

Modelos de Contornos Activos para la Segmentación de Parásitos en Imágenes de Muestras de Sangre

por

L.M. Juan Antonio Ríos Briceño



Universidad Autónoma
de Yucatán

Facultad de Matemáticas

Tesis presentada para obtener el grado de
Maestro en Ciencias Matemáticas.

Director de Tesis

Dr. Carlos Francisco Brito Loeza

Mérida, Yucatán, México
Octubre 2013

Dedicatorias

A Dios:

Por todas las oportunidades que me ha brindado, cuidarme y guiarme a lo largo de mi vida.

A mis padres:

Juan Melchor Ríos Xool y María Guadalupe Briceño Sosa, por todos los esfuerzos y sacrificios que han hecho por mí y por todo su apoyo a lo largo de mi vida, los cuales no podré pagar ni con todo el dinero del mundo.

A mi esposa:

Jessica Aracely Cervera Yah, por ser mi motivación, mi compañera y mi amiga incondicional, por todo su amor, su cariño y sus cuidados, los cuales siempre tendré presentes.

A mis hermanitos:

Edwin Roberto, Moises Alejandro y Daniel Eduardo, por todos los ratos de diversión y alegría que han vivido conmigo y por ser partícipes de todos mis logros.

Con todo mi aprecio, cariño y amor:

Tony

Agradecimientos

A toda mi familia:

Por todos sus consejos y apoyo que me han brindado, los cuales siempre tendré presentes.

A mis suegros:

Quienes me han compartido parte de sus vivencias y han sido un gran apoyo en esta etapa de mi vida.

A mi asesor:

Dr. Carlos Francisco Brito Loeza por todo su apoyo y motivación en la elaboración de esta tesis de Maestría, por permitirme conocer más acerca de esta área de investigación y por la ayuda en mi camino hacia el doctorado.

Al Dr. Ricardo Legarda Sáenz por haberme enseñado importantes conocimientos sobre la librería CUDA y por sus valiosas sugerencias y comentarios acerca de esta tesis.

Al Dr. José Matías Navarro Soza por haberme enseñado importantes conocimientos en las clases de Geometría Diferencial y por la revisión de la presente tesis.

A los profesores que han contribuido en mi formación académica a lo largo de la Maestría: Dr. Didier Solís, Dr. Carlos Brito, Dr. Ricardo Legarda, Dr. Víctor Bautista, Dr. Matías Navarro, Dr. Jorge Lugo, Dr. Gabriel Murrieta, Dr. Gerardo Almeida y Dr. Aarón Aguayo.

A todos mis compañeros de posgrado, amigos de la Olimpiada de Matemáticas y amigos que me acompañaron y apoyaron a lo largo de mis estudios, en especial a: Cristian Oliva, Manuel Novelo, Irving Chan y César Cobos.

¡Muchas Gracias!

Resumen

La detección de parásitos en frotis de sangre teñidas, es muy importante para el tratamiento de enfermedades como la malaria, el chagas, la toxoplasmosis y la leishmaniasis. La automatización de este proceso ayudaría en la reducción del tiempo tomado para diagnósticos y de la posibilidad de errores humanos. Sin embargo, la variabilidad y los objetos presentes en las imágenes microscópicas de muestras de sangre plantean desafíos significativos para la detección precisa. En este trabajo se prueban algunos modelos de contornos activos en la segmentación de parásitos en imágenes de muestras de sangre, los cuales se caracterizan por evolucionar una curva hasta que ésta minimice un funcional de energía, estos modelos han demostrado ser una potente herramienta en el área de segmentación de imágenes debido a su robustez y su flexibilidad en la formulación. La implementación se realizó usando el método de conjuntos de nivel, cuya ventaja principal es que la energía no depende de la parametrización de la curva y admite cambios en su topología. Se presentan los resultados obtenidos de cada modelo y se muestran las ventajas de la paralelización en GPUs para este tipo de algoritmos.

Índice general

| | |
|------------------------------------------------------------------------------|-----------|
| Dedicatorias | I |
| Agradecimientos | II |
| Resumen | III |
| 1. Introducción | 1 |
| 1.1. Segmentación de imágenes | 2 |
| 1.2. Modelos de contornos activos para la segmentación de imágenes | 3 |
| 1.3. Justificación y alcances | 7 |
| 1.4. Estructura de la tesis | 8 |
| 2. Preliminares Matemáticos | 10 |
| 2.1. Cálculo variacional | 10 |
| 2.1.1. Integrales variacionales extendidas | 15 |
| 2.2. Modelos de contornos activos | 19 |
| 2.2.1. Formulación en conjuntos de nivel | 19 |
| 2.2.2. Minimización | 20 |
| 2.2.3. Método numérico | 21 |
| 3. Modelo de Contornos Activos sin Bordes | 23 |
| 3.1. Formulación en conjuntos de nivel | 25 |
| 3.2. Minimización | 25 |
| 3.3. Método numérico | 27 |
| 3.4. Resultados | 28 |
| 3.5. Discusión de resultados | 32 |

ÍNDICE GENERAL

| | |
|-----------------------------------------------------------------------|-----------|
| 4. Minimización de la Energía de Ajuste de la Región Escalable | 33 |
| 4.1. Formulación en conjuntos de nivel | 36 |
| 4.2. Minimización | 37 |
| 4.3. Método numérico | 38 |
| 4.4. Resultados | 40 |
| 4.5. Discusión de resultados | 43 |
| 5. Contornos Activos para la Segmentación de Distribuciones | 44 |
| 5.1. Formulación en conjuntos de nivel | 46 |
| 5.2. Minimización | 46 |
| 5.3. Método numérico | 47 |
| 5.4. Resultados | 48 |
| 5.5. Discusión de resultados | 48 |
| 6. Comentarios finales | 50 |
| 6.1. Resultados de la paralelización | 50 |
| 6.2. Conclusiones | 51 |
| 6.3. Trabajos futuros | 52 |
| A. Paralelización | 53 |
| A.1. Historia | 53 |
| A.2. CUDA | 54 |
| A.3. Programando en CUDA | 55 |
| A.3.1. Kernel | 55 |
| A.3.2. Invocación a un kernel | 56 |
| A.3.3. Gestión de la memoria | 56 |
| A.3.4. Sincronización y memoria compartida | 57 |
| A.4. Thrust | 57 |
| A.4.1. Reducciones | 58 |
| Bibliografía | 59 |

Índice de figuras

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.1. El mínimo se obtiene cuando la curva coincide con el contorno del objeto . . | 24 |
| 3.2. Detección de objetos en imágenes con ruido, el número de iteraciones para esta imagen fue de 90 | 28 |
| 3.3. Detección de objetos en imágenes con ruido, el número de iteraciones para esta imagen fue de 461 | 29 |
| 3.4. Segmentación del tumor de un cerebro, el número de iteraciones para esta imagen fue de 80 | 29 |
| 3.5. Segmentación de un caballo, el número de iteraciones para esta imagen fue de 225 | 30 |
| 3.6. Falla al segmentar la materia gris de un cerebro, el número de iteraciones para esta imagen fue de 157 | 30 |
| 3.7. Falla al segmentar vasos sanguíneos, el número de iteraciones para esta imagen fue de 130 | 31 |
| 3.8. Falla al segmentar el Trypanosoma Cruzi, el número de iteraciones para esta imagen fue de 847 | 31 |
| 4.1. Al segmentar en una región de x , el valor del pixel en el punto y debe aportar más en la segmentación, que el valor en el pixel z | 33 |
| 4.2. El término \mathcal{E}_x^{Aju} es como la energía de Chan-Vese solo que con ponderaciones, aplicado a una región de x | 35 |
| 4.3. El contorno final va a ser el que mejor se ajuste a los contornos de todas las segmentaciones locales centradas en cada pixel de la imagen. | 36 |
| 4.4. Segmentación de la materia gris de un cerebro, el número de iteraciones para esta imagen fue de 269 | 40 |

ÍNDICE DE FIGURAS

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 4.5. Segmentación de vasos sanguíneos, el número de iteraciones para esta imagen fue de 237 | 41 |
| 4.6. Segmentación del Trypanosoma Cruzi, el número de iteraciones para esta imagen fue de 71 | 41 |
| 4.7. Segmentación del Trypanosoma Cruzi en varias muestras de sangre | 42 |
| 4.8. Segmentación de la leishmania (a), plasmodium (b) y el toxoplasma gondii(d) y (c) | 42 |
| 5.1. Histograma de frecuencias e histograma con el método de Parzen | 45 |
| 5.2. Ejemplos de segmentación usando la distancia de Bhattacharyya para 3 imágenes sintéticas (a), (b) y (c), y para el parásito toxoplasma gondii (d) | 48 |

Capítulo 1

Introducción

Una de las capacidades más impresionantes del cerebro humano es la Visión, el poder inferir cosas del mundo exterior basados en la información que llega a nuestros ojos. Poder determinar distancias, colores, texturas, orientación y reconocer objetos, personas, animales, etc. La Visión Computacional es una rama de la inteligencia artificial que trata de emular esta capacidad en las computadoras, mediante la interpretación de las imágenes adquiridas, por ejemplo con una cámara.

Existen varias definiciones para la Visión, entre ellas:

- “Visión es saber que hay y dónde mediante la vista”, Aristóteles.
- “Visión es recuperar de la información de los sentidos (vista) propiedades válidas del mundo exterior”, Gibson.
- “Visión es un proceso que produce a partir de las imágenes del mundo exterior una descripción que es útil para el observador y que no tiene información irrelevante”, Marr.

De éstas, la definición de Marr nos da una idea más clara de lo que es la Visión computacional y su objetivo, solo hay que tener presente que el proceso es computacional, la descripción a obtener depende de los objetivos del proceso y de las características que debe satisfacer la imagen, y es necesario eliminar la información que no sea útil.

Un área muy ligada con la visión es el procesamiento de imágenes, aunque tienen muchas cosas en común, el objetivo final es diferente, el objetivo del procesamiento de imágenes es mejorar la calidad de las imágenes para su posterior utilización o interpretación. Por ejemplo:

1.1 Segmentación de imágenes

- Remover defectos.
- Remover problemas por movimiento o desenfoque.
- Mejorar ciertas propiedades como color, contraste, estructura, etc.
- Agregar “colores falsos” a imágenes monocromáticas.

1.1. Segmentación de imágenes

La segmentación de imágenes es uno de los problemas fundamentales en la visión computacional y el procesamiento de imágenes, es un componente esencial para diversas aplicaciones, por ejemplo los sistemas de visión automatizada y aplicaciones médicas. Su objetivo es encontrar una partición de la imagen en un número finito de regiones semánticamente importantes.

Varios modelos matemáticos han sido desarrollados para lograr la segmentación de una imagen. Los modelos más prometedores están basados en enfoques variacionales y ecuaciones diferenciales parciales. Estos modelos se benefician de teorías matemáticas bien fundamentadas, que nos permite analizar, entender, mejorar los métodos existentes y trabajar en un ambiente continuo.

Esta tesis se centra en modelos de contornos activos para la segmentación de imágenes, cuya característica común es que definen una segmentación óptima como un minimizador de una función objetivo que generalmente depende de la imagen dada y de las características para identificar las diferentes regiones segmentadas. La ecuación de Euler-Lagrange de estos modelos a menudo se puede describir utilizando una ecuación diferencial parcial parabólica, la cual es iterada hasta que alcanza el estado estacionario.

En estos modelos un contorno o curva es introducido para localizar las fronteras de las características, envolviéndolas hasta alcanzar el estado estacionario, dividiendo la imagen en regiones. Un método muy potente y popular para representar el contorno, es el método de conjuntos de nivel, originalmente desarrollado por Osher y Sethian [92], el cual representa el contorno implícitamente como un conjunto de nivel particular (usualmente el cero). La

principal ventaja de esta representación es que los cambios topológicos, tales como la fusión y división de contornos pueden ser capturadas naturalmente a través de cambios suaves a la función de conjunto de nivel.

1.2. Modelos de contornos activos para la segmentación de imágenes

Varios modelos de contornos activos han sido propuestos para extraer objetos de interés en imágenes, tales como el bien conocido y exitoso modelo de los contornos activos (o snakes), propuesto inicialmente por Kass, Witkin y Terzopoulos en [1]. Este método tiene numerosas aplicaciones en el procesamiento de imágenes, por ejemplo en imágenes médicas para extraer estructuras anatómicas [2], [3], [4].

El enfoque clásico de los snakes [1] está basado en la deformación de un contorno inicial C_0 hasta el contorno del objeto a detectar. La deformación se obtiene tratando de minimizar un funcional diseñado de manera que su mínimo (local) se obtiene en el límite del objeto. Estos contornos activos son ejemplos de la técnica general de hacer coincidir modelos deformables a datos de la imagen mediante minimización de energía [5].

