 0 & \text{sobre } \partial\Omega. \end{cases}$$

La ecuación diferencial para ϕ_2 , según lo obtenido en (6.6) y (6.12), tenemos

$$\frac{\partial \phi_2}{\partial t} = \delta(\phi_2) \left\{ \mu_2 \nabla \cdot \left(\frac{\nabla \phi_2}{|\nabla \phi_2|} \right) - \left[(\nu_1(g - c_{11})^2 - \nu_2(g - c_{10})^2) H(\phi_1) + (\nu_3(g - c_{01})^2 - \nu_4(g - c_{00})^2) (1 - H(\phi_1)) \right] \right\} \text{ en } \Omega \times (0, \infty)$$

bajo las condiciones

$$\begin{cases} \phi_2(x, 0) = \phi_2^0(x) & \text{en } \Omega, \\ \frac{\delta(\phi_2)}{|\nabla \phi_2|} \frac{\partial \phi_2}{\partial N} = 0 & \text{sobre } \partial\Omega. \end{cases}$$

Con estas notaciones, nosotros podemos expresar la función que representa a la imagen f (imagen segmentada) como:

$$\begin{aligned} f &= c_{11}H(\phi_1)H(\phi_2) + c_{10}H(\phi_1)(1 - H(\phi_2)) \\ &+ c_{01}(1 - H(\phi_1))H(\phi_2) + c_{00}(1 - H(\phi_1))(1 - H(\phi_2)). \end{aligned} \quad (6.18)$$

6.3. Implementación Numérica

En esta sección vamos a describir algunos aspectos importantes que nos permitirán implementar eficientemente las ecuaciones diferenciales mostradas en la sección anterior. Al final de esta sección formularemos el algoritmo para la segmentación de imágenes basado en nuestro modelo.

Como hemos visto, la función Heaviside H es una función discontinua, y ya que la función Delta de Dirac δ se concibe como la derivada de Heaviside, ésta no está bien definida. Debido a que la función δ define el movimiento de la funciones de conjuntos de nivel, es indispensable regularizar dichas funciones.

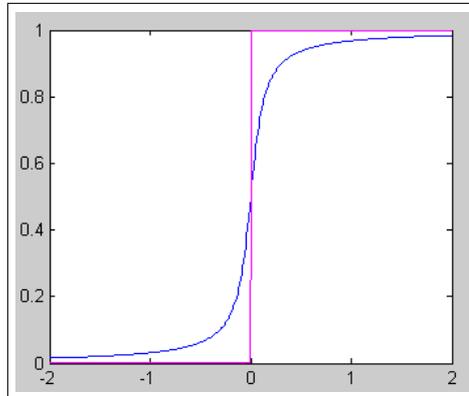


Figura 6.2: Muestra la gráfica de la función Heaviside junto con su regularización.

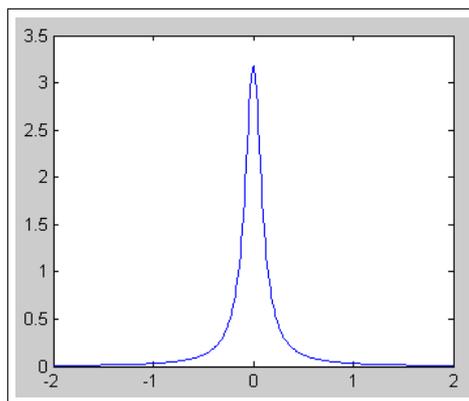


Figura 6.3: Muestra la gráfica de la función Delta de Dirac regularizada.

Para nuestro modelo hemos tomado en cuenta la regularización de H propuesta en [9]

$$H_\varepsilon(z) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan \frac{z}{\varepsilon} \right), \quad (6.19)$$

donde ε es un parámetro tal que H_ε converge a H cuando $\varepsilon \rightarrow 0$. Ahora teniendo a H_ε entonces la consiguiente regularización de δ (Ver figuras 6.2 y 6.3) es

$$\delta_\varepsilon(z) = \frac{\varepsilon}{\pi} \left(\frac{1}{\varepsilon^2 + z^2} \right) \quad (6.20)$$

Para discretizar las ecuaciones en ϕ_1 y ϕ_2 , haremos uso del esquema de *Diferencias Finitas*, siguiendo estas notaciones: sea h el paso de espacio, Δt el paso de tiempo, y $(x_i, y_j) = (ih, jh)$ la malla de puntos, para $1 \leq i \leq N$ y $1 \leq j \leq M$, donde N, M son las dimensiones de la imagen. Sea además, $\phi_{i,j}^n = \phi(x_i, y_j, n\Delta t)$ como una aproximación de $\phi(x, y, t)$, con $n \geq 0$, $\phi^0 = \phi_0$. Las diferencias finitas son

$$\begin{aligned}\Delta_-^x \phi_{i,j} &= \phi_{i,j} - \phi_{i-1,j}, & \Delta_+^x \phi_{i,j} &= \phi_{i+1,j} - \phi_{i,j}, \\ \Delta_-^y \phi_{i,j} &= \phi_{i,j} - \phi_{i,j-1}, & \Delta_+^y \phi_{i,j} &= \phi_{i,j+1} - \phi_{i,j},\end{aligned}$$

Además tomando en cuenta la discretización de la divergencia propuesta en [27], y asumiendo que ϕ y φ son respectivamente ϕ_1 y ϕ_2 (para facilitar la notación) la discretización de las ecuaciones diferenciales es

$$\begin{aligned}\frac{\phi_{i,j}^{t+1} - \phi_{i,j}^t}{\Delta t} &= \delta_\varepsilon(\phi_{i,j}^t) \left[\frac{\mu_1}{h^2} \Delta_-^x \cdot \left(\frac{\Delta_+^x \phi_{i,j}^{t+1}}{\sqrt{\frac{(\Delta_+^x \phi_{i,j}^t)^2}{h^2} + \frac{(\phi_{i,j+1}^t - \phi_{i,j-1}^t)^2}{(2h)^2}}} \right) \right. \\ &\quad + \frac{\mu_1}{h^2} \Delta_-^y \cdot \left(\frac{\Delta_+^y \phi_{i,j}^{t+1}}{\sqrt{\frac{(\phi_{i+1,j}^t - \phi_{i-1,j}^t)^2}{(2h)^2} + \frac{(\Delta_+^y \phi_{i,j}^t)^2}{h^2}}} \right) \\ &\quad - [(\nu_1(g - c_{11}(\phi^t, \varphi^t))^2 - \nu_3(g - c_{01}(\phi^t, \varphi^t))^2) H(\varphi^t) \\ &\quad \left. + (\nu_2(g - c_{10}(\phi^t, \varphi^t))^2 - \nu_4(g - c_{00}(\phi^t, \varphi^t))^2) (1 - H(\varphi^t))] \right] \end{aligned} \tag{6.21}$$

$$\begin{aligned}
\frac{\varphi_{i,j}^{t+1} - \varphi_{i,j}^t}{\Delta t} = \delta_\varepsilon(\varphi_{i,j}^t) & \left[\frac{\mu_2}{h^2} \Delta_-^x \cdot \left(\frac{\Delta_+^x \varphi_{i,j}^{t+1}}{\sqrt{\frac{(\Delta_+^x \varphi_{i,j}^t)^2}{h^2} + \frac{(\varphi_{i,j+1}^t - \varphi_{i,j-1}^t)^2}{(2h)^2}}} \right) \right. \\
& + \frac{\mu_2}{h^2} \Delta_-^y \cdot \left(\frac{\Delta_+^y \varphi_{i,j}^{t+1}}{\sqrt{\frac{(\varphi_{i+1,j}^t - \varphi_{i-1,j}^t)^2}{(2h)^2} + \frac{(\Delta_+^y \varphi_{i,j}^t)^2}{h^2}}} \right) \\
& - [(\nu_1(g - c_{11}(\phi^t, \varphi^t))^2 - \nu_2(g - c_{10}(\phi^t, \varphi^t))^2)H(\phi^t) \\
& \left. + (\nu_3(g - c_{01}(\phi^t, \varphi^t))^2 - \nu_4(g - c_{00}(\phi^t, \varphi^t))^2)(1 - H(\phi^t))] \right]
\end{aligned} \tag{6.22}$$

Finalmente planteamos el siguiente algoritmo:

Algoritmo 4 Algoritmo de Segmentación Mumford-Shah

Entrada: Imagen g

Salida: Imagen segmentada f

- 1: **Procedimiento** SEGMENTACIÓNMUMFORDSHAH(g)
 - 2: Inicializar ϕ_1^0 y ϕ_2^0 con la función de distancia con signo, $t = 0$
 - 3: $anterior = 0$
 - 4: $actual = 10000$
 - 5: **Repetir**
 - 6: $anterior = actual$
 - 7: Computar c_{11} , c_{10} , c_{01} y c_{00} por (6.14),(6.15),(6.16) y (6.17).
 - 8: Obtener ϕ_1^{t+1} y ϕ_2^{t+1} por (6.21) y (6.22).
 - 9: $actual = EvaluarFuncional(\phi_1^{t+1}, \phi_2^{t+1}, c_{11}, c_{10}, c_{01}, c_{00}, g)$
 - 10: **Hasta** $|actual - anterior| < T$
 - 11: Componer la imagen segmentada f por (6.18).
 - 12: **Retornar** f ▷ La imagen segmentada
 - 13: **Fin Procedimiento**
-

El umbral T representa la mínima diferencia de valores del funcional en el tiempo que será permitida en el algoritmo. En imágenes en donde se necesita conservar la mayor cantidad de detalles, T debería ser lo suficientemente pequeño. En las pruebas realizadas en este trabajo, nos encontramos con

$0 < T \leq 10$ y más específicamente, $0 < T < 1$ para mantener el máximo nivel de detalle, y $1 < T < 10$ para realizar un proceso más rápido y con bosquejos de segmentación con menos detalles.

Para calcular la complejidad de nuestro algoritmo, tomaremos en cuenta, algunos aspectos importantes. La inicialización de las funciones ϕ_1^0 y ϕ_2^0 con la función de distancia con signo (línea 2) es $O(1)$, debido a que, para efectos de implementación, la inicialización de las funciones antes mencionadas se realizó una única vez, con dimensiones considerablemente mayores a todas las imágenes de prueba con las que contamos para este trabajo. Éstas funciones se almacenaron en memoria permanente, por lo que en la implementación del algoritmo sólo se realiza un proceso de cargado de éstas funciones, dependiendo del tamaño de la imagen.

Para el cálculo de las constantes $c_{11}, c_{10}, c_{01}, c_{00}$ (línea 7) en el peor de los casos, el cómputo de éstas constantes se realizará en un tiempo $O(n \times m)$, donde n, m son las dimensiones de la imagen.

El proceso de obtener ϕ_1^{t+1} y ϕ_2^{t+1} , se realiza en cada $\phi(x, y)$, por lo que es de orden $O(n \times m)$, al igual que la evaluación del funcional.

Por lo tanto, el tiempo que toma una sola iteración de nuestro algoritmo es $O(n \times m)$. Si asumimos que para una imagen determinada, el algoritmo la segmenta en d iteraciones, entonces nuestro algoritmo es del orden $O(n \times m \times d)$. El factor d depende de la cantidad de detalles de la imagen, ya que las imágenes más simples necesitarán menos iteraciones que las imágenes más complejas.

6.4. Resultados

Concluimos este capítulo presentando algunos resultados obtenidos sobre imágenes reales y sintéticas, producto de la aplicación del algoritmo Mumford-Shah. Mostraremos la imagen original g y la aproximación constante por piezas f . En nuestros experimentos, nosotros elegimos los siguientes parámetros: $\nu_1 = \nu_2 = \nu_3 = \nu_4 = 1$, $h = 1$ (paso espacial), $\Delta t = 0,1$ (paso de tiempo). Sólo los parámetros de longitud μ_1 y μ_2 no son los mismos en todas las pruebas realizadas. Para todas nuestras pruebas, asumimos $\mu_1 = \mu_2 = \mu$. Si nosotros deseamos segmentar todos o la mayoría de objetos como sea posible y de cualquier tamaño, entonces μ debe ser pequeño. Si por el contrario queremos segmentar sólo objetos grandes y no detectar objetos pequeños como puntos de ruido, entonces μ debe ser grande.

Además, empleamos las regularizaciones H_ϵ y δ_ϵ con $\epsilon = 1$. En los siguientes resultados, mostramos a la izquierda la imagen original g , al centro la imagen segmentada f y a la derecha el conjunto de bordes Γ .

Todas las pruebas se realizaron en una computadora de 2,8GHz con 512MB de RAM.



Figura 6.4: Segmentación de una imagen con la presencia de múltiples objetos de la misma clase. Tamaño 175×162 . $\mu = 0,01 \cdot 255^2$.

La imagen original mostrada en 6.4 presenta objetos bien diferenciados del fondo, por lo que la segmentación ha sido buena. Ha logrado separar adecuadamente los objetos del fondo y ha preservado la forma de los objetos presentes en la escena.

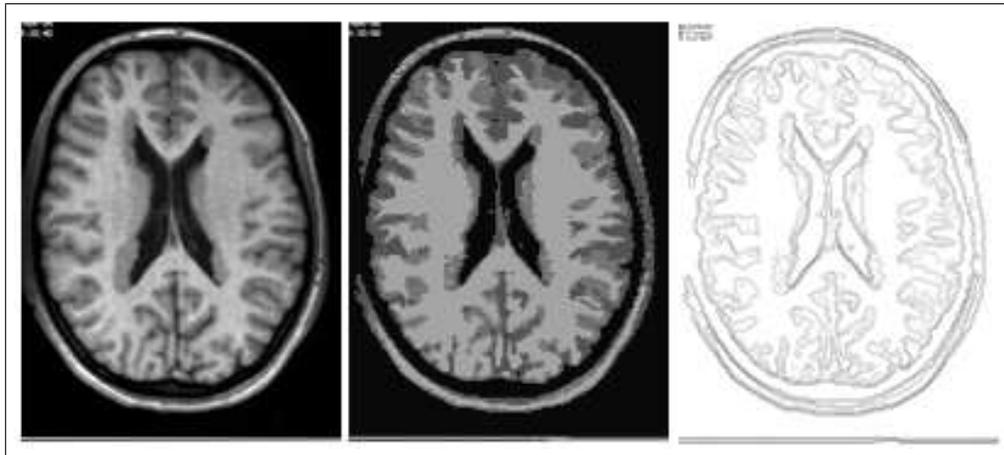


Figura 6.5: Muestra la segmentación de una imagen médica de la corteza cerebral. Tamaño 131×173 . $\mu = 0,001 \cdot 255^2$.

En la imagen 6.5 se aprecia la correcta segmentación de las cuatro regiones más saltantes de la imagen original, preservando los detalles y la forma de los objetos presentes en la imagen original.



Figura 6.6: Imagen facial real. Tamaño 192×192 . $\mu = 0,01 \cdot 255^2$.

En la imagen 6.6 observamos una buena segmentación de las componentes principales del rostro, ojos, cejas, nariz y boca.

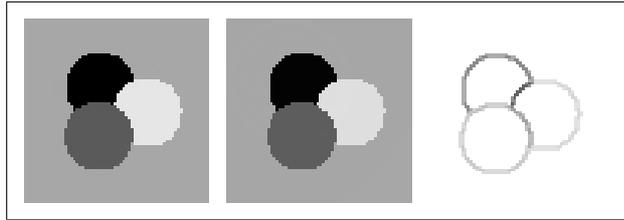


Figura 6.7: Imagen sintética. Tamaño 64×64 . $\mu = 0,01 \cdot 255^2$.

La segmentación de la imagen sintética 6.7 es bastante buena, el algoritmo ha logrado encontrar una imagen f que contiene objetos bien definidos.

Capítulo 7

Análisis de Resultados

A continuación presentamos el análisis de una serie de imágenes de todo tipo, sometidas al proceso de segmentación, usando los diferentes métodos ya descritos en este trabajo.

Este análisis trata de evaluar los logros alcanzados al segmentar las imágenes, aplicando los métodos clásicos de segmentación frente a los resultados obtenidos por segmentar usando el algoritmo de Mumford-Shah.

Esta evaluación ha sido elaborada principalmente mediante inspección visual, y radica en el análisis de los indicadores tomados para considerar que una segmentación es óptima. Estos indicadores fueron descritos en el capítulo 1.

Las imágenes consideradas para ser segmentadas forman parte de una base de datos construída para llevar a cabo nuestro objetivo, y está conformada por diversas categorías de imágenes que permitirán evaluar diversos aspectos, evidentemente dependiendo del contexto de la imagen, que ayudarán a conocer las fortalezas y debilidades del desempeño de los algoritmos clásicos y del algoritmo de Mumford-Shah.

También debemos conocer los aspectos importantes de cada algoritmo clásico. Para el algoritmo multiumbral, se empleó un $k = 2$, es decir sólo encontrará 3 regiones. Para el algoritmo de crecimiento de regiones, el criterio de homogeneidad tomado en cuenta es el de la similitud de tonos de gris entre píxeles vecinos. Se consideraron píxeles similares aquellos cuya diferencia en tonos de gris no era mayor a 30.