Sea $C(q) : [0, 1] \rightarrow \mathbb{R}^2$ una curva plana parametrizada y sea $u_0 : [0, a] \times [0, b] \rightarrow \mathbb{R}^+$ una imagen dada en la cual queremos detectar los bordes de los objetos. El enfoque clásico de snakes [1] asocia a la curva C con una energía dada por:

$$E(C) = \alpha \int_0^1 |C'(q)|^2 dq + \beta \int_0^1 |C''(q)|^2 dq - \lambda \int_0^1 |\nabla u_0(C(q))| dq$$

donde α , β y λ son constantes reales. Los primeros dos términos controlan la suavidad del contorno a ser detectado (energía interna), mientras el tercer término es el responsable de atraer el contorno hacia el objeto en la imagen (energía externa).

Este modelo de energía no es capaz de manejar los cambios en la topología de la evolución del contorno, cuando se realizan implementaciones directas. Por lo tanto, la topología de la

1.2 Modelos de contornos activos para la segmentación de imágenes

curva final será como la de C_0 (la curva inicial). Éste es un problema cuando uno no conoce el número de objetos que deben ser simultáneamente detectados. Este enfoque también es no intrínseco, ya que la energía depende de la parametrización de la curva y no está directamente relacionado a la geometría de los objetos.

Siguiendo el primer modelo de contornos activos, Caselles, Catté, Coll, Dibos en [6] propusieron una nueva mejora basada en la teoría de evolución de curvas y flujo geométrico, en estos contornos activos, la curva es propagada por medio de una velocidad que contiene dos términos, uno relacionado a la regularidad de la curva y el otro que lo encoge o lo expande hasta el límite. El modelo está dado por un flujo geométrico expresado por una ecuación diferencial parcial, basado en la evolución de curvatura media. Este modelo está motivado por el enfoque de la evolución de curvatura y no en una minimización de energía, lo que permite cambios automáticos en la topología cuando implementamos usando el algoritmo numérico basado en conjuntos de nivel [7]. Así, varios objetos pueden ser detectados simultáneamente sin previo conocimiento del número exacto en la escena y sin usar procedimientos de rastreo.

En los siguientes snakes y modelos de contornos activos, un detector de bordes es usado para detener la evolución de la curva en la frontera del objeto deseado. Usualmente, éste es una función de borde positiva, decreciente y regular $g(|\nabla u_0|)$, tal que $\lim_{t \rightarrow \infty} g(t) = 0$. Por ejemplo,

$$g(|\nabla u_0|) = \frac{1}{1 + |\nabla G_\sigma * u_0|^p}, p \geq 1,$$

donde $G_\sigma * u_0$, es una versión más suave de u_0 , es la convolución de la imagen u_0 con el Gaussiano $G_\sigma(x, y) = \sigma^{-1/2} e^{-(x^2+y^2)/4\sigma}$. La función $g(|\nabla u_0|)$ es estrictamente positiva en regiones homogéneas, y cercana a cero en los bordes.

El modelo basado en la evolución de curvatura media, está dado por la siguiente ecuación de evolución:

$$\begin{cases} \frac{\partial \phi}{\partial t} = g(|\nabla u_0|) |\nabla \phi| \left(\operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \nu \right), \\ \text{en } [0, \infty[\times \mathbb{R}^2 \\ \phi(0, x) = \phi_0(x) \text{ en } \mathbb{R}^2, \end{cases} \quad (1.1)$$

donde $g(|\nabla u_0|)$ es la función borde definida anteriormente con $p = 2$, y ν es una constante positiva. ϕ_0 es la función de conjunto de nivel inicial. Su curva de nivel cero se mueve en la

1.2 Modelos de contornos activos para la segmentación de imágenes

dirección normal con la velocidad

$$g(|\nabla u_0|) \left(\operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \nu \right)$$

y por tanto para en la frontera deseada, donde g se vuelve cero. La constante ν es un término de corrección elegido tal que $\operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \nu$ se mantenga siempre positiva. Esta constante puede ser interpretada como una fuerza de empuje de la curva hacia el objeto, cuando la curvatura se vuelve nula o negativa (por otro lado, la presencia de la constante ν es una restricción en la zona dentro de la curva, y esto también aumenta la propagación velocidad).

Caselles, Kimmel y Sapiro en [8] y Kichenassamy, Kumar, Olver, Tannenbaum y Yezzi en [9] propusieron una nueva versión de mejora del modelo snake llamado modelo del contorno de la geodésica activa (del contorno geométrico activo). Estas nuevas formulaciones se dicen geoméricamente intrínsecas porque la energía snake propuesta es invariante con respecto a la parametrización de la curva. Este modelo es definido por el siguiente problema de minimización:

$$\inf_C J_2(C) = 2 \int_0^1 |C'(s)| \cdot g(|\nabla u_0(C(s))|) ds. \quad (1.2)$$

Éste es un problema de cálculo de la geodésica en un espacio de Riemman, de acuerdo a una métrica inducida por la imagen u_0 .

El modelo del contorno de la geodésica activa [8] tiene también una formulación en conjuntos de nivel:

$$\begin{cases} \frac{\partial \phi}{\partial t} = |\nabla \phi| \left(\operatorname{div} \left(g(|\nabla u_0|) \frac{\nabla \phi}{|\nabla \phi|} \right) + \nu g(|\nabla u_0|) \right), \\ \text{en } [0, \infty[\times \mathbb{R}^2 \\ \phi(0, x) = \phi_0(x) \text{ en } \mathbb{R}^2. \end{cases} \quad (1.3)$$

Aquí también la constante ν es agregado para incrementar la velocidad de evolución y atrae a la curva hacia el límite y constituye en realidad un área adicional basada en la velocidad.

Todas estas mejoras usan la función de borde g , dependiendo del gradiente de la imagen $|\nabla u_0|$ para parar la evolución de la curva, estos modelos solo pueden detectar con bordes definidos por gradiente. Para resolver este problema Tony F. Chan y Luminita A. Vese proponen en [10] el modelo de contornos activos sin bordes, en el cual usan un término de paro basado en

1.2 Modelos de contornos activos para la segmentación de imágenes

técnicas de segmentación de Mumford-Shah [11] y no una función definida por gradiente, por lo cual se pueden obtener contornos con o sin gradiente, por ejemplo objetos con fronteras muy suaves o incluso con fronteras discontinuas. Este modelo también tiene una formulación en conjuntos de nivel:

$$\left\{ \begin{array}{l} c_1(\phi) = \frac{\int_{\Omega} u_0 H_{\varepsilon}(\phi) dx dy}{\int_{\Omega} H_{\varepsilon}(\phi) dx dy}, \quad c_2(\phi) = \frac{\int_{\Omega} u_0 (1 - H_{\varepsilon}(\phi)) dx dy}{\int_{\Omega} (1 - H_{\varepsilon}(\phi)) dx dy}, \\ \frac{\partial \phi}{\partial t} = \delta_{\varepsilon}(\phi) \left[\mu \rho \left(\int_{\Omega} \delta_{\varepsilon}(\phi) |\nabla \phi|^{\rho-1} \right) \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu \right. \\ \left. - \lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2 \right] \quad \text{en } \Omega, \\ \phi(t, x, y) = \phi_0(x, y) \quad \text{en } \Omega, \\ \rho \left(\int_{\Omega} \delta_{\varepsilon}(\phi) |\nabla \phi|^{\rho-1} \right) \frac{\delta_{\varepsilon}(\phi)}{|\nabla \phi|} \frac{\partial \phi}{\partial n} = 0 \quad \text{en } \partial \Omega. \end{array} \right.$$

Luego de este planteamiento, en 1999 Tony F. Chan, B. Yezriev Sandberg, y Luminita A. Vese propusieron en [12] una extensión al caso vector valuado, se propone un algoritmo de contornos activos para detección de objetos en imágenes vector valuadas (tales como RGB o multi-espectrales), como el anterior modelo, éste también detecta bordes de los objetos con o sin gradientes, en el artículo se muestran ejemplos que no pueden ser detectados en cualquier representación escalar. Por ejemplo, objetos con diferentes partes perdidas en diferentes canales son completamente detectados (tales como el problema de oclusión).

Varias mejoras se han dado en los últimos años sobre el tema de segmentación de imágenes a color (RGB), mencionando algunas de ellas, en 2009 Zhai, Wu, Zhang y Sun proponen en [13] un modelo variacional para la segmentación de objetos de interés en imágenes a color, inspirado por el modelo de la geodésica activa[8], el modelo de ajuste de la región escalable[14], el modelo ponderado de variación acotada[15] y los modelos de contornos activos basados en el modelo de Mumford-Shah [16], cuyo funcional de energía incluye una función de discriminación que determina si un pixel de la imagen pertenece al objeto deseado o no. Este nuevo modelo no sólo puede evitar el inconveniente usual en el enfoque de conjuntos de nivel (ej. inicialización y reinicialización), sino también detectar los objetos de interés con precisión. En 2010, Yang, Li, Kao, y Osher proponen en [17] la incorporación del método de segmentación global convexa y del método de partición de Bregman para la minimización de la energía del modelo de ajuste de la región escalable y en 2012, Yuan, Ukwatta, Tai, Fenser, y Schörr proponen en [18] una optimización global más rápida basada en el enfoque de evolución de contornos con forma genérica previa.

1.3. Justificación y alcances

La observación microscópica de muestras de sangre es considerada el estándar de oro para el diagnóstico de muchas enfermedades causadas por parásitos en la sangre, tales como la malaria, la toxoplasmosis, la leishmaniasis y el chagas, las cuales han afectado a cientos de millones de personas en varias partes del mundo. Para llevarla a cabo los frotis de sangre son usualmente observados manualmente por una persona y la calidad de esta valoración depende del nivel de capacitación del observador. Además, el tiempo tomado para observar un frotis normalmente oscila entre 15 y 20 minutos. Es aquí en donde el procesamiento automático de las imágenes de muestras de sangre ayudará a proporcionar una evaluación del paciente de forma rápida y consistente.

En la literatura existen varios algoritmos aplicados para este propósito como [19], [20] y [21], los cuales son basados en conceptos estadísticos, umbralización, corte normalizado y transformación divisoria. Sin embargo no se encontraron aplicaciones de los modelos de contornos activos en la segmentación de los parásitos en muestras de sangre, es por ello que en este trabajo usaremos estos modelos variacionales, los cuales se caracterizan por evolucionar una curva hasta que ésta minimice un funcional de energía y los cuales han demostrado ser una potente herramienta en el área de segmentación de imágenes debido a su robustez y su flexibilidad en la formulación.

En esta tesis se estudiaron diversos modelos de contornos activos, de los cuales se implementaron tres de ellos, el modelo Chan - Vese [10], el modelo de la minimización de la energía de ajuste de la región escalable [14] y la segmentación de distribuciones [22], la implementación se realizó usando el método de conjuntos de nivel, cuya ventaja principal es que la energía no depende de la parametrización de la curva y admite cambios en su topología. Se evaluaron en problemas diversos de segmentación, en particular en nuestro tema de interés, la segmentación de parásitos en imágenes de muestras de sangre, se presentan y discuten los resultados obtenidos, así como posibles soluciones para mejorarlos y se muestran las ventajas de la paralelización de GPUs para este tipo de algoritmos.

1.4. Estructura de la tesis

Capítulo 2

Este capítulo se encuentra dividido en dos secciones, en la primera se encuentran varios conceptos y definiciones de cálculo variacional que usaremos en repetidas ocasiones, tales como la primera variación, derivada variacional, ecuación de Euler-Lagrange y las integrales variacionales, los cuales son muy importantes para la definición de funcionales de energía de los modelos analizados, sin embargo debido a que esta teoría está ligada solamente a integrales variacionales, y el modelo de contornos activos para la segmentación de distribuciones [23], [22] no pueden ser expresados de esta forma, agregamos una subsección en donde extendemos estas integrales variacionales para que los modelos como el anterior puedan ser representados. En la segunda sección se muestra un panorama general sobre los modelos de contornos activos, su formulación en conjuntos de nivel y la discretización que se usó.

Capítulo 3

En este capítulo se revisa el modelo de contornos activos sin bordes [10], se explica su formulación en conjuntos de nivel, el procedimiento para minimizar la energía y la adaptación del método numérico presentado en el capítulo 2, además se presentan varios resultados, algunos en donde se obtienen buenas segmentaciones y otros en donde no.

Capítulo 4

En este capítulo se revisa el modelo de la minimización de la energía de ajuste de la región escalable [14], se explica detalladamente la formulación del modelo en conjuntos de nivel, su minimización y la adaptación de la discretización vista en el capítulo 2, al final se presentan varios resultados en donde se observan las grandes mejoras que se obtienen con este modelo local, comparado con el de Chan - Vese [10], además se muestran varias segmentaciones realizadas con el algoritmo, en imágenes de parásitos en muestras de sangre, en donde se obtuvieron buenos resultados.

Capítulo 5

En este capítulo se revisa un modelo de contornos activos para segmentación de distribuciones [23], [22], se explica su formulación en conjuntos de nivel, el procedimiento para minimizar la energía y la adaptación del método numérico presentado en el capítulo 2, además se presentan varios resultados en imágenes sintéticas y en la aplicación a la segmentación de parásitos en imágenes de muestras de sangre.

Capítulo 6

En este capítulo final, presentamos los resultados de la paralelización del modelo de contornos activos sin bordes, las conclusiones y proponemos posibles trabajos futuros derivados del trabajo de esta tesis.

Capítulo 2

Preliminares Matemáticos

2.1. Cálculo variacional

Los modelos variacionales usados en esta tesis minimizan funcionales. En esta sección, revisamos algunas fórmulas que usaremos repetidas veces en la definición y minimización de estos funcionales, además definiremos los conceptos de primera variación, derivada variacional, ecuación de Euler-Lagrange y las integrales variacionales e introduciremos un tipo de funcionales que nos servirá en la formulación de los términos de ajuste de los modelos, los cuales incluyen a la integral variacional, a este tipo de funcionales los nombraremos integrales variacionales extendidas.

Lema 2.1.1 (Lema Fundamental del Cálculo Variacional) *Sea $f(x)$ una función real valuada y continua en algún conjunto abierto Ω de \mathbb{R}^n , y suponer que*

$$\int_{\Omega} f(x)\eta(x)dx \geq 0 \text{ para todo } \eta \in C_c^{\infty}(\Omega) \text{ con } \eta \geq 0 \quad (2.1)$$

o

$$\int_{\Omega} f(x)\eta(x)dx = 0 \text{ para todo } \eta \in C_c^{\infty}(\Omega) \quad (2.2)$$

se tienen. Entonces se cumple que $f(x) \geq 0$ o $f(x) = 0$, respectivamente, para todo $x \in \Omega$.

Demostración:

Supongamos que 2.1 se satisface, y asumamos que existe un punto $x_0 \in \Omega$ con $f(x_0) < 0$.

2.1 Cálculo variacional

Entonces existe un $\epsilon > 0$ y una bola $B_r(x_0) \subset \Omega$, tal que $f(x) < -\epsilon$ en $B_r(x_0)$. Por medio de la función $\eta \in C_c^\infty(\Omega)$ definida por

$$\eta(x) := \begin{cases} \exp(-1/(r^2 - |x - x_0|^2)) & \text{para } x \in B_r(x_0), \\ 0 & \text{para } x \in \Omega - B_r(x_0). \end{cases}$$

llegamos a la siguiente contradicción

$$0 \leq \int_{\Omega} f(x)\eta(x)dx = \int_{B_r(x_0)} f(x)\eta(x)dx < -\epsilon \int_{B_r(x_0)} \eta(x)dx < 0.$$

Así, la relación implica que $f(x) \geq 0$ para todo $x \in \Omega$. La segunda afirmación es una consecuencia inmediata de la primera.

Lema 2.1.2 *Sea Ω una k -variedad C^1 en \mathbb{R}^n , sea $f(x)$ una función real valuada y continua en Ω , sea \mathcal{H}^k la medida de Hausdorff k dimensional y suponer que*

$$\int_{\Omega} f(x)\eta(x)d\mathcal{H}^k(x) \geq 0 \text{ para todo } \eta \in C_c^\infty(\Omega) \text{ con } \eta \geq 0 \quad (2.3)$$

o

$$\int_{\Omega} f(x)\eta(x)d\mathcal{H}^k(x) = 0 \text{ para todo } \eta \in C_c^\infty(\Omega) \quad (2.4)$$

se tienen. Entonces se cumple que $f(x) \geq 0$ o $f(x) = 0$, respectivamente, para todo $x \in \Omega$.

Demostración:

Supongamos que 2.3 se satisface, y asumamos que existe un punto $x_0 \in \Omega$ con $f(x_0) < 0$. Entonces existe un $\epsilon > 0$ y una bola $B_r(x_0) \subset \mathbb{R}^n$, tal que $f(x) < -\epsilon$ en $B_r(x_0) \cap \Omega$. Por medio de la función $\eta \in C_c^\infty(\Omega)$ definida por

$$\eta(x) := \begin{cases} \exp(-1/(r^2 - |x - x_0|^2)) & \text{para } x \in \Omega \cap B_r(x_0), \\ 0 & \text{para } x \in \Omega - \Omega \cap B_r(x_0). \end{cases}$$

llegamos a la siguiente contradicción

2.1 Cálculo variacional

$$0 \leq \int_{\Omega} f(x)\eta(x)d\mathcal{H}^k(x) = \int_{\Omega \cap B_r(x_0)} f(x)\eta(x)d\mathcal{H}^k(x) < -\epsilon \int_{\Omega \cap B_r(x_0)} \eta(x)d\mathcal{H}^k(x) < 0.$$

Así, la relación implica que $f(x) \geq 0$ para todo $x \in \Omega$. La segunda afirmación es una consecuencia inmediata de la primera.

Definición 2.1.3 (Primera Variación) *Sea V un subconjunto de un espacio vectorial X sobre \mathbb{R} , $\mathcal{F} : V \rightarrow \mathbb{R}$ un funcional, $u_0 \in V$, $\zeta \in X$, supongamos que el segmento $\{u : u = u_0 + \varepsilon\zeta, |\varepsilon| < \varepsilon_0\}$ está contenido en V para algún $\varepsilon_0 > 0$. Entonces definimos la función $\phi : (-\varepsilon_0, \varepsilon_0) \rightarrow \mathbb{R}$ por $\phi(\varepsilon) := \mathcal{F}(u_0 + \varepsilon\zeta)$ y, si $\phi'(0)$ existe, lo llamamos la primera variación de \mathcal{F} en u_0 en dirección de ζ y escribimos:*

$$\delta\mathcal{F}(u_0, \zeta) = \phi'(0)$$

Consideremos funcionales \mathcal{F} del tipo:

$$\mathcal{F} = \int_{\Omega} F(x, u(x), Du(x))dx,$$

Llamadas *integrales variacionales*. El integrando $F(x, u, p)$ de una integral \mathcal{F} será denotado como *Lagrangiano*, o *integrando variacional*, o *función de Lagrange*.

Aquí Ω es un dominio abierto de \mathbb{R}^n , $n \geq 1$, y $u(x)$ denota una función definida en Ω la cual en general es un mapeo vector valuado de Ω a \mathbb{R}^N , $N \geq 1$, eso es, $u(x) = (u^1(x), \dots, u^N(x))$, y Du denota la primera derivada de u ,

$$Du = (D_{\alpha}u^i), \quad i = 1, \dots, N, \quad \alpha = 1, \dots, n.$$

Las notaciones que usaremos son $x = (x^{\alpha})$, $u = (u^i)$, $p = (p_{\alpha}^i)$, $p = (p_1, \dots, p_n)$, $p_{\alpha} = (p_{\alpha}^1, \dots, p_{\alpha}^N)$, $F_{p_{\alpha}} = (F_{p_{\alpha}^1}, \dots, F_{p_{\alpha}^N})$ y usaremos la convención de Einstein, se tienen sumas sobre dobles apariciones de índices griegos de 1 a n , y sobre repeticiones de índices latinos de 1 a N .

Teorema 2.1.4 (Ecuación de Euler) *Suponer que $F \in C^2(\mathcal{U})$, (con \mathcal{U} abierto de $\mathbb{R}^n \times \mathbb{R}^N \times \mathbb{R}^{nN}$, conteniendo el conjunto $\{(x, u(x), Du(x)) : x \in \bar{\Omega}\}$) y que u es un extremo débil*

2.1 Cálculo variacional

de \mathcal{F} , esto es,

$$\delta\mathcal{F}(u, \psi) = \int_{\Omega} \{F_u(x, u, Du) \cdot \psi + F_p(x, u, Du) \cdot D\psi\} dx = 0 \quad \forall \psi \in C_c^\infty(\Omega, \mathbb{R}^N).$$

Además sea u de clase $C^2(\Omega, \mathbb{R}^N)$. Entonces $u(x)$ satisface la ecuación de Euler

$$D_\alpha F_{p_\alpha^i}(x, u(x), Du(x)) - F_{u^i}(x, u(x), Du(x)) = 0, \quad 1 \leq i \leq N, \quad (2.5)$$

para todo $x \in \Omega$.

Demostración:

Por la regla del producto tenemos:

$$F_{p_\alpha^i}(x, u, Du) D_\alpha \psi^i = -D_\alpha F_{p_\alpha^i}(x, u, Du) \psi^i + D_\alpha (F_{p_\alpha^i}(x, u, Du) \psi^i)$$

pero $D_\alpha (F_{p_\alpha^i}(x, u, Du) \psi^i) = \nabla \cdot (F_{p_\alpha^i}(x, u, Du) \psi^i)$, por lo que usando el teorema de la divergencia en el plano, tenemos:

$$\int_{\Omega} \nabla \cdot (F_{p_\alpha^i}(x, u, Du) \psi^i) dx = \int_{\partial\Omega} (F_{p_\alpha^i}(x, u, Du) \psi^i) \cdot \vec{n} d\mathcal{H}^{n-1}(x)$$

Así nos queda:

$$\begin{aligned} \delta\mathcal{F}(u, \psi) &= \int_{\Omega} \{F_{u^i}(x, u(x), Du(x)) - D_\alpha F_{p_\alpha^i}(x, u(x), Du(x))\} \psi^i(x) dx \\ &\quad + \int_{\partial\Omega} (F_{p_\alpha^i}(x, u, Du) \psi^i) \cdot \vec{n} d\mathcal{H}^{n-1}(x) \end{aligned} \quad (2.6)$$

Y como $\psi \in C_c^\infty(\Omega, \mathbb{R}^N)$, entonces

$$\int_{\Omega} \{F_{u^i}(x, u(x), Du(x)) - D_\alpha F_{p_\alpha^i}(x, u(x), Du(x))\} \psi^i(x) dx = 0$$

se tiene para cualquier función de prueba $\psi(x) = (\psi^1(x), \dots, \psi^N(x))$ de clase $C_c^\infty(\Omega, \mathbb{R}^N)$. Fijando algún índice i y eligiendo $\psi^j(x) = 0$ para todo $j \neq i$ y $\psi^i = \eta$ para algún $\eta \in C_c^\infty$ arbitrario, llegamos a

2.1 Cálculo variacional

$$\int_{\Omega} \{F_{u^i}(x, u(x), Du(x)) - D_{\alpha}F_{p_{\alpha}^i}(x, u(x), Du(x))\} \eta(x) dx = 0$$

y usando el lema fundamental del calculo variacional se sigue que:

$$F_{u^i}(x, u(x), Du(x)) - D_{\alpha}F_{p_{\alpha}^i}(x, u(x), Du(x)) = 0.$$

Definición 2.1.5 (Derivada variacional) Sea \mathcal{F} una integral variacional con lagrangiano F , llamamos a

$$\frac{\delta \mathcal{F}}{\delta u} = F_u(\cdot, u(\cdot), Du(\cdot)) - D_{\alpha}F_{p_{\alpha}}(\cdot, u(\cdot), Du(\cdot)). \quad (2.7)$$

derivada variacional del funcional \mathcal{F} en u .

Proposición 2.1.6 (Condiciones Naturales de Frontera) Sea $\partial\Omega$ una variedad de clase C^1 , $F \in C^2(\mathcal{U})$, $u \in C^2(\bar{\Omega}, \mathbb{R}^N)$, y si $\delta F(u, \psi) = 0$ para todo $\psi \in C^1(\bar{\Omega}, \mathbb{R}^N)$, entonces u es un extremo de \mathcal{F} el cual, en $\partial\Omega$, satisface las siguientes “condiciones naturales de frontera”

$$n_{\alpha}F_{p_{\alpha}^i}(x, u, Du) = 0, \quad 1 \leq i \leq N$$

Aquí $n(x) = (n_1(x), \dots, n_n(x))$ denota la normal exterior de $\partial\Omega$ en el punto $x \in \partial\Omega$.