Para el algoritmo de crecimiento de regiones implementado aquí, se consideró como entrada una única semilla colocada dentro del objeto de interés en la imagen, con el objetivo de que segmente dicho objeto del fondo y del resto de objetos presentes en la escena.

Para el algoritmo Split-Merge, se empleó como criterio de homogeneidad, la similitud en tonos de gris de píxeles vecinos. El valor umbral para el proceso de descomposición fue 25, al igual que el valor umbral para la fusión.

Iniciamos evaluando la imagen 7.1 que muestra originalmente una imagen 7.1(a) de corte común, que se caracteriza por presentar objetos bien definidos.

En la Fig. 7.1(b) apreciamos el resultado de segmentar la imagen al aplicar el algoritmo Multiumbral.

Se puede apreciar una segmentación aceptable con adecuada separación de los objetos de la escena con el fondo, además de una buena segmentación de los objetos importantes de la escena, en este caso los números y letras.

Podemos apreciar también que el algoritmo ha mantenido la forma de los objetos de interés, a pesar de que ha creado regiones oscuras alrededor de los números, debido a las sombras que son casi imperceptibles y que dan la sensación de que el algoritmo ha creado objetos innecesariamente.



Figura 7.1: (a)Imagen Original.(b)Imagen segmentada usando Multiumbral. (c)Imagen segmentada usando Crecimiento de Regiones.(d)Imagen segmentada usando Split-Merge. (e)Imagen segmentada usando Mumford-Shah.

En la Fig. 7.1(c) podemos apreciar el resultado de segmentar la imagen original 7.1(a) usando crecimiento de regiones. Aquí evidentemente es casi imposible elegir una semilla de tal forma que ésta sea capaz de segmentar todos los objetos, pues cada uno presenta intensidades de gris bien diferenciadas. Es fácil ver que no se mantienen adecuadamente la forma de los objetos, ya que ha eliminado las partes interiores de los números y letras.

En la Fig. 7.1(d) apreciamos la segmentación aplicando el algoritmo Split-Merge. En cuanto a la separación de los objetos y el fondo, al parecer lo logra bien, el problema es que prácticamente la imagen segmentada es muy similar a la original, crea una región por cada objeto de manera que la imagen se vuelve más compleja, es decir, no uniformiza ni reduce el número de regiones que permiten que la imagen sea más simple, que es justamente el objetivo de segmentar.

Finalmente analicemos la imagen segmentada producto de aplicar el algoritmo Mumford-Shah, imagen 7.1(e). Presenta una adecuada diferenciación de las clases de objetos de la imagen, llámense fondo, números y letras. De lo anterior podemos afirmar que en términos generales se observa una buena segmentación de los objetos significativos además de mantener la forma y dimensiones originales de los objetos.

La figura 7.2(a), se trata de una imagen de tipo médico, que se caracteriza por presentar una iluminación y contraste bastante aceptables, contiene objetos grandes pero también detalles finos, además a simple vista se pueden observar la presencia de tres regiones bien definidas.

Analizando para la segmentación Multiumbral, imagen 7.2(b), de acuerdo al primer indicador, debemos decir que no existe una buena diferenciación

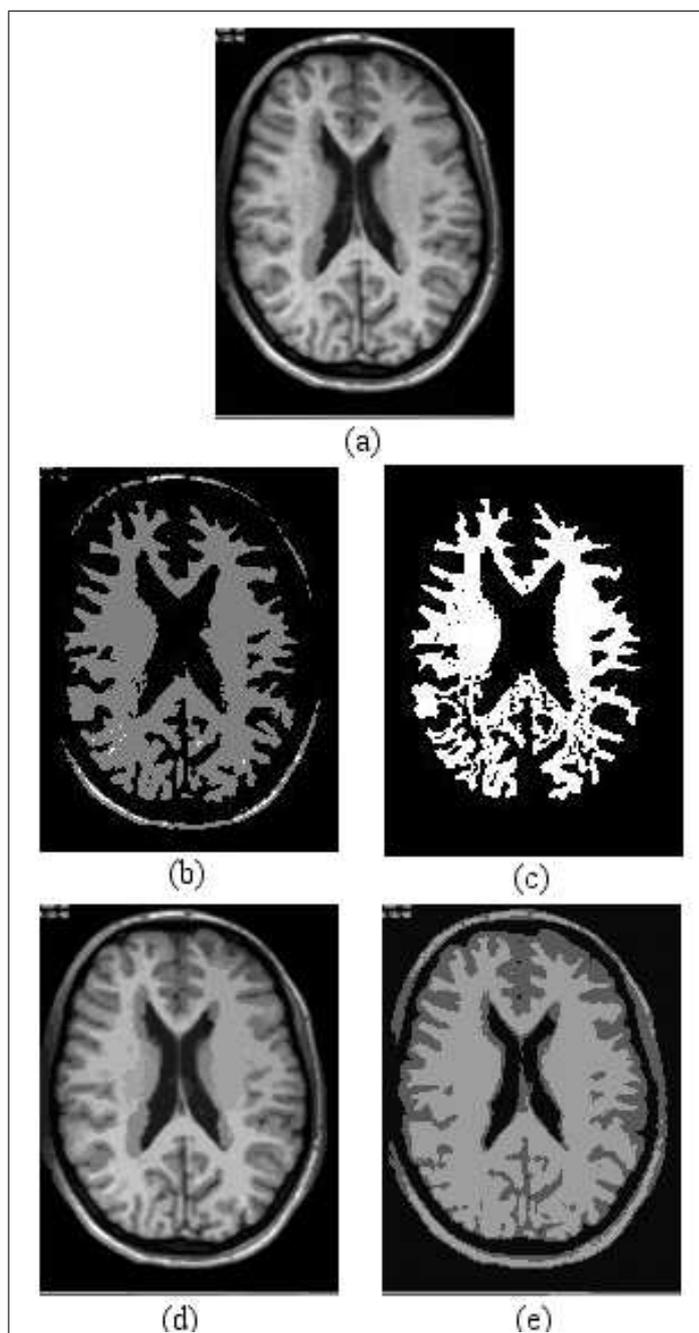


Figura 7.2: (a)Imagen Original.(b)Imagen segmentada usando Multiumbral. (c)Imagen segmentada usando Crecimiento de Regiones.(d)Imagen segmentada usando Split-Merge. (e)Imagen segmentada usando Mumford-Shah.

de los objetos saltantes y el fondo, pues se observa la fusión de las regiones conformadas por los detalles finos que forman parte de los bordes de los objetos característicos de la imagen.

Dando un vistazo también podemos ver que se observa la presencia de puntos blancos, ésto puede interpretarse como que no existe una total homogeneidad en la región correspondiente a la masa cerebral, debido en parte a la imagen original que tiene cierta iluminación en esa misma zona, seguramente al momento de segmentar estos píxels más iluminados se cuelan a manera de ruido.

Una consecuencia de la fusión entre los bordes que rodean a los objetos significativos y los objetos significativos, es la pérdida de los detalles finos de los bordes, ésto también hace que se pierda la forma y tamaño real de los objetos, en realidad la segmentación hace que se cree un objeto nuevo.

En el caso de la segmentación por crecimiento de regiones, imagen 7.2(c), si bien la elección de la semilla ha permitido segmentar el objeto más significativo en la imagen(masa cerebral) y el fondo, ésto no es lo más deseable pues debería de segmentarse todos y cada uno de los objetos que conforman la imagen.

Nuevamente, la fusión trae como consecuencia que durante la segmentación de los objetos significativos (masa cerebral)se pierda gran parte de la información que tiene que ver con los detalles finos que se encuentran en los bordes. Además se pierde información acerca del tamaño y forma original de los objetos.

En cuanto a la segmentación lograda por aplicar el algoritmo Split-Merge, Fig. 7.2(d), la separación de los objetos y el fondo es adecuada, ya que la imagen original presenta regiones bien difereciables.

Si bien se logra capturar los detalles, el problema es que se crean regiones innecesarias, principalmente en las zonas donde existen más iluminación.

Conserva adecuadamente el tamaño y forma real de los objetos.

Finalmente, el caso de la imagen segmentada usando Mumford-Shah Fig. 7.2(e), consigue una segmentación óptima con una separación adecuada de los objetos saltantes, la parte comprendida por las bandas que rodean a dichos objetos y el fondo. es decir identifica las tres regiones que a simple vista conforma la imagen original, Fig. 7.2(a)

A diferencia de otros métodos, Mumford-Shah presenta regiones homogéneas, aún cuando la iluminación de la imagen en la zona central claramente deja apreciar dos tonos distintos del fondo.