Demostración:

Por la formula 2.6

$$\delta \mathcal{F}(u, \psi) = \int_{\Omega} \frac{\delta \mathcal{F}}{\delta u} \cdot \psi dx + \int_{\partial\Omega} n_{\alpha}(x) F_{p_{\alpha}^i}(x, u(x), Du(x)) \psi^i(x) d\mathcal{H}^{n-1}(x).$$

donde $d\mathcal{H}^{n-1}$ representa el elemento de área $(n-1)$ -dimensional de $\partial\Omega$. Por el teorema 2.1.4 $\frac{\delta \mathcal{F}}{\delta u}(u) = 0$ en Ω , de donde

$$\delta \mathcal{F}(u, \psi) = \int_{\partial\Omega} n_{\alpha}(x) F_{p_{\alpha}^i}(x, u(x), Du(x)) \psi^i(x) d\mathcal{H}^{n-1}(x).$$

para todo $\psi \in C^1(\bar{\Omega}, \mathbb{R}^N)$. Por el lema 2.1.2 concluimos que

$$n_{\alpha}(x) F_{p_{\alpha}^i}(x, u(x), Du(x)) = 0, \quad \forall x \in \partial\Omega, \quad 1 \leq i \leq N,$$

2.1 Cálculo variacional

Lo cual en forma vectorial podemos escribirlo como:

$$n(x)F_p(x, u(x), Du(x)) = 0, \text{ en } \partial\Omega$$

2.1.1. Integrales variacionales extendidas

Definición 2.1.7 Sea Ω un conjunto abierto de \mathbb{R}^n tal que $\partial\Omega$ es una variedad de clase C^1 y $u \in C^2(\bar{\Omega}, \mathbb{R}^N)$, diremos que un funcional $\mathcal{F}(u)$ es una integral variacional extendida, si tiene primera variación $\delta\mathcal{F}(u, \psi)$ y se cumple que

$$\delta\mathcal{F}(u, \psi) = \int_{\Omega} f(x, u(x), Du(x)) \cdot \psi dx + \int_{\partial\Omega} g(x, u(x), Du(x)) \cdot \psi d\mathcal{H}^{n-1}(x) \quad \forall \psi \in C^1(\bar{\Omega}, \mathbb{R}^N)$$

para algunos $f, g \in C^1(\mathcal{U})$. A tal función f la llamaremos derivada variacional de \mathcal{F} y la denotaremos como:

$$\frac{\delta\mathcal{F}}{\delta u} = f(u)$$

y a la función g la llamaremos función en la frontera y la denotaremos por:

$$\partial\mathcal{F}(u) = g(u)$$

Notemos que las integrales variacionales están incluidas en las integrales variacionales extendidas, además la derivada variacional definida para una integral variacional, coincide con la derivada variacional definida para una integral variacional.

Proposición 2.1.8 Sea \mathcal{F} una integral variacional extendida, con extremo u , es decir $\delta\mathcal{F}(u, \psi) = 0$, para todo $\psi \in C^1(\bar{\Omega}, \mathbb{R}^N)$, entonces la derivada variacional y su función de frontera se anulan en u , es decir

$$\frac{\delta\mathcal{F}}{\delta u} = 0 \quad \forall x \in \Omega \quad y$$

$$\partial\mathcal{F}(u) = 0 \quad \forall x \in \partial\Omega$$

2.1 Cálculo variacional

Demostración:

Como \mathcal{F} es una integral variacional extendida, entonces:

$$\delta \mathcal{F}(u, \psi) = \int_{\Omega} f(x, u(x), Du(x)) \cdot \psi dx + \int_{\partial\Omega} g(x, u(x), Du(x)) \cdot \psi d\mathcal{H}^{n-1}(x) \quad \forall \psi \in C^1(\bar{\Omega}, \mathbb{R}^N) \quad (2.8)$$

para algunos $f, g \in C^1(\mathcal{U})$. Eligiendo $\psi \in C_c^\infty(\Omega, \mathbb{R}^N)$, obtenemos que

$$0 = \delta \mathcal{F}(u, \psi) = \int_{\Omega} f(x, u(x), Du(x)) \cdot \psi dx$$

usando el lema fundamental del cálculo variacional tenemos que

$$\frac{\delta \mathcal{F}}{\delta u} = f(x, u(x), Du(x)) = 0 \quad \forall x \in \Omega \quad (2.9)$$

sustituyendo en 2.8, nos queda que

$$0 = \delta \mathcal{F}(u, \psi) = \int_{\partial\Omega} g(x, u(x), Du(x)) \cdot \psi d\mathcal{H}^{n-1}(x)$$

usando el lema 2.1.2, concluimos que las condiciones naturales de frontera nos queda:

$$\partial \mathcal{G}(u) = g(x, u(x), Du(x)) = 0, \quad \forall x \in \partial\Omega$$

Proposición 2.1.9 *Sea \mathcal{G} un funcional definido por*

$$\mathcal{G}(u) = g(\mathcal{F}^1(u), \dots, \mathcal{F}^m(u))$$

donde $g : \mathbb{R}^m \rightarrow \mathbb{R}$ es una función C^1 y \mathcal{F}^i son integrales variacionales extendidas, entonces \mathcal{G} también es una integral variacional extendida, con derivada variacional:

$$\frac{\delta \mathcal{G}}{\delta u} = \frac{\partial g}{\partial z^i} \frac{\delta \mathcal{F}^i}{\delta u}$$

y con función de frontera:

$$\partial \mathcal{G}(u) = \frac{\partial g}{\partial z^i} \partial \mathcal{F}^i(u), \quad \forall x \in \partial\Omega$$

2.1 Cálculo variacional

Demostración:

$$\begin{aligned}
 \delta \mathcal{G}(u, \psi) &= \left. \frac{\partial}{\partial \varepsilon} g(\mathcal{F}_1(u + \varepsilon\psi), \dots, \mathcal{F}_m(u + \varepsilon\psi)) \right|_{\varepsilon=0} \\
 &= \left[\frac{\partial g}{\partial x^i} \frac{\partial \mathcal{F}_i(u + \varepsilon\psi)}{\partial \varepsilon} \right] \Big|_{\varepsilon=0} \\
 &= \frac{\partial g}{\partial x^i}(\mathcal{F}_1(u), \dots, \mathcal{F}_m(u)) \delta \mathcal{F}_i(u, \psi) \\
 &= \frac{\partial g}{\partial x^i}(\mathcal{F}_1(u), \dots, \mathcal{F}_m(u)) \left[\int_{\Omega} \frac{\delta \mathcal{F}_i}{\delta u} \cdot \psi + \int_{\partial\Omega} \partial \mathcal{F}_i(u) \cdot \psi \mathcal{H}^{n-1}(x) \right] \\
 &= \int_{\Omega} \left[\frac{\partial g}{\partial x^i} \frac{\delta \mathcal{F}_i}{\delta u} \right] \cdot \psi + \int_{\partial\Omega} \left[\frac{\partial g}{\partial x_i} \partial \mathcal{F}_i \right] \cdot \psi d\mathcal{H}^{n-1}(x)
 \end{aligned}$$

de donde concluimos lo pedido.

Proposición 2.1.10 *Sea \mathcal{G} un funcional definido por:*

$$\mathcal{G}(u) = \int_{\mathcal{L}} \mathcal{F}_z dz$$

donde \mathcal{F}_z son integrales variacionales extendidas, entonces \mathcal{G} es una integral variacional extendida, en donde su derivada variacional está dada por:

$$\frac{\delta \mathcal{G}}{\delta u} = \int_{\mathcal{L}} \frac{\delta \mathcal{F}_z}{\delta u} dz$$

y su función de frontera está dado por:

$$\partial \mathcal{G}(u) = \int_{\mathcal{L}} \partial \mathcal{F}_z(u) dz$$

Demostración:

2.1 Cálculo variacional

$$\begin{aligned}\delta\mathcal{L}(u, \psi) &= \frac{\partial}{\partial\varepsilon} \int_{\mathcal{L}} \mathcal{F}_z dz \Big|_{\varepsilon=0} \\ &= \int_{\mathcal{L}} \frac{\partial}{\partial\varepsilon} \mathcal{F}_z dz \Big|_{\varepsilon=0} \\ &= \int_{\mathcal{L}} \delta\mathcal{F}_z(u, \psi) dz \\ &= \int_{\mathcal{L}} \left[\int_{\Omega} \frac{\delta\mathcal{F}_z}{\delta u} \cdot \psi + \int_{\partial\Omega} \partial\mathcal{F}_z(u) \cdot \psi d\mathcal{H}^{n-1}(x) \right] dz \\ &= \int_{\Omega} \left[\int_{\mathcal{L}} \frac{\delta\mathcal{F}_z}{\delta u} dz \right] \cdot \psi dx + \int_{\partial\Omega} \left[\int_{\mathcal{L}} \partial\mathcal{F}_z(u) dz \right] \cdot \psi d\mathcal{H}^{n-1}(x)\end{aligned}$$

de donde concluimos lo pedido.

2.2. Modelos de contornos activos

La idea básica de los modelos de contornos activos es evolucionar una curva hasta que coincida con el borde del objeto que queremos detectar, el objeto que nuestro modelo segmentará dependerá de las características que le hayamos pedido minimizar en un funcional de energía de la curva. Supongamos que nuestro funcional de energía es una integral variacional extendida $\mathcal{E}(\Gamma)$, donde Γ es la curva a evolucionar (el funcional podría depender también de otras variables o funciones), este funcional de energía lo llamaremos *término de ajuste*.

Así la curva que contenga nuestro objeto a segmentar, será la que resuelva el problema:

$$\inf \mathcal{E}(\Gamma)$$

Para resolver numéricamente los modelos de contornos activos, como el Chan-Vese [10] y el de ajuste de la región escalable [14], es necesario agregar términos que regularicen el problema, por ejemplo, que suavicen el contorno que está evolucionando, uno de los términos comúnmente usados es el *término de longitud de arco*, en donde lo que se espera es obtener un contorno de longitud mínima. Siendo $\mathcal{L}(\Gamma)$ la longitud de la curva Γ , tendremos que nuestro nuevo funcional quedaría:

$$\mathcal{F}(\Gamma) = \mathcal{E}(\Gamma) + \mu\mathcal{L}(\Gamma)$$

2.2.1. Formulación en conjuntos de nivel

Uno de los métodos más eficaces para resolver este tipo de problemas, es el método de conjuntos de nivel, en el que se define una función $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ que sea Lipschitz ¹, de tal forma que Γ está representada como el conjunto de nivel 0 de ϕ , tal que:

$$\begin{cases} \Gamma & = \{(x, y) \in \mathbb{R}^2 : \phi(x, y) = 0\}, \\ dentro(\Gamma) & = \{(x, y) \in \mathbb{R}^2 : \phi(x, y) > 0\}, \\ fuera(\Gamma) & = \{(x, y) \in \mathbb{R}^2 : \phi(x, y) < 0\}, \end{cases}$$

así la curva desconocida de menor dimensión Γ se sustituye por otra variable desconocida de

¹Una función $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ se dice que es Lipschitz si existe una constante K tal que $\|f(x) - f(y)\| \leq K\|x - y\| \forall x, y \in \mathbb{R}^n$

2.2 Modelos de contornos activos

mayor dimensión ϕ .

Expresando el término de longitud de arco \mathcal{L} en términos de ϕ :

$$\mathcal{L}(\phi) = \int_{\Omega} |\nabla H(\phi)| dx dy = \int_{\Omega} \delta(\phi) |\nabla \phi| dx dy$$

Donde H es la función de Heaviside y δ es la medida de Dirac de 1 dimensión concentrada en 0 dadas por:

$$H(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

y $\delta(x) = \frac{d}{dx}H(x)$. Aquí $\mathcal{L}(\phi)$ sería la longitud de la curva de nivel cero de ϕ .

Así la formulación del funcional \mathcal{F} en conjuntos de nivel quedaría:

$$\mathcal{F}(\phi) = \mathcal{E}(\phi) + \mu \int_{\Omega} \delta(\phi) |\nabla \phi| dx dy$$

2.2.2. Minimización

Ya que la función de Heaviside H no es diferenciable en 0, necesitamos regularizarla numéricamente, esto es encontrar funciones diferenciables que cumplan que para los números no negativos la función esté cerca de uno y para los números negativos la función esté cerca de cero. Denotemos las versiones regularizadas de H y δ por H_{ϵ} y δ_{ϵ} respectivamente y definamos como en [10]:

$$H_{\epsilon} = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan\left(\frac{x}{\epsilon}\right) \right) \quad (2.10)$$

$$\delta_{\epsilon}(x) = H'_{\epsilon}(x) = \frac{1}{\pi} \left(\frac{\epsilon}{\epsilon^2 + x^2} \right) \quad (2.11)$$

Regularizando con esto al funcional \mathcal{F} obtenemos:

$$\mathcal{F}_{\epsilon}(\phi) = \mathcal{E}_{\epsilon}(\phi) + \mu \int_{\Omega} \delta_{\epsilon}(\phi) |\nabla \phi| dx dy$$

Usando la proposición 2.8 encontramos que la ecuación de Euler es:

2.2 Modelos de contornos activos

$$\frac{\delta \mathcal{F}_\epsilon}{\delta \phi} = \frac{\delta \mathcal{E}_\epsilon}{\delta \phi} - \mu \delta_\epsilon(\phi) \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) = 0 \quad (2.12)$$

y que se tienen las siguientes condiciones naturales de frontera

$$\partial \mathcal{F}(\phi) + \left(\delta_\epsilon(\phi) \frac{\nabla \phi}{|\nabla \phi|} \right) \cdot \vec{n} = 0, \text{ en } \partial \Omega \quad (2.13)$$

Para resolver numéricamente consideraremos el siguiente problema de evolución, en donde la dirección del descenso es parametrizada por una variable artificial del tiempo t .

$$\begin{cases} \frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \frac{\delta \mathcal{E}_\epsilon}{\delta \phi} & \text{en } \Omega, \\ \partial \mathcal{F}(\phi) + \frac{\delta_\epsilon(\phi)}{|\nabla \phi|} \frac{\partial \phi}{\partial n} = 0 & \text{en } \partial \Omega. \end{cases} \quad (2.14)$$

2.2.3. Método numérico

Supongamos que la medida de la imagen es $m \times n$. Usaremos el método de diferencias finitas para la discretización. Sea $x, y \in \Omega$ el espacio de la malla con h_x, h_y iguales a $\frac{1}{n}, \frac{1}{m}$. Sea $(x_i, y_j) = (ih_x, jh_y)$ para $i = 1, 2, \dots, n$ y $j = 1, \dots, m$. También tomemos $\phi_{i,j}^k = \phi(k\Delta t, x_i, y_j)$ como aproximación de $\phi(t, x, y)$, donde $k \geq 0$ y $\phi^0 = \phi_0$ serían dados (aproximaciones iniciales). Las diferencias son denotadas por:

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

Usando el método semi-implícito (SI) encontraremos ϕ^k . Esta discretización da la siguiente ecuación en ϕ :

$$\begin{aligned} \frac{\phi_{ij}^{k+1} - \phi_{ij}^k}{\Delta t} &= \delta_\epsilon(\phi_{ij}^k) \left[\frac{\mu}{h_x^2} \Delta_-^x \left(\frac{\Delta_+^x \phi_{ij}^{k+1}}{\sqrt{(\Delta_+^x \phi_{ij}^k)^2/h_x^2 + (\Delta_c^y \phi_{i,j}^k)^2/h_y^2}} \right) \right. \\ &+ \left. \frac{\mu}{h_y^2} \Delta_-^y \left(\frac{\Delta_+^y \phi_{ij}^{k+1}}{\sqrt{(\Delta_c^x \phi_{i,j}^k)^2/h_x^2 + (\Delta_+^y \phi_{ij}^k)^2/h_y^2}} \right) \right] \\ &- \frac{\delta \mathcal{E}_\epsilon}{\delta \phi}(\phi_{i,j}^k). \end{aligned}$$

entonces

2.2 Modelos de contornos activos

$$\begin{aligned}
\frac{\phi_{ij}^{k+1} - \phi_{ij}^k}{\Delta t} &= \delta_\epsilon(\phi_{ij}^k) \left[\frac{\mu}{h_x^2} \left(\frac{\Delta_x \phi_{ij}^{k+1}}{\sqrt{(\Delta_x \phi_{ij}^k/h_x)^2 + ((\Delta_c^y \phi_{i,j}^k)/h_y)^2}} \right. \right. \\
&\quad \left. \left. - \frac{\Delta_x \phi_{i-1,j}^{k+1}}{\sqrt{(\Delta_x \phi_{i-1,j}^k/h_x)^2 + ((\Delta_c^y \phi_{i-1,j}^k)/h_y)^2}} \right) \right. \\
&\quad \left. + \frac{\mu}{h_y^2} \left(\frac{\Delta_y \phi_{ij}^{k+1}}{\sqrt{(\Delta_c^x \phi_{i,j}^k/h_x)^2 + (\Delta_y \phi_{ij}^k/h_y)^2}} \right. \right. \\
&\quad \left. \left. - \frac{\Delta_y \phi_{i,j-1}^{k+1}}{\sqrt{(\Delta_c^x \phi_{i,j-1}^k/h_x)^2 + (\Delta_y \phi_{i,j-1}^k/h_y)^2}} \right) \right] \\
&\quad - \frac{\delta E_\epsilon}{\delta \phi}(\phi_{i,j}^k).
\end{aligned}$$

Despejando, obtenemos:

$$\begin{aligned}
\phi_{ij}^{k+1} &= \phi_{ij}^k + \Delta t \delta_\epsilon(\phi_{ij}^k) \left[\frac{\mu}{h_x^2} \left(\frac{\Delta_x \phi_{ij}^{k+1}}{\sqrt{(\Delta_x \phi_{ij}^k/h_x)^2 + ((\Delta_c^y \phi_{i,j}^k)/h_y)^2}} \right. \right. \\
&\quad \left. \left. - \frac{\Delta_x \phi_{i-1,j}^{k+1}}{\sqrt{(\Delta_x \phi_{i-1,j}^k/h_x)^2 + ((\Delta_c^y \phi_{i-1,j}^k)/h_y)^2}} \right) \right. \\
&\quad \left. + \frac{\mu}{h_y^2} \left(\frac{\Delta_y \phi_{ij}^{k+1}}{\sqrt{(\Delta_c^x \phi_{i,j}^k/h_x)^2 + (\Delta_y \phi_{ij}^k/h_y)^2}} \right. \right. \\
&\quad \left. \left. - \frac{\Delta_y \phi_{i,j-1}^{k+1}}{\sqrt{(\Delta_c^x \phi_{i,j-1}^k/h_x)^2 + (\Delta_y \phi_{i,j-1}^k/h_y)^2}} \right) \right] \\
&\quad - \Delta t \frac{\delta E_\epsilon}{\delta \phi}(\phi_{i,j}^k).
\end{aligned}$$

Denotemos los coeficientes de $\phi_{i+1,j}^{k+1}$, $\phi_{i-1,j}^{k+1}$, $\phi_{i,j+1}^{k+1}$, $\phi_{i,j-1}^{k+1}$ por A_1, A_2, A_3, A_4 respectivamente, así obtenemos el siguiente sistema de ecuaciones lineales:

$$\begin{aligned}
&\phi_{ij}^{k+1} [1 + \Delta t \mu \delta_\epsilon(\phi_{ij}^k)(A_1/h_x^2 + A_2/h_x^2 + A_3/h_y^2 + A_4/h_y^2)] \\
&= \phi_{ij}^k + \Delta t \mu \delta_\epsilon(\phi_{ij}^k) [(A_1/h_x^2 \phi_{i+1,j}^{k+1} + A_2/h_x^2 \phi_{i-1,j}^{k+1} + A_3/h_y^2 \phi_{i,j+1}^{k+1} + A_4/h_y^2 \phi_{i,j-1}^{k+1})] \quad (2.15) \\
&- \Delta t \frac{\delta E_\epsilon}{\delta \phi}(\phi_{i,j}^k).
\end{aligned}$$

Este sistema de ecuaciones lineales es el que resolveremos mediante algún método iterativo apropiado. Este sistema es diagonal dominante si

$$1 + \Delta t \mu \delta_\epsilon(\phi_{ij}^k)(A_1/h_x^2 + A_2/h_x^2 + A_3/h_y^2 + A_4/h_y^2) > \Delta t \mu \delta_\epsilon(\phi_{ij}^k)(A_1/h_x^2 + A_2/h_x^2 + A_3/h_y^2 + A_4/h_y^2).$$

Lo cual es cierto ya que $1 > 0$, por lo tanto como este sistema es estrictamente diagonal dominante, entonces la sucesión generada por el método de Jacobi converge a la solución del sistema [24].

Capítulo 3

Modelo de Contornos Activos sin bordes

T.F. Chan y L.A. Vese en [10] proponen un modelo variacional para la segmentación de imágenes, el cual no usa el gradiente de la imagen u como el término de paro (éste término de paro mencionado es una función g que depende del gradiente de la imagen ∇u que se usa para indicar cuando la curva está sobre el contorno buscado [10]) pero éste depende de la técnica de segmentación de Mumford-Shah [11]. Esto significa que este modelo puede detectar contornos con o sin gradientes. La idea básica de los contornos o serpientes es evolucionar una curva con el fin de detectar objetos en esa imagen.

Sea u la imagen dada, como una función acotada. Asumimos que u está formada por dos regiones de aproximadamente intensidades constantes de distintos valores u_d y u_f . También asumimos que el objeto a detectar está representado por la región con intensidad u_d y su contorno es Γ_f . Consideremos el término de ajuste:

$$\mathcal{E}(\Gamma) = \mathcal{F}_1(\Gamma) + \mathcal{F}_2(\Gamma) = \int_{dentro(\Gamma)} |u - c_1|^2 dx dy + \int_{fuera(\Gamma)} |u - c_2|^2 dx dy \quad (3.1)$$

donde Γ es la curva envolvente desconocida y c_1 y c_2 son los promedios de la imagen u dentro y fuera de Γ , respectivamente. En este caso es claro que Γ_f , el contorno del objeto, es el minimizador de la energía. Para notar esto, observemos la figura 4.1, la cual es una imagen en condiciones idóneas, es decir que está dividida en dos regiones de distintas intensidades constantes.

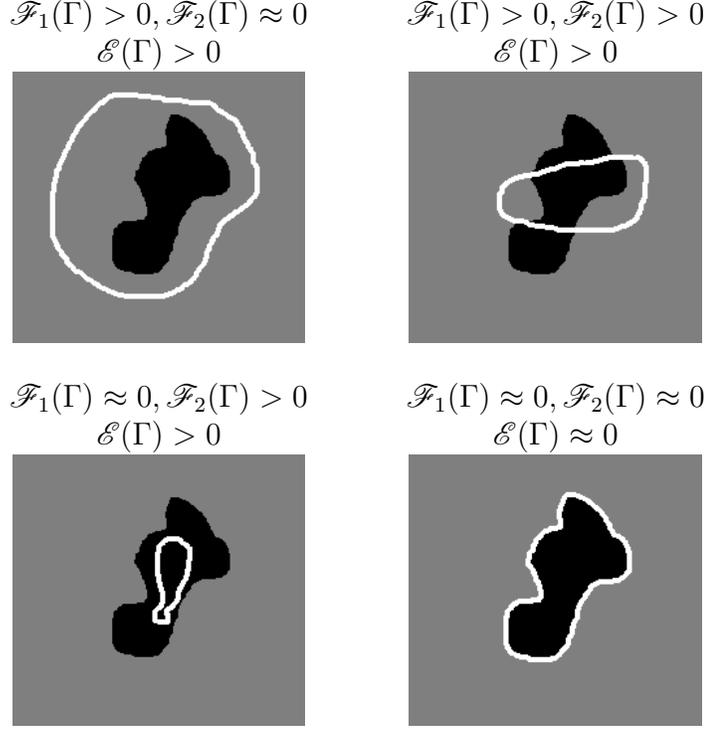


Figura 3.1: El mínimo se obtiene cuando la curva coincide con el contorno del objeto

En condiciones idóneas, es fácil notar que el mínimo se obtiene cuando la curva envuelve por completo al objeto en su contorno, pero normalmente las imágenes analizadas para realizar la segmentación no están divididas solo en dos regiones con intensidades constantes, sino que hay varios tonos de gris, razón por la cual la curva podría no quedar suave o incluso separarse en muchos puntos, es por eso que se necesitan usar términos de regularización, tales como la curva de longitud mínima o la curva que encierra el área mínima. Los términos de regularización usados en [10] son la longitud de Γ y el área de la región dentro de Γ . En esta tesis trabajaremos con el término de longitud de arco como se presentó en el capítulo anterior, así consideraremos ahora la energía siguiente:

$$\begin{aligned} \mathcal{F}(\Gamma, c_1, c_2) = & \mu(\mathcal{L}(\Gamma))^p + \lambda_1 \int_{dentro(\Gamma)} |u - c_1|^2 dx dy \\ & + \lambda_2 \int_{fuera(\Gamma)} |u - c_2|^2 dx dy \end{aligned}$$

donde c_1 y c_2 son constantes desconocidas, $\mu \geq 0, \lambda_1, \lambda_2 > 0$ son parámetros fijos. En [10]

3.1 Formulación en conjuntos de nivel

$\lambda_1 = \lambda_2 = 1$ y $p > 0$, en particular nosotros tomaremos $p = 1$. Así, nos queda el siguiente problema de minimización.

$$\inf_{\Gamma, c_1, c_2} \mathcal{F}(\Gamma, c_1, c_2) = \mu \mathcal{L}(\Gamma) + \mathcal{E}_{cv}(\Gamma, c_1, c_2)$$

En donde \mathcal{E}_{cv} es el término de ajuste dado por:

$$\mathcal{E}_{cv}(\Gamma, c_1, c_2) = \lambda_1 \int_{dentro(\Gamma)} |u - c_1|^2 dx dy + \lambda_2 \int_{fuera(\Gamma)} |u - c_2|^2 dx dy \quad (3.2)$$

3.1. Formulación en conjuntos de nivel

Sea $\phi : R^2 \rightarrow R$ una función Lipschitz, de tal forma que Γ está representada como el conjunto de nivel 0 de ϕ , como en el capítulo anterior.