La segmentación de los objetos significativos es aceptable, sobre todo en el caso de las bandas que rodean la masa cerebral(exterior y centro) lo que origina la preservación de los detalles de los objetos, por ésto también es que se conserva de mejor manera la forma y las dimensiones de los objetos.

La figura 7.3(a) presenta una imagen facial frontal que posee una iluminación y contraste aceptable y que por su naturaleza tiene objetos significativos más complejos como las cejas, los ojos, los labios y la nariz.

En la Fig. 7.3(b) se muestra la imagen segmentada con el algoritmo multi-umbral. Presenta una segmentación aceptable del rostro y la parte superior del fondo de la imagen. También se puede apreciar una buena segmentación de los componentes principales del rostro(ojos, cejas, nariz y boca). Aunque hay deficiencias a nivel del fondo en la parte inferior que tiene tonos de gris muy similares a los usados para segmentar objetos como orejas y aretes.

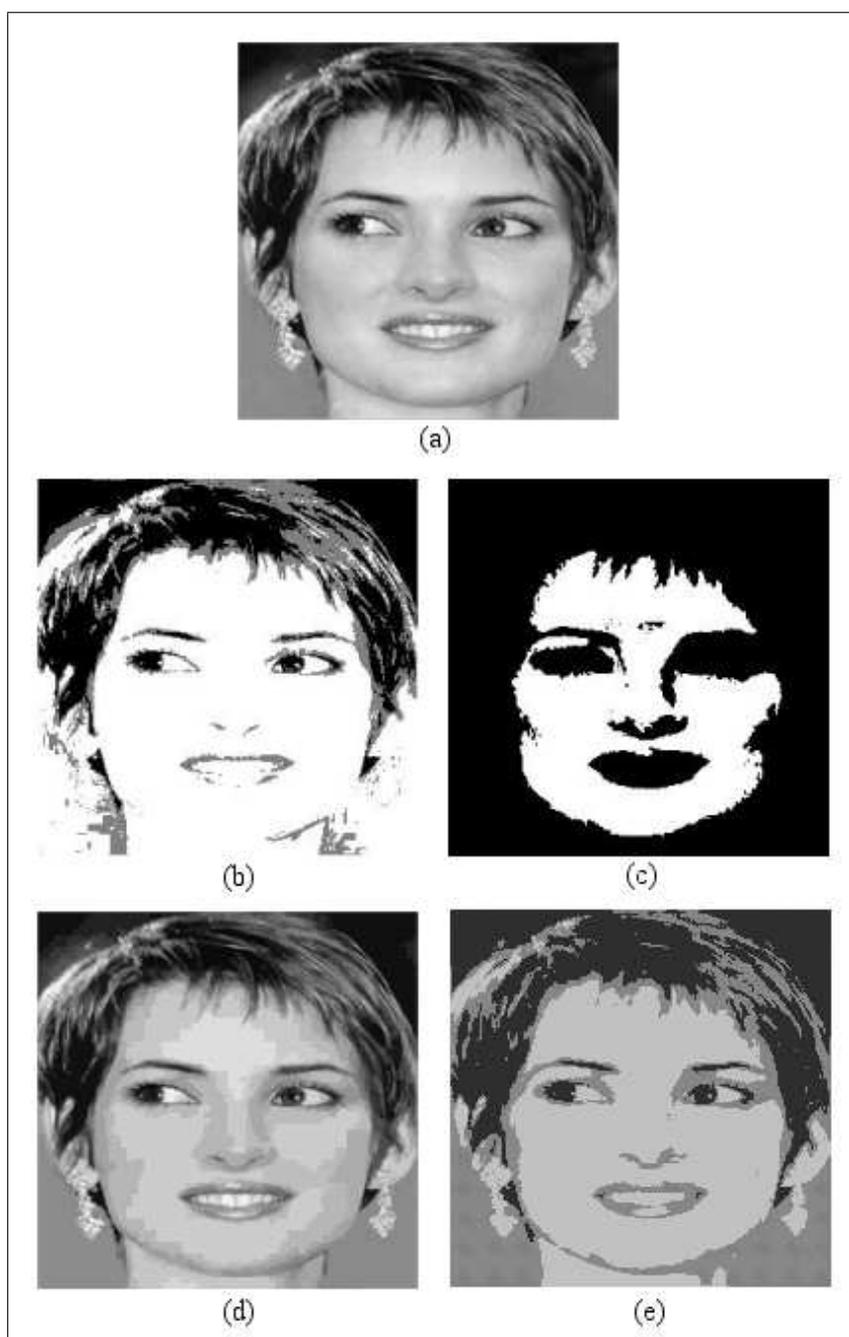


Figura 7.3: (a)Imagen Original.(b)Imagen segmentada usando Multiumbral. (c)Imagen segmentada usando Crecimiento de Regiones.(d)Imagen segmentada usando Split-Merge. (e)Imagen segmentada usando Mumford-Shah.

Además la imagen segmentada 7.3(b) mantiene la forma correcta y dimensiones de los elementos significativos.

La Fig. 7.3(c) es el resultado de la segmentación de la imagen facial con el algoritmo de crecimiento de regiones. No existe una buena separación de los objetos significativos y el fondo debido a las deficiencias del criterio de similitud, pues éste es restringido y permite que la luz y las sombras se consideren parte del rostro.

Además, no se mantienen la forma de los objetos, ya que depende mucho del criterio de similaridad de los píxels vecinos, que varían mucho en un mismo objeto.

La Fig. 7.3(d) presenta el resultado del algoritmo split-merge. Logra segmentar adecuadamente el objeto principal(rostro) del fondo de la imagen, el problema es que crea regiones adicionales, las cuales corresponden a las zonas del rostro que presentan rasgos de sombras y donde hay presencia ligera de luz. Esto puede representar una dificultad en el sentido de que la imagen se vuelve más compleja, cuando lo que se busca es hacerla más sencilla.

Ya que la imagen segmentada y la original son muy parecidas, prácticamente los rasgos faciales han permanecido como regiones individuales, así también, otras características como las orejas, el cabello y los aretes.

Además, preserva adecuadamente la forma y dimensiones de los objetos de la imagen original.

En la Fig. 7.3(e) se muestra el resultado de la segmentación Mumford-Shah para la imagen facial. Se observa una buena segmentación del fondo y los objetos. Se han segmentado adecuadamente todas las características faciales

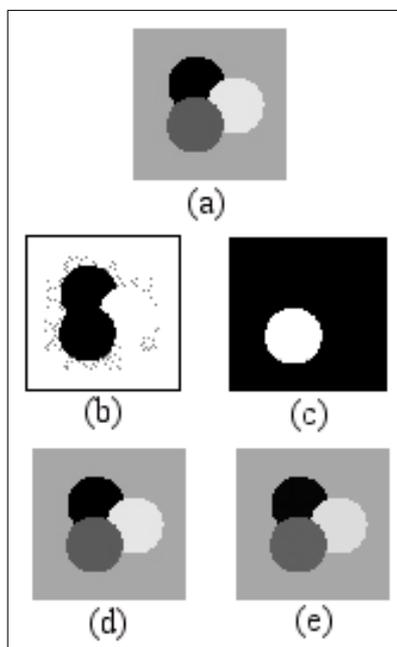


Figura 7.4: (a)Imagen Original.(b)Imagen segmentada usando Multiumbral. (c)Imagen segmentada usando Crecimiento de Regiones.(d)Imagen segmentada usando Split-Merge. (e)Imagen segmentada usando Mumford-Shah.

más importantes(ojos, boca, cejas y nariz). El resultado es una imagen más simple que la original, pero que preserva las características principales. Reduce al máximo el número de regiones, haciéndolas uniformes y manteniendo la forma y dimensiones de las características significativas.

La figura 7.4(a) presenta una imagen sintética totalmente nítida, adecuada iluminación y contraste, y con objetos bien definidos.

En la Fig. 7.4(b) se muestra el resultado de aplicar el algoritmo de segmentación multiumbral para la imagen sintética 7.4(a).

Se puede observar que no existe una adecuada segmentación, pues las cuatro regiones de la imagen original han sido segmentadas en sólo dos regiones. La primera región está formada por la fusión de los círculos más oscuros y la segunda región la conforman el fondo y el círculo más claro, lo cual dista

mucho de la realidad. Además parte del fondo original aparece segmentado con ruido.

La imagen segmentada parece estar ruidosa por la mala segmentación del fondo de la imagen.

En la Fig. 7.4(b) existe una mala segmentación de los objetos significativos, pues los transforma al fusionarlos, y a la vez crea nuevos objetos, los cuales no están presentes en la imagen original. Además se observa pérdida de información del fondo de la imagen.

En la Fig. 7.4(c) se presenta la segmentación de la imagen sintética con el algoritmo de crecimiento de regiones. La separación de los objetos y el fondo es mala. El criterio usado para la devolución de las semillas origina la unión del fondo y de los objetos más oscuros como una sola región, mientras el objeto restante forma una región distinta. Por tanto, no existe una buena segmentación de los objetos significativos, y a la vez no se conserva la forma y las dimensiones originales de los objetos.

En la Fig. 7.4(d) se muestra el resultado de aplicar el algoritmo split-merge en la imagen sintética 7.4(a). Presenta una adecuada segmentación de los objetos y el fondo, creando para ello una región característica para cada objeto y el fondo. Por ello, existe una buena segmentación de los objetos significativos, y además mantiene la forma y tamaño de los objetos originales.

En la Fig. 7.4(e) se observa la imagen segmentada con el algoritmo Mumford-Shah propuesto. Se hace notoria una buena segmentación de los objetos significativos y del fondo, manteniendo su forma y sus dimensiones.

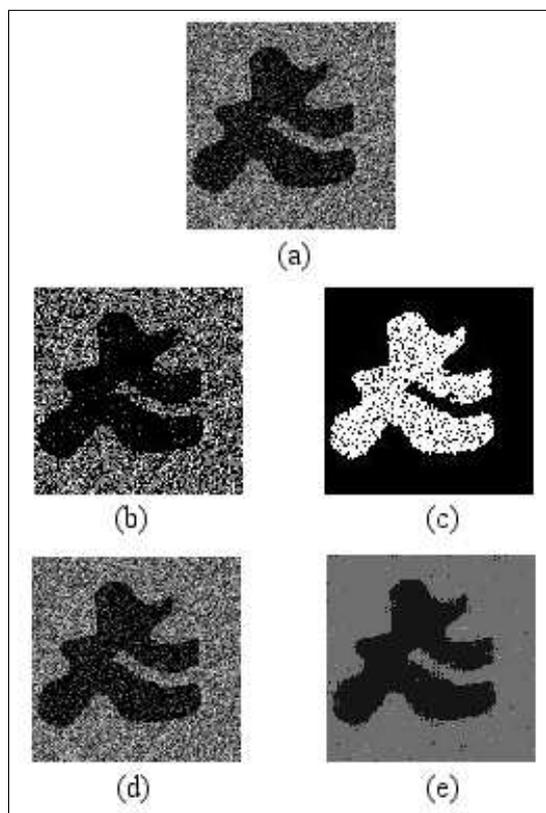


Figura 7.5: (a)Imagen Original.(b)Imagen segmentada usando Multiumbral. (c)Imagen segmentada usando Crecimiento de Regiones.(d)Imagen segmentada usando Split-Merge. (e)Imagen segmentada usando Mumford-Shah.

En la Fig. 7.5(a) se muestra una imagen sintética a la que se le ha añadido una cantidad considerable de ruido. El criterio que evaluaremos en esta imagen, será la sensibilidad al ruido de los algoritmos clásicos y del algoritmo propuesto.

El algoritmo de segmentación multiumbral(Fig. 7.5(b)) es totalmente sensible al ruido, debido a que sólo toma en cuenta la información global de las intensidades de los píxeles, los cuales varían mucho en un mismo objeto ante la presencia de ruido. El algoritmo de crecimiento de regiones(Fig. 7.5(c)) también es sensible al ruido. Esto es debido a que, al existir ruido en el objeto de la imagen, el criterio de homogeneidad es sensible al ruido, el cual hace que la intensidad dentro del objeto varía considerablemente. Por el mismo motivo, el

algoritmo Split-Merge(Fig. 7.5(d)) segmenta demasiadas regiones, considerando incluso al ruido como regiones.

En la segmentación con nuestro algoritmo(Fig. 7.5(e)), vemos que la sensibilidad al ruido se ha reducido en comparación con los otros algoritmos. El algoritmo ha logrado, exitosamente, segmentar el objeto del fondo, definiendo bien sus bordes. Aunque se hacen notorias algunas marcas en la imagen segmentada por el hecho de que la imagen original no es adecuada por el ruido, la segmentación es buena.

Cabe recalcar que para obtener el resultado de la Fig. 7.5 se emplearon los siguientes parámetros: $\mu = 0,15 \times 255^2$, $T = 0,0006$. La imagen tiene un tamaño de 100×100 y el algoritmo realizó 600 iteraciones para hallar el resultado mostrado.

A continuación mostramos una tabla con los tiempos(en segundos) que emplearon los 4 algoritmos analizados para realizar las segmentaciones.

Imagen	Multiumbral	Crec. Regiones	Split-Merge	Mumford-Shah
4627	7	3	4	13
Cerebro	7	1	2	6
Rostro	8	6	2	37
Círculos	2	1	1	2
Ruidosa	7	1	3	154

Cuadro 7.1: Tiempo de ejecución de los algoritmos clásicos y de Mumford-Shah.

7.1. Otros resultados

En esta sección mostraremos otros resultados adicionales obtenidos al aplicar nuestro algoritmo de segmentación basado en el Funcional Mumford-Shah. Las imágenes mostradas presentan a la imagen original a la izquierda, la imagen

segmentada al centro y la imagen que representa al conjunto de bordes, a la derecha. Además se especificará el parámetro μ , el número de iteraciones y el tiempo empleado por el algoritmo para segmentar las imágenes.



Figura 7.6: Imagen sintética. Tamaño 64×64 , $\mu = 0,01 \times 255 \times 255$. 6 iteraciones. 1s.
 $T = 10$

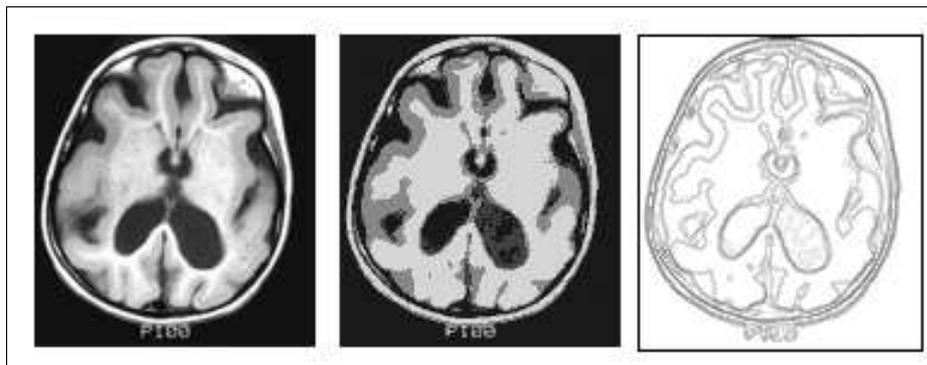


Figura 7.7: Imagen cerebral. Tamaño 117×131 , $\mu = 0,01 \times 255 \times 255$. 13 iteraciones. 8s.
 $T = 5$

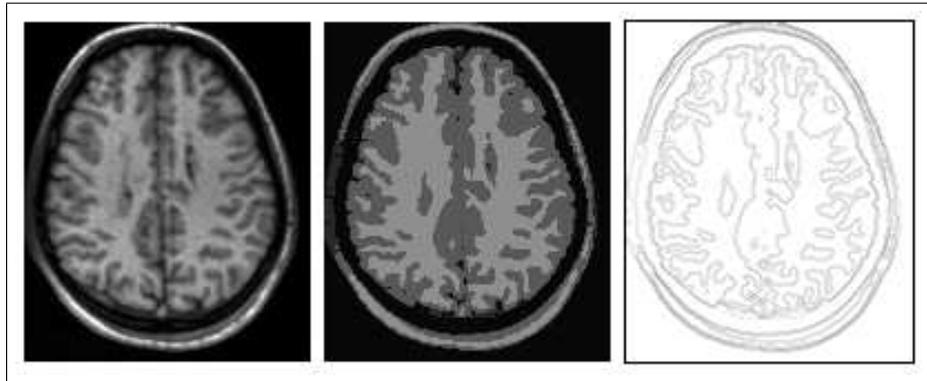


Figura 7.8: Imagen cerebral. Tamaño 129×154 , $\mu = 0,01 \times 255 \times 255$. 10 iteraciones. 7s. $T = 5$