Expresando el término de ajuste, en términos de ϕ :

$$\begin{aligned} \int_{\phi \geq 0} |u - c_1|^2 dx dy &= \int_{\Omega} |u - c_1|^2 H(\phi) dx dy \\ \int_{\phi < 0} |u - c_2|^2 dx dy &= \int_{\Omega} |u - c_2|^2 (1 - H(\phi)) dx dy \end{aligned}$$

Así la formulación del término de ajuste para el conjunto de nivel queda:

$$\mathcal{E}^{cv}(\phi, c_1, c_2) = \lambda_1 \int_{\Omega} |u - c_1|^2 H(\phi) dx dy + \lambda_2 \int_{\Omega} |u - c_2|^2 (1 - H(\phi)) dx dy$$

una vez obtenido ϕ , la segmentación de la imagen está dada por:

$$w = c_1 H(\phi) + c_2 (1 - H(\phi)).$$

La existencia y unicidad de la minimización son discutidas con detalle en el artículo de Chan-Vese [10].

3.2. Minimización

Para poder minimizar el funcional tomemos las versiones suaves de la función de Heaviside y la delta de Dirac del capítulo anterior, así obtenemos:

3.2 Minimización

$$\mathcal{E}_\epsilon^{cv}(\phi, c_1, c_2) = \lambda_1 |u - c_1|^2 H_\epsilon(\phi) dx dy + \lambda_2 |u - c_2|^2 (1 - H_\epsilon(\phi))$$

Para minimizar el funcional separamos las variables, primero tomemos a la función ϕ fija y minimicemos con respecto a c_1 y c_2 .

Minimizando con respecto a c_1 , derivemos con respecto a él:

$$\frac{\partial \mathcal{E}_\epsilon^{cv}}{\partial c_1} = - \int_{\Omega} 2(u - c_1) H(\phi) dx dy$$

igualando a cero y despejando concluimos que:

$$c_1(\phi) = \frac{\int_{\Omega} u H(\phi) dx dy}{\int_{\Omega} H(\phi) dx dy}$$

para comprobar que es mínimo obtengamos la segunda derivada:

$$\frac{\partial^2 \mathcal{E}_\epsilon^{cv}}{\partial c_1^2} = 2 \int_{\Omega} H(\phi) dx dy > 0$$

ya que este término es el área dentro de la curva.

Análogamente obtenemos el mínimo con respecto a c_2 en términos de la función ϕ ,

$$c_2(\phi) = \frac{\int_{\Omega} u(1 - H(\phi)) dx dy}{\int_{\Omega} (1 - H(\phi)) dx dy} \quad (3.3)$$

Ahora suponiendo c_1 y c_2 fijas, obtenemos la derivada variacional de $\mathcal{E}_\epsilon^{cv}$ con respecto a ϕ .

$$\frac{\delta E_\epsilon}{\delta \phi} = \delta_\epsilon(\phi) (\lambda_1 |u - c_1|^2 - \lambda_2 |u - c_2|^2)$$

y su función en la frontera $\partial \mathcal{E}_\epsilon^{cv}(\phi, c_1, c_2) = 0$.

Usando la fórmula 2.14, obtenemos el siguiente problema de evolución:

$$\begin{cases} \frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - v - \lambda_1 (u - c_1)^2 + \lambda_2 (u - c_2)^2 \right] & \text{en } \Omega, \\ \frac{\partial \phi}{\partial n} = 0 & \text{en } \partial \Omega. \end{cases}$$

En la siguiente sección derivaremos la aproximación numérica del modelo y discutiremos la

3.3 Método numérico

existencia de métodos usados para resolver la evolución del problema.

3.3. Método numérico

Discreticemos el problema de evolución que tenemos,

$$\begin{cases} c_1(\phi) = \frac{\int_{\Omega} uH(\phi)dxdy}{\int_{\Omega} H(\phi)dxdy}, & c_2(\phi) = \frac{\int_{\Omega} u(1-H(\phi))dxdy}{\int_{\Omega} (1-H(\phi))dxdy}, \\ \frac{\partial \phi}{\partial t} = \delta_{\epsilon}(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - v - \lambda_1(u - c_1)^2 + \lambda_2(u - c_2)^2 \right] & \text{en } \Omega, \\ \frac{\partial \phi}{\partial n} = 0 & \text{en } \partial\Omega. \end{cases}$$

Dado que el sistema tiene dos variables además de la ϕ , primero calcularemos los valores de c_1 y c_2 , y luego sustituiremos estos valores en la ecuación de evolución, una vez encontrada la nueva ϕ , actualizamos los valores de $c_1(\phi)$ y $c_2(\phi)$ y continuamos así sucesivamente.

$$c_1(\phi^k) = \frac{\sum_{ij} u_{ij} H_{\epsilon}(\phi_{ij}^k)}{\sum_{ij} H_{\epsilon}(\phi_{ij}^k)}, \quad c_2(\phi^k) = \frac{\sum_{ij} u_{ij} (1 - H_{\epsilon}(\phi_{ij}^k))}{\sum_{ij} (1 - H_{\epsilon}(\phi_{ij}^k))} \quad (3.4)$$

Usando 2.15 obtenemos:

$$\begin{aligned} & \phi_{ij}^{k+1} \left[1 + \Delta t \mu \delta_{\epsilon}(\phi_{ij}^k) (A_1/h_x^2 + A_2/h_x^2 + A_3/h_y^2 + A_4/h_y^2) \right] \\ = & \phi_{ij}^k + \Delta t \mu \delta_{\epsilon}(\phi_{ij}^k) \left[(A_1/h_x^2 \phi_{i+1,j}^{k+1} + A_2/h_x^2 \phi_{i-1,j}^{k+1} + A_3/h_y^2 \phi_{i,j+1}^{k+1} + A_4/h_y^2 \phi_{i,j-1}^{k+1}) \right] \\ - & \Delta t \delta_{\epsilon}(\phi_{ij}^k) (\lambda_1 (u_{ij} - c_1(\phi^k))^2 - \lambda_2 (u_{ij} - c_2(\phi^k))^2). \end{aligned} \quad (3.5)$$

el cual se resolverá con el método de Jacobi para obtener ϕ^{k+1} . El algoritmo descrito se podría resumir de la siguiente manera:

1. Inicializar ϕ_0 , en esta tesis tomamos la ϕ inicial como $\phi(x, y) = r - \sqrt{(x - x_0)^2 + (y - y_0)^2}$.
2. Calcular $c_1(\phi^n)$ y $c_2(\phi^n)$ de acuerdo a la ecuación 3.4.
3. Resolver la ecuación diferencial parcial del modelo, por medio de la ecuación 3.5 para obtener la nueva ϕ^{n+1} .
4. Verificar si la solución es estacionaria, en nuestro caso calculamos la proporción de el número de pixeles que cambiaron de región, entre el tamaño de la imagen y si no es menor que una tolerancia dada, regresamos al paso 2.

3.4. Resultados

En esta sección presentamos los principales resultados del modelo Chan-Vese para la segmentación de imágenes. En cada ejemplo mostramos ocho imágenes, en la primera fila tenemos cuatro imágenes en donde se puede observar la evolución de la curva con respecto al paso de las iteraciones y en la segunda fila vemos la segmentación de la imagen con respecto a la curva en la imagen arriba de ella. Los primeros cuatro ejemplos prueban la robustez de este modelo al segmentar imágenes con ruido, así como la efectividad al segmentar algunos objetos de interés, sin embargo, los últimos tres ejemplos, muestran como la no homogeneidad en las imágenes pueden causar dificultades al segmentar con este modelo. En cada ejemplo, los parámetros usados fueron distintos, sin embargo los que se mantuvieron fueron $\lambda_1 = \lambda_2 = 1$ y se usó una iteración de Jacobi, se hicieron pruebas con un mayor número de iteraciones, pero no aumentaba significativamente la razón de convergencia, en cambio si alentaba el algoritmo.

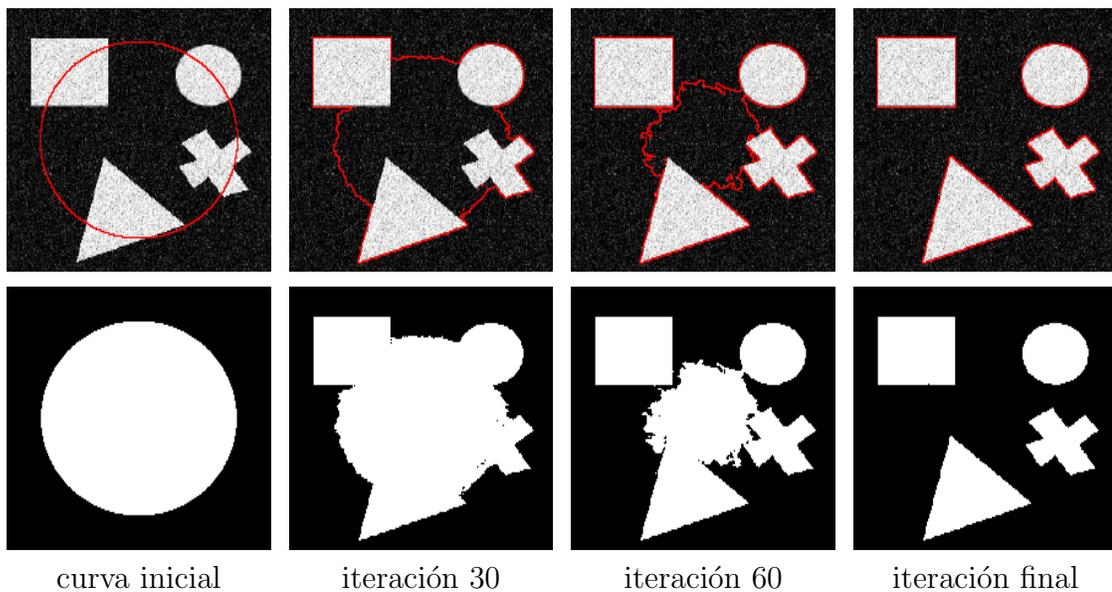


Figura 3.2: Detección de objetos en imágenes con ruido, el número de iteraciones para esta imagen fue de 90

3.4 Resultados

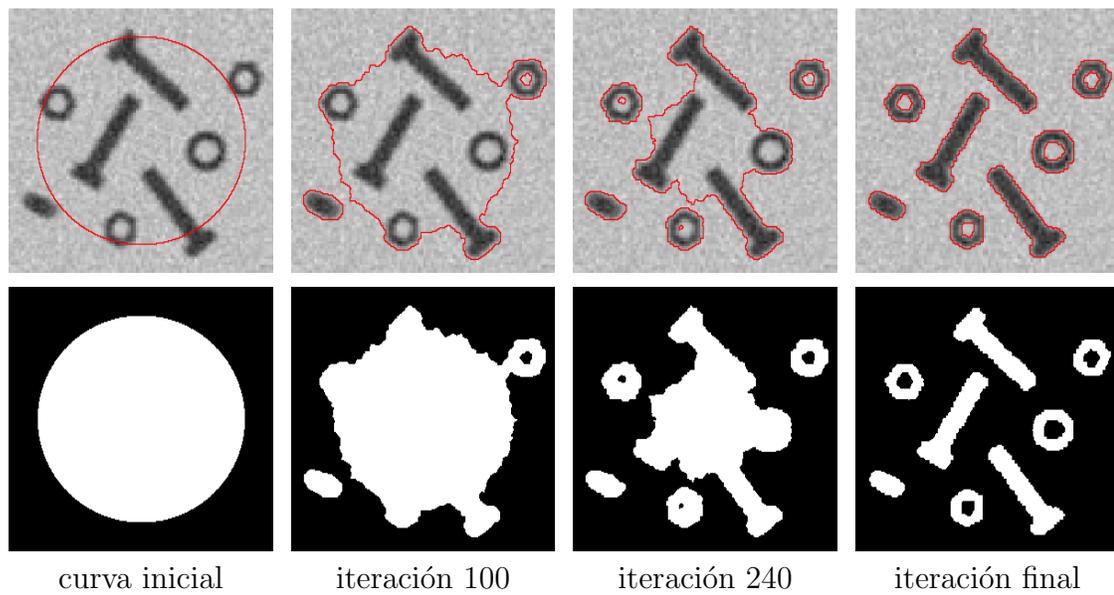


Figura 3.3: Detección de objetos en imágenes con ruido, el número de iteraciones para esta imagen fue de 461