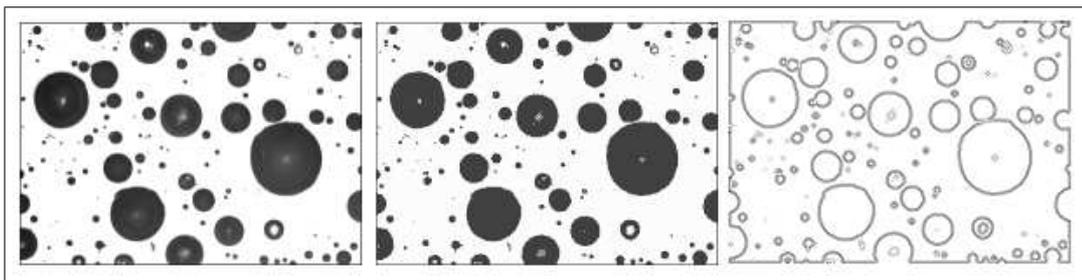


Figura 7.9: Imagen microscópica de glóbulos rojos. Tamaño 196×140 , $\mu = 0,01 \times 255 \times 255$. 7 iteraciones. 7s. $T = 10$



Figura 7.10: Imagen facial. Tamaño 118×184 , $\mu = 0,01 \times 255 \times 255$. 30 iteraciones. 28s. $T = 1$

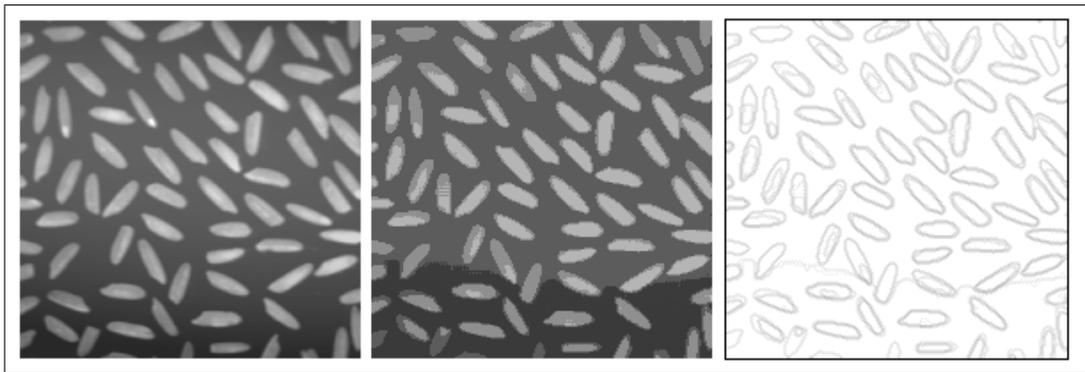


Figura 7.11: Imagen común de arroces. Tamaño 192×192 , $\mu = 0,01 \times 255 \times 255$. 37 iteraciones. 55s. $T = 0,52$

Capítulo 8

Conclusiones

En este capítulo presentaremos las conclusiones a las que hemos llegado después de finalizar nuestro trabajo.

- La mayoría de algoritmos clásicos de Segmentación de Imágenes contemplan factores globales como la tonalidad de gris, pero no usan la información espacial de los objetos presentes en la imagen, por lo que dejan de ser útiles en muchos casos en donde se requiere separar e identificar objetos en una escena.
- El Funcional de Mumford-Shah provee un marco de trabajo adecuado para un algoritmo de segmentación de imágenes óptimo, que mejore el funcionamiento de los algoritmos clásicos existentes.
- El problema de desarrollar el Funcional de Mumford-Shah para obtener una imagen segmentada que cumpla las condiciones establecidas por este funcional es un problema inverso, el cual necesita ser regularizado. Este tipo de problemas no tienen solución directa, lo que hace que los métodos para resolverlo no sean fáciles de desarrollar.

-
- La formulación original de Mumford-Shah es una formulación intuitiva, en la que se define el comportamiento de una imagen bien segmentada, pero matemáticamente hablando, la desventaja principal de la formulación de Mumford-Shah es la no uniformidad de los términos que comprenden medidas de superficies y longitudes de curva.
 - Muchos estudios sobre el tema indican que la mejor manera de analizar el funcional de Mumford-Shah es obteniendo una formulación débil en la que se tiene que reemplazar el término de longitud de bordes por la medida Hausdorff del conjunto de bordes. Este cambio es debido a la necesidad de definir el espacio de funciones en el que existe una solución.
 - La descripción analítica de la solución del Funcional Mumford-Shah permite conocer que la solución depende de la curvatura del conjunto de bordes, por lo que el Funcional involucra medidas geométricas. Este punto es muy importante, pues nos permite decidir el uso de Conjuntos de Nivel para formular un nuevo funcional numéricamente manipulable.
 - El Problema de Partición Mínimo, como caso especial del Funcional Mumford-Shah, ha resultado un excelente criterio para mantener el problema de segmentación mucho más simple sin dejar de obtener buenos resultados al aplicarlo.
 - El método de Conjuntos de Nivel posee el marco de trabajo idóneo para buscar la solución del Funcional Mumford-Shah como un problema de evolución de curvas, en donde intentamos buscar las mejores curvas que cumplan con el criterio del funcional.
 - El Teorema de los Cuatro Colores, de la teoría de grafos, y su aplicación en el coloreamiento de mapas, han permitido que nuestra formulación

del Funcional Mumford-Shah con Conjuntos de Nivel sólo emplee dos funciones de conjuntos de nivel para describir cualquier imagen, tomando en cuenta que se generarán 4 fases. Vale la pena señalar que una fase puede contener a varios objetos, por lo que sólo se necesitan 4 fases para segmentar cualquier tipo de imagen.

- Un aspecto muy importante de la evolución de nuestra solución en las Ecuaciones Diferenciales Parciales es la Condición de Neumann. Esta condición describe el comportamiento de las funciones de conjuntos de nivel en la frontera, es decir en los bordes de la imagen. La buena implementación de esta condición durante todo el proceso de evolución permiten que los resultados obtenidos sean buenos.
- El efecto de los parámetros que controlan la longitud de los bordes (μ_1 y μ_2) es muy importante, ya que son ellos los que controlan si la segmentación contiene más detalles o sólo representan un bosquejo simple sin muchas características. Dependerá mucho del tipo de imagen que se quiere segmentar para poder asignarle un valor adecuado a estos parámetros.
- Nuestro algoritmo es sensible a imágenes con bajo contraste y pobres condiciones de iluminación. Planteamos que las imágenes que presentan estos problemas deberían antes pasar por un pre-procesamiento, ya sea de mejoramiento de contraste o de corrección de luz. Resulta casi imposible pensar que el algoritmo pueda reconocer una sombra o un brillo en un objeto.
- La aplicación del Problema de Partición Mínimo ha permitido que los resultados obtenidos sean imágenes simples, pero sin perder las características esenciales de las originales. Imágenes segmentadas simples harán que

un posterior proceso de análisis de imágenes o visión computacional requieran menos trabajo y sean más exactos.

- Los tiempos obtenidos con nuestro algoritmo son superiores a los de los algoritmos clásicos. Pero debemos tener en cuenta que el algoritmo multi-umbral empleado en este trabajo sólo segmenta 3 regiones, el algoritmo por crecimiento de regiones no ofrece resultados generales para posteriores procesos y el algoritmo Split-Merge produce imágenes un poco menos o igual de complejas que las originales. Las imágenes resultantes con nuestro algoritmo ofrecen una buena segmentación de los objetos significativos de la escena, los cuales pueden ser usados para extraer características en procesos posteriores. Además, nuestras imágenes segmentadas son lo suficientemente simples como para apoyar el desempeño de posteriores procesos.
- Nuestro algoritmo es poco sensible al ruido, debido a los parámetros que controlan la longitud de los bordes. Cuando μ es grande, el algoritmo segmenta aceptablemente los objetos en presencia de ruido. Esto es debido a que cuando μ es grande, el algoritmo no toma en cuenta los detalles de la imagen, dejando de lado los puntos aislados que conforman el ruido.
- Nuestro algoritmo es aplicable a muchas tareas. Hemos demostrado que puede emplearse para apoyar la tarea de profesionales como médicos, haciendo las imágenes más simples pero preservando las características de la imagen, como forma y tamaño. También en el reconocimiento de rostros, y sobretodo, en la etapa de extracción de características, ya que preserva todas las características faciales importantes en este tipo de aplicaciones. Entre otras aplicaciones posibles están el reconocimiento óptico de caracteres y la visión computacional aplicada a procesos industriales.

Apéndice A

Aspectos Matemáticos

En este apéndice daremos a conocer todos los aspectos matemáticos importantes que empleamos en el desarrollo de este trabajo.

A.1. Cálculo Variacional

A.1.1. Definición de Funcional

Sea M una clase de funciones $y(x)$. Si a toda función $y(x) \in M$ le corresponde, según una regla, un número determinado J se dice que en la clase M está definida la funcional J y se escribe $J = J[y(x)]$.

La clase M de funciones $y(x)$ en la que está definida la funcional $J[y(x)]$ se denomina campo de definición de la funcional.

A.1.2. Extremo de un Funcional

El problema fundamental del cálculo variacional radica en extremar un funcional, esto significa encontrar una función que cumpla con ser un extremo débil de dicho funcional, es decir, que de entre todas las funciones que satis-

fagan las condiciones de frontera de nuestro problema $y(a) = A$ e $y(b) = B$ hay que hallar una función que maximice o minimice un funcional, tal como:

$$J(y(x)) = \int_a^b F(x, y, y')$$

Pero ahora surge la pregunta ¿Para qué valores una función es máxima o mínima?. La condición necesaria para considerar a un punto como extremo local de una función diferenciable $y = y(x)$ es:

$$\frac{\partial y}{\partial x} = 0,$$

Si todos los puntos en x que cumplen con este requerimiento se denominan puntos estacionarios, la ecuación de Euler-Lagrange provee la condición necesaria para establecer la estacionaridad de un funcional, pero antes de realizar ello debemos describir el teorema fundamental del cálculo variacional que será útil en un posterior desarrollo [20, 12].

A.1.3. Teorema Fundamental del Cálculo Variacional

El siguiente teorema está íntimamente relacionado a muchas expresiones de tipo variacional.

Teorema A.1.1. *Sea la función continua¹ $n \in \mathcal{C}^0$ se cumple que:*

$$\int_a^b n(x)m(x) = 0; \forall m(x) \in \mathcal{C}^0 \quad (\text{A.1})$$

donde $n(x) \equiv 0$, para $a < x < b$.

¹Esta notación expresa el grado de continuidad de una función n . Si n es continua, nosotros decimos que $n \in \mathcal{C}^0$ siendo \mathcal{C}^0 el espacio de funciones continuas. Además, \mathcal{C}^1 es el espacio de funciones cuya primera derivada es continua, y así sucesivamente.

Demostración. Probaremos este teorema por contradicción, diciendo:

Si $n(x_0) > 0$ para algún $x_0 \in [a, b]$. Sabiendo que $n(x)$ es según el enunciado de arriba continuo, entonces existe un vecindario $U = (x_0 - \epsilon, x_0 + \epsilon)$ de x_0 cuando $n(x) > 0$. Ahora seleccionando una función $m(x)$ tal que $m(x) > 0$ para $x \in (x_0 - \epsilon, x_0 + \epsilon)$ y $m(x) = 0$ para todos los otros $x \in [a, b]$. Entonces $\int n(x)m(x) > 0$, lo cual es una contradicción. De manera similar puede analizarse cuando $n(x) < 0$ [20, 11]. \square

A.1.4. Ecuación de Euler-Lagrange

Ahora, volviendo sobre el problema de encontrar una función $y(x) \in \mathcal{V}$ que extrema al funcional $J(y)$, donde \mathcal{V} define un espacio de funciones suaves. Aquí es necesario indicar que no todas las funciones $y(x)$ son aptas para nuestra búsqueda de un extremo, aquí entran a tallar las condiciones de frontera justamente para restringir estos valores de $y(x)$ con respecto a un intervalo $[x_1, x_2]$.

Cual función $y(x)$ minimiza el funcional:

$$J = J(y) = \int_{x_1}^{x_2} F(x, y, y') dx, \quad (\text{A.2})$$

de acuerdo a las condiciones de frontera

$$y(x_1) = y_1 \quad y(x_2) = y_2 \quad (\text{A.3})$$

Si suponemos que la función $y(x)$ minimiza el funcional J , entonces nosotros procederemos a expandir la función $y(x)$ utilizando una función auxiliar $h(x) \in \mathcal{C}^1$ definida también bajo el mismo intervalo $[x_1, x_2]$ y tal que $h(x_1) = h(x_2) = 0$. Si ahora consideramos también un pequeño parámetro positivo ϵ , entonces la

función queda expresada así:

$$y(x) + \epsilon h(x) \quad (\text{A.4})$$

La función anterior (A.4) se denomina función de desviación, el término $\epsilon h(x)$ representa pequeños cambios en la función $y(x)$ sobre el intervalo $[x_1, x_2]$, por eso se le llama la variación de $y(x)$, esto se denota como $\partial y(x) := \epsilon h(x)$. La función de desviación $y(x) + \epsilon h(x)$ siempre pasa a través de los valores de frontera estipulados en (A.3), desde que $\epsilon(x_1) = \epsilon(x_2) = 0$.

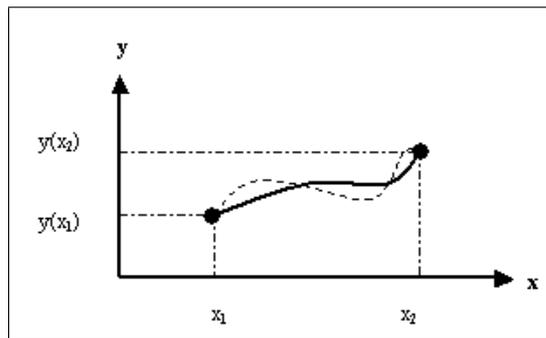


Figura A.1: Se observa que la función $\epsilon h(x)$ es añadida como variación a la función $y(x)$ con valores de bordes fijos. La función de desviación se denota como $y + \epsilon h$.

Si consideramos a $y(x)$ y $h(x)$ como funciones fijas entonces $J(y + \epsilon h)$ evoluciona dependiendo de ϵ podríamos denotar esto así $J(\epsilon)$. Esta función $J(\epsilon)$ debería ser mínima si hacemos que $\epsilon = 0$. Entonces

$$\forall h(x) : J(y + \epsilon h) \text{ es mínimo para } \epsilon = 0,$$

Si la condición anterior no se cumple para cualquier $h_0(x)$, podríamos optar por elegir un valor $\epsilon \neq 0$ para hacer que $J(y + \epsilon h)$ sea más pequeño que $J(y)$, es decir $J(y + \epsilon h) < J(y)$ esto parece contradictorio, pero deja entre ver que minimizar el funcional depende directamente del valor que asuma ϵ por lo tanto

se debe evaluar J con respecto a este parámetro. Esta ecuación se conoce como la condición de estacionaridad :

$$\left(\frac{dJ(y + \epsilon h)}{d\epsilon} \right)_{\epsilon=0}, \forall h(x) \quad (\text{A.5})$$

La ecuación (A.5) se considera una condición necesaria para determinar un extremo local. Ahora, reemplazando la ecuación (A.4) en la ecuación(A.2), la ecuación reescrita quedaría así:

$$J(y + \epsilon h) = \int_{x_1}^{x_2} F(x, y + \epsilon h, y' + \epsilon h') dx. \quad (\text{A.6})$$

Después aplicando un proceso de expansión mediante las series de Taylor en $J(y + \epsilon h)$ acerca de $F(x, y, y')$ obtendremos lo siguiente:

$$\begin{aligned} J(y + \epsilon h) = & \int_{x_1}^{x_2} F(x, y, y') dx + \frac{\partial F}{\partial x}(x, y, y') dx + \frac{\partial F}{\partial y}(x, y, y') \epsilon h(x) dx \\ & + \frac{\partial F}{\partial y'}(x, y, y') \epsilon h'(x) dx \end{aligned} \quad (\text{A.7})$$

Teniendo esta ecuación procedemos a omitir términos de orden ϵ^2 o más grande, después de realizado esto tenemos:

$$\left(\frac{dJ(y + \epsilon h)}{d\epsilon} \right)_{\epsilon=0} = \int_{x_1}^{x_2} \frac{\partial F(x, y, y')}{\partial y} h(x) + \frac{\partial F(x, y, y')}{\partial y'} h'(x)$$

De acuerdo a la condición de estacionaridad establecida en la ecuación (A.5), toda la expresión obtenida en la ecuación de arriba debe ser igual a cero. Entonces tenemos:

$$0 = \left(\frac{dJ(y + \epsilon h)}{d\epsilon} \right)_{\epsilon=0}$$

$$\int_{x_1}^{x_2} \left(\frac{\partial F}{\partial y}(x, y, y')h(x) + \frac{\partial F}{\partial y'}(x, y, y')h'(x) \right) dx$$

Para desarrollar esta integral consideraremos que el primer término se vuelve cero ya que al integrar en el intervalo $[x_1, x_2]$ los valores de estos corresponden a los de las condiciones de frontera que son $n(x_1) = n(x_2) = 0$. El segundo término lo trabajaremos aplicando integración por partes generando lo siguiente:

$$\int \left(\frac{\partial F}{\partial y}(x, y, y') - \frac{\partial}{\partial x} \frac{\partial F}{\partial y'}(x, y, y') \right) h(x) dx$$

Podemos hacer que la expresión de arriba sea equivalente a tener $(F(x, y, y')h(x)) dx$, entonces será posible aplicar el teorema fundamental del cálculo variacional haciendo que $(F(x, y, y')h(x)) dx = 0$, es decir:

$$\frac{\partial F}{\partial y}(x, y, y') - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial y'}(x, y, y') \right) = 0 \quad (\text{A.8})$$

Finalmente, la ecuación obtenida es la ecuación de Euler-Lagrange, la cual determina una ecuación diferencial ordinaria de segundo orden que permite encontrar la función $y(x)$. Si la función $y(x)$ satisface esta ecuación, el funcional $J(y(x))$ es estacionario [20, 28].