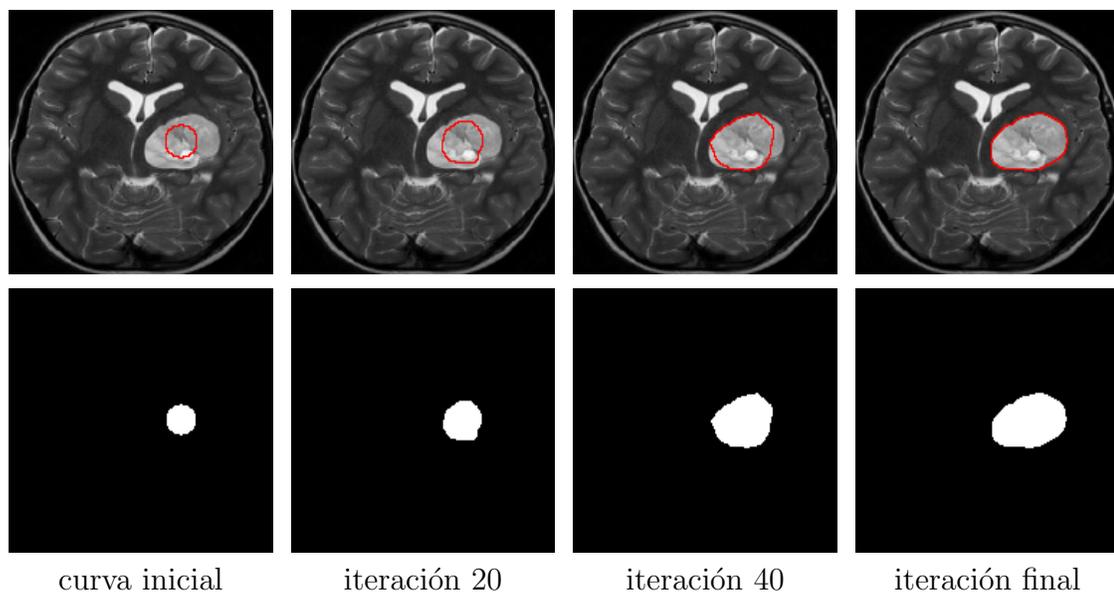


Figura 3.4: Segmentación del tumor de un cerebro, el número de iteraciones para esta imagen fue de 80

3.4 Resultados

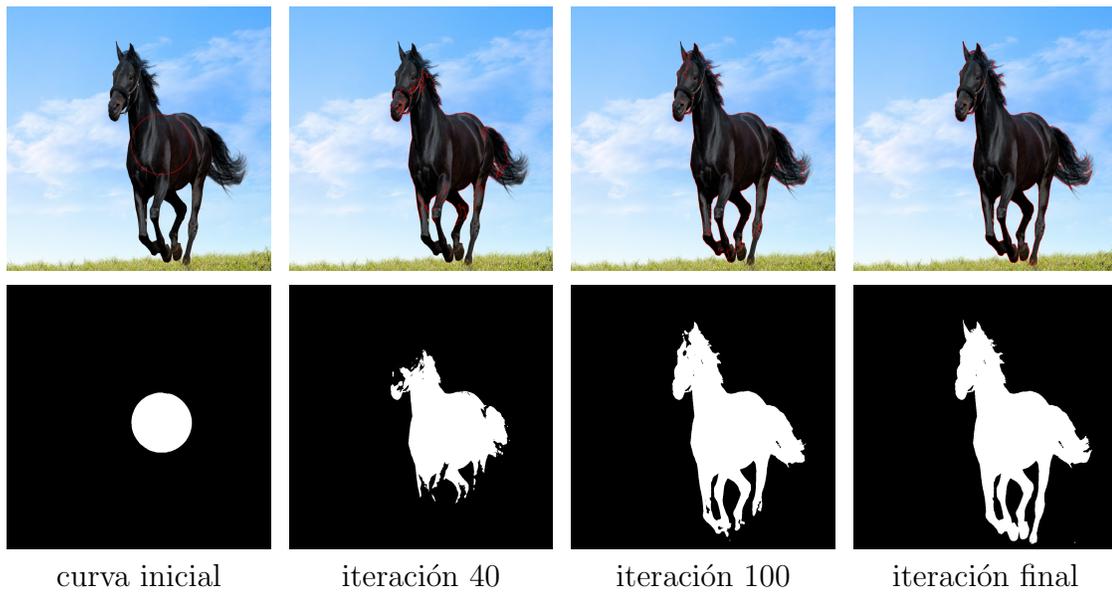


Figura 3.5: Segmentación de un caballo, el número de iteraciones para esta imagen fue de 225

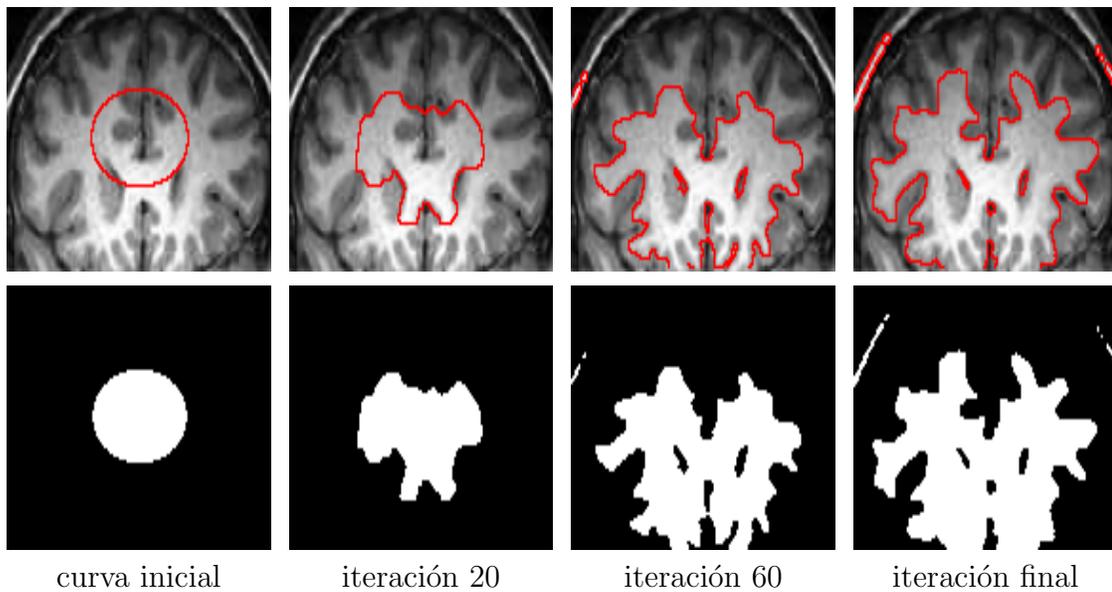


Figura 3.6: Falla al segmentar la materia gris de un cerebro, el número de iteraciones para esta imagen fue de 157

3.4 Resultados

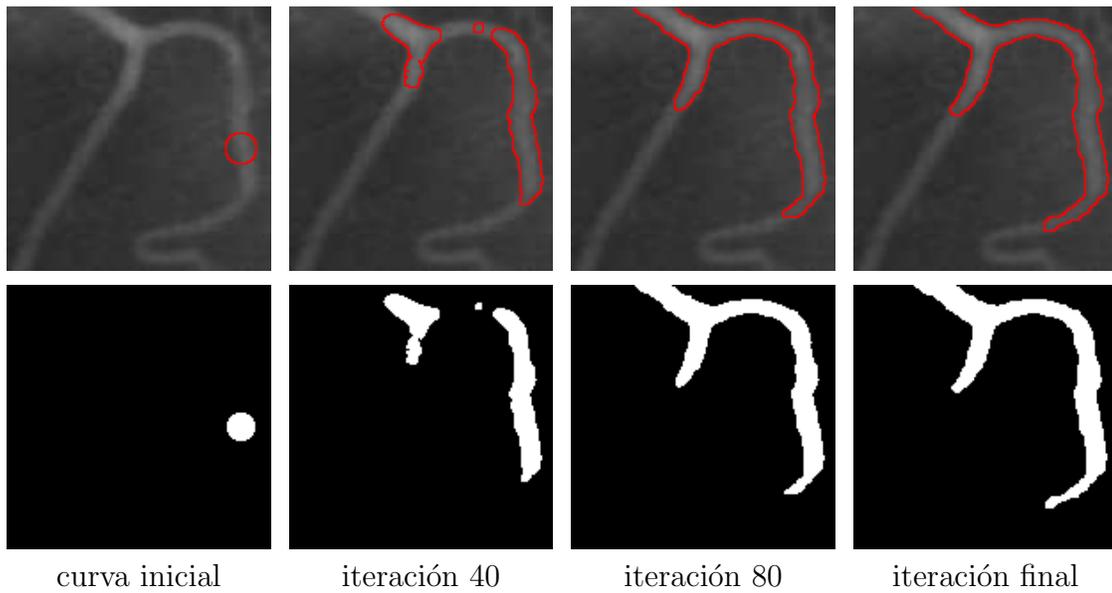


Figura 3.7: Falla al segmentar vasos sanguíneos, el número de iteraciones para esta imagen fue de 130

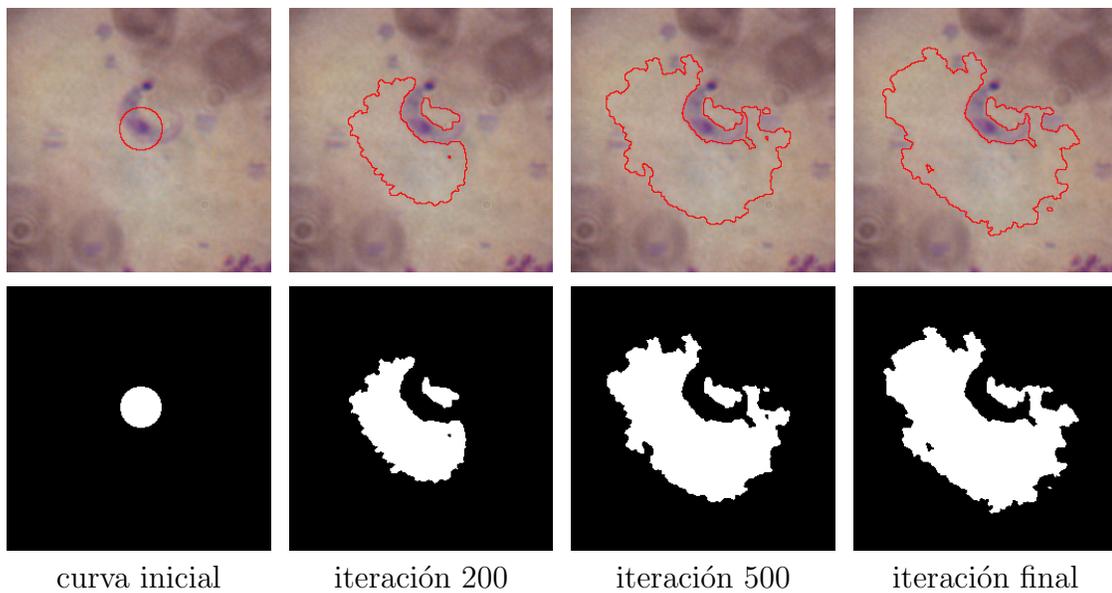


Figura 3.8: Falla al segmentar el Trypanosoma Cruzi, el número de iteraciones para esta imagen fue de 847

3.5. Discusión de resultados

En las figuras 3.2 y 3.3 podemos notar la gran eficiencia y robustez del modelo, al segmentar objetos en imágenes con mucho ruido, escenario en donde algoritmos basados en umbralización o en el gradiente de la imagen no lo lograrían. En la figura 3.4 observamos una de las aplicaciones médicas importantes del modelo de Chan - Vese, en donde se segmenta un tumor en el cerebro, un parámetro importante para este ejemplo es la μ ya que nos ayuda a segmentar objetos cercanos a la curva inicial. En la figura 3.5 mostramos la eficiencia del modelo al segmentar objetos sobresalientes en imágenes, en donde la región a segmentar tiene una distribución homogénea.

En la figura 3.6 ejemplificamos un caso en donde el modelo no obtuvo una buena segmentación de la materia gris del cerebro, esto es debido a que la región en donde se encuentra la materia gris, es no homogénea, es decir, tiene diversos tonos de gris, que aunque al observarlo en ciertas localidades de la región no se nota mucho el cambio (ya que cambia suavemente), al ver la región completa notamos que la varianza de los tonos es grande.

En la figura 3.7 vemos un caso en donde el modelo no obtuvo una buena segmentación de unos vasos sanguíneos, podemos notar como la curva avanza, sin embargo, se detiene antes de segmentarlos por completo, ésto es debido a que si envolviera la parte faltante, la varianza de la región dentro de la curva crecería, lo cual en el modelo no se quiere, sin embargo, notamos que si solo vemos la parte faltante, en sus localidades si hay diferencia en el color entre el vaso sanguíneo y el fondo, lo cual se aprovechará posteriormente.

En la figura 3.8 vemos un ejemplo de lo que es nuestro interés, la segmentación de parásitos en imágenes de frotis de sangre, en este caso la imagen es difícil de segmentar debido a que los objetos que se encuentran alrededor del parásito tienen tonos de gris similares a la de él. La imagen es del parásito *Trypanosoma Cruzi*, en donde, a pesar de haber tomado la curva inicial cercana al parásito, la segmentación no fue buena, ya que segmentó la región que rodea al parásito, y lo demás por tener un tono similar se colocó en una sola región junto con el parásito. Sin embargo, también notamos que aunque globalmente el tono del trypanosoma si coincide con los tonos de los objetos que están a su alrededor, al observarlo en la localidad de él, se distingue claramente entre el fondo y el parásito.

Capítulo 4

Minimización de la Energía de Ajuste de la Región Escalable

En imágenes del mundo real encontramos frecuentemente intensidades no homogéneas, lo cual puede causar muchas dificultades en la segmentación de imágenes. Con el objetivo de superarlas, Chunming Li, Chiu-Yen Kao, John C. Gore y Zhaohua Ding en [14], proponen un modelo de contornos activos que usa la información en regiones locales con una escala controlable.

Este modelo está basado en una segmentación local, es decir dado un punto x en Ω , segmentará solo en una región cerca de él, para controlar que puntos contribuirán a la segmentación local, usaremos una función con la cual si un punto z está muy lejos del punto x la evaluación dará un número muy cercano a 0, es decir que ese punto no contribuirá en la segmentación, mientras que si un punto y está cercano a x la función deberá dar un valor positivo, indicando lo valioso que es en la contribución, es decir, mientras más cerca esté, más grande será el valor que se le asigne, lo cual podemos observar en la figura 4.1.

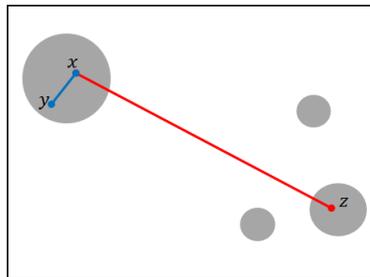


Figura 4.1: Al segmentar en una región de x , el valor del pixel en el punto y debe aportar más en la segmentación, que el valor en el pixel z

4 Minimización de la Energía de Ajuste de la Región Escalable

La función que usaremos es una función kernel no negativa $K : \mathbb{R}^n \rightarrow [0, +\infty)$ con las siguientes propiedades:

1. $K(-u) = K(u)$;
2. $K(u) \geq K(v)$, si $|u| < |v|$, y $\lim_{|u| \rightarrow \infty} K(u) = 0$;
3. $\int K(x)dx = 1$.

A la propiedad 2) la llamamos propiedad de localización del kernel K . La función kernel y su propiedad de localización juegan un papel importante en el método propuesto.

Consideremos una imagen vector valuada $I : \Omega \rightarrow \mathbb{R}^d$, donde $\Omega \subset \mathbb{R}^n$ es el dominio de la imagen, y $d \geq 1$ es la dimensión del vector $I(x)$. En particular $d = 1$ para imágenes en tonos de grises, mientras $d = 3$ para imágenes a color. Sea C una curva cerrada en el dominio de la imagen Ω , el cual separa Ω en dos regiones: $\Omega_1 = fuera(C)$ y $\Omega_2 = dentro(C)$. Para un punto dado $x \in \Omega$, definimos la siguiente energía de ajuste de la intensidad local:

$$\mathcal{E}_x^{Aju}(C, f_1(x), f_2(x)) = \lambda_1 \int_{\Omega_1} K(x-y)|I(y) - f_1(x)|^2 dy + \lambda_2 \int_{\Omega_2} K(x-y)|I(y) - f_2(x)|^2 dy \quad (4.1)$$

donde λ_1 y λ_2 son constantes positivas, y $f_1(x)$ y $f_2(x)$ son dos valores que aproximan intensidades de la imagen en Ω_1 y Ω_2 , respectivamente.

Las intensidades que intervienen efectivamente en la energía de ajuste de arriba están en una región local centrada en el punto, cuyo tamaño puede ser controlada por la función de núcleo. Por tanto, llamamos a la energía de ajuste de la intensidad local anterior una energía de ajuste de la región escalable de un contorno C en un punto x . El kernel utilizado esta dado por

$$K_\sigma(u) = \frac{1}{(2\pi)^{\frac{n}{2}} \sigma^n} e^{-|u|^2/2\sigma^2}, \quad (4.2)$$

con un parámetro de escalamiento $\sigma > 0$.

El término \mathcal{E}_x^{Aju} es un error cuadrático medio ponderado de la aproximación de las intensidades de la imagen $I(y)$ fuera y dentro del contorno C , dadas por el ajuste de los valores

4 Minimización de la Energía de Ajuste de la Región Escalable

$f_1(x)$ y $f_2(x)$, respectivamente, con $K(x-y)$ como el peso asignado a cada intensidad $I(y)$ en y . También debido a la propiedad de localización de la función kernel, la contribución de la intensidad $I(y)$ a la energía de ajuste \mathcal{E}_x^{Aju} decrece y se aproxima a cero cuando el punto y se aleja del punto central x . Por tanto, la energía \mathcal{E}_x^{Aju} es dominada por las intensidades $I(y)$ de los puntos y en una vecindad de x . En particular el kernel gaussiano $K_\sigma(x-y)$ decrece drásticamente a cero cuando y se aleja de x . En términos generales, el kernel gaussiano $K_\sigma(x-y)$ es muy aproximada a cero cuando $|x-y| > 3\sigma$. Por tanto, solo las intensidades en la vecindad $\{y : |x-y| \leq 3\sigma\}$ son dominantes en la energía \mathcal{E}_x^{Aju} . En este sentido, decimos que la energía de ajuste \mathcal{E}_x^{Aju} es localizado alrededor del punto x , como podemos ver en la figura 4.2.

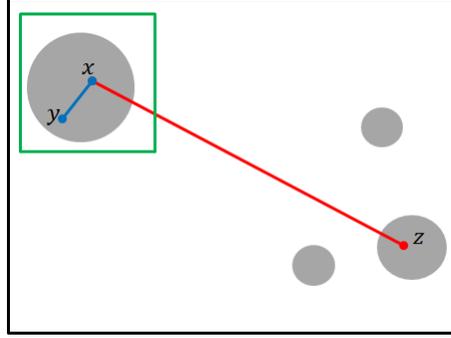


Figura 4.2: El término \mathcal{E}_x^{Aju} es como la energía de Chan-Vese solo que con ponderaciones, aplicado a una región de x .

Dado un punto central x , la energía de ajuste \mathcal{E}_x^{Aju} puede ser minimizada cuando el contorno C esta exactamente en la frontera del objeto y los valores de ajuste f_1 y f_2 optimizan aproximadamente las intensidades locales en la imagen en ambos lados de C . Para obtener la frontera del objeto completamente, debemos encontrar el contorno C que minimiza la energía \mathcal{E}_x^{Aju} , sobre todos los puntos centrales x en el dominio de la imagen, como vemos en la figura 4.3, quedando nuestro término de ajuste de la siguiente manera

$$\mathcal{E}^{re} = \int_{\Omega} \mathcal{E}_x^{Aju}(C, f_1(x), f_2(x)) dx \quad (4.3)$$

4.1 Formulación en conjuntos de nivel

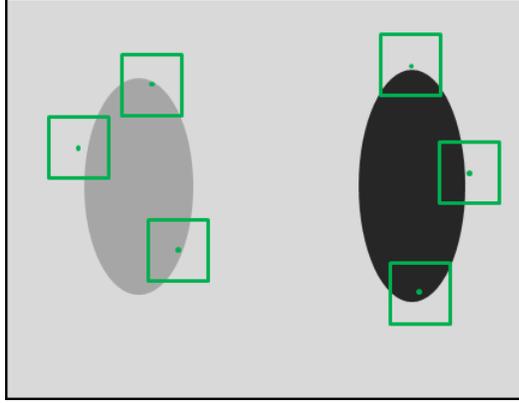


Figura 4.3: El contorno final va a ser el que mejor se ajuste a los contornos de todas las segmentaciones locales centradas en cada pixel de la imagen.

Así, regularizando el funcional a minimizar, agregando el término de longitud de arco, nos queda:

$$\mathcal{F} = \mathcal{E}^{re}(C, f_1, f_2) + \mu \mathcal{L}(C)$$

4.1. Formulación en conjuntos de nivel

En la formulación en conjuntos de nivel, un contorno $C \in \Omega$ es representado por un conjunto de nivel cero de una función Lipschitz $\phi : \Omega \rightarrow \mathbb{R}$, definido como en el capítulo uno y tomando a H como la función de Heaviside, entonces el funcional de energía $\mathcal{E}_x^{Aju}(C, f_1(x), f_2(x))$ puede ser expresado como

$$\mathcal{E}_x^{Aju}(\phi, f_1(x), f_2(x)) = \sum_{i=1}^2 \int K_\sigma(x-y) |I(y) - f_i(x)|^2 M_i(\phi(y)) dy$$

donde $M_1(\phi) = H(\phi)$ y $M_2(\phi) = 1 - H(\phi)$. Así:

$$\mathcal{E}^{re}(\phi, f_1, f_2) = \sum_{i=1}^2 \int \int (K_\sigma(x-y) |I(y) - f_i(x)|^2 M_i(\phi(y)) dy) dx + \mu \int |\nabla H(\phi(x))| dx$$

4.2. Minimización

Regularizando la función de Heaviside y la delta de Dirac como en las ecuaciones 2.10 y 2.11, tenemos que

$$\mathcal{E}_\epsilon^{re}(\phi, f_1, f_2) = \sum_{i=1}^2 \int_{\Omega} \int_{\Omega} (K_\sigma(x-y)|I(y) - f_i(x)|^2 M_i^\epsilon(\phi(y)) dy) dx + \mu \int_{\Omega} |\nabla H_\epsilon(\phi(x))| dx$$

donde $M_1(\phi) = H_\epsilon(\phi)$ y $M_2(\phi) = 1 - H_\epsilon(\phi)$.

Para una función en conjuntos de nivel ϕ , minimizamos el funcional $\mathcal{F}(\phi, f_1, f_2)$ en la ecuación anterior, con respecto a las funciones $f_1(x)$ y $f_2(x)$ (manteniendo ϕ fijo). Por la proposición 2.1.10 se puede mostrar que las derivadas variacionales de $\mathcal{F}(\phi, f_1, f_2)$, con respecto a las funciones $f_1(x)$ y $f_2(x)$ son:

$$\frac{\delta \mathcal{E}_\epsilon^{re}}{\delta f_i}(\phi, f_1, f_2) = -2 \int_{\Omega} K_\sigma(x-y) M_i^\epsilon(\phi(y)) (I(y) - f_i(x)) dy, \quad i = 1, 2.$$

Y como las funciones f_i que minimizan el funcional deben cumplir $\frac{\delta \mathcal{E}_\epsilon^{re}}{\delta f_i}(\phi, f_1, f_2) = 0$, obtenemos:

$$f_i(x) = \frac{K_\sigma(x) * [M_i^\epsilon(\phi(x)) I(x)]}{K_\sigma(x) * M_i^\epsilon(\phi(x))}, \quad i = 1, 2$$

lo cual minimiza el funcional de energía $\mathcal{F}(\phi, f_1, f_2)$ para una ϕ fija.

Manteniendo f_1 y f_2 fijos, minimizamos el funcional de energía $\mathcal{F}(\phi, f_1, f_2)$ con respecto