A.2. Fórmula de Green

Teorema A.2.1. *Sea $u, v \in C^2(\Omega)$. Entonces:*

$$\int_{\Omega} \nabla v \cdot \nabla u \, dx = - \int_{\Omega} u \Delta v \, dx + \int_{\partial\Omega} \frac{\partial v}{\partial n} u \, ds.$$

A.3. Coloreamiento de Mapas

El Teorema de los Cuatro Colores está íntimamente ligado con el coloreamiento de mapas [35]. Sea un mapa conteniendo diversos países, se desea conocer cuantos colores serán necesarios para colorear el mapa completo de tal forma que dos países fronterizos no queden coloreados del mismo color. Este enunciado constituye uno de los problemas más clásicos del teorema de los cuatro colores, es conveniente, sin embargo, indicar que este mapa debe ser un grafo planar,² y k -colorable³.

Aplicando el teorema de los cuatro colores para mapas, entonces podemos establecer que cada mapa, que cumpla lo dicho en el párrafo anterior, es 4 -colorable.

No ilustraremos la prueba de este teorema, ya que ésta es bastante extensa y complicada, por eso sólo nos limitaremos a afirmar que cada grafo planar es 4 -colorable.

²grafo planar, es aquel grafo el cual puede ser dibujado en el plano de tal forma que todos sus arcos pueden representarse por líneas rectas, sin que estas se crucen entre ellas.

³Un grafo es K -colorable, si es un grafo sin loops, y además si a cada uno de sus nodos podemos asignarle uno de los k colores, de tal forma que dos nodos adyacentes no tengan el mismo color.

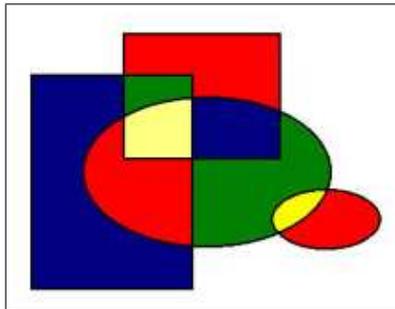


Figura A.2: Figura que muestra un la coloración de un gráfico *4-colorable*.

Bibliografía

- [1] Ambrosio, L. *A compactness theorem for a new class of functions of bounded variation*. Bolletino della Unione Matematica Italiana. 1989.
- [2] Ambrosio, L., Tortorelli, V. *Approximation of Functionals depending on jumps by elliptic functionals via Γ -Convergence*. Communications on Pure and Applied Mathematics. 1990.
- [3] Ambrosio, L., Tortorelli, V. *On the Approximation of Free Discontinuity Problems*. Department of Mathematics. University of Bollogna. 1992.
- [4] Aubert, G., Kornprobst, P. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer-Verlag. 2002.
- [5] Blake, A., Zisserman, A. *Visual Reconstruction*. MIT Press. 1987.
- [6] Brook, A., Kimmel, R., Nir, A. *Variational Segmentation for Color Images*. Dept. of Mathematics, Technion, Israel. Dept. of Computer Science, Technion, Israel.
- [7] Chambolle, A. *Inverse Problem in Image Processing and Image Segmentation: some mathematical and numerical aspects*. Trieste Ed. 2000.
- [8] Chakraborty, A. *Feature and Module Integration for Image Segmentation*. Ph.D.Dissertation. Graduate School. Yale University. 1996.

-
- [9] Chan, T., Vese, L. *Active Contours Without Edges*. IEEE Transactions on Image Processing, Vol. 10, No. 2. 2001.
- [10] Chan, T., Vese, L. *A Level Set Algorithm for Minimizing the Mumford-Shah Functional in Image Processing*. IEEE Workshop on Variational and Level Set Methods, pp. 161-168, Vancouver, Canadá. 2001.
- [11] Craggs, J. *Cálculo de Variaciones*. George Allen & Unwin Ltd. 1973.
- [12] Elsgoltz, L. *Ecuaciones Diferenciales y Cálculo Variacional*. Editorial MIR. Moscú. 1969.
- [13] Garrido, L. *Hierarchical Region Based Processing of Images and Video Sequences: Application to Filtering, Segmentation and Information Retrieval*. Ph.D. Dissertation. Department of Signal Theory and Communications. Universitat Politecnica de Catalunya. 2002.
- [14] Geman, S., Geman, D. *Stochastic Relaxation, Gibbs Distribution and the Bayesian Restoration of Images*. Division of Applied Mathematics. Brown University. 1984.
- [15] Gonzalez, R., Woods, R. *Digital Image Processing*. 2nd Edition. Prentice Hall. 2002.
- [16] Hewer, G., Kenney, C., Manjunath, B. *Variational image segmentation using boundary functions*. IEEE Transactions on Image Processing, Vol. 7, NO. 9, 1998.
- [17] Jähne, B. *Digital Image Processing*. 5th Edition. Springer. 2002.
- [18] Kato, Z. *Mumford-Shah: Energy Functional*. Course on Variational and Level Set Methods in Image Processing. 2002.

- [19] Kato, Z. *Snakes: Active Contours*. Course on Variational and Level Set Methods in Image Processing. 2002.
- [20] Krasnov, M., Makarenko, G., Kiseliiov, A. *Cálculo Variacional*. Editorial MIR. Moscú. 1976.
- [21] Maintz, T. *Digital and Medical Image Processing*. 2002.
- [22] Mumford, D., Shah, J. *Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems*. Communications on Pure and Applied Mathematics. John Wiley y Sons, Inc. 1989.
- [23] Osher, S. *Level Set Method: Applications to Imaging Science*. Mathematics Department. UCLA. Los Angeles, California. 2003.
- [24] Osher, S., Paragios, N. *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Springer-Verlag, New York. 2003.
- [25] Petia, R., Martí, E., . *Facial Features Segmentation by Model-Based Snakes*. Departamento d'Informàtica. Facultat de Ciències. Universitat Autònoma de Barcelona. 2000.
- [26] Pratt, W. *Digital Image Processing: PIKS Inside*. John Wiley and Sons Inc. 2001.
- [27] Rudin, L., Osher, S., Fatemi, E. *Nonlinear total variation based noise removal algorithms*. Elsevier Science Publishers. 1992.
- [28] Russak, I. *Calculus of Variations: Lecture Notes*. Department of Mathematics. Naval Postgraduate School. California 2002.
- [29] Seeman, T. *Digital Image Processing using Local Segmentation*. School of Computer Science and Software Engineering. Monash University. 2002.

-
- [30] Sethian, J. *Level Set Methods: An Act of Violence*. Applied and Computational Mathematics Department. Berkeley University. 1996.
- [31] Sethian, J. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press. 1996.
- [32] Shah, J. *Properties of Energy-Minimizing Segmentations*. Mathematics Department, Northeastern University, Boston . National Science Foundation. 1991.
- [33] Shapiro, L., Stockman, G. *Computer Vision*. Prentice Hall. 2000.
- [34] Vese, L., Chan, T. *Image Segmentation using Levels Sets and the Piecewise Constant Mumford-Shah Model*. Kluwer Academic Publishers. 2000.
- [35] Wilson, R. *Introduction to Graph Theory*. Longman Scientific and Technical. 1987.
- [36] Zhu, W., Chan, T., Esedoglu, S. *Segmentation with Depth: A Level Set Approach*. Department of Mathematics, UCLA. Los Angeles, California. 2004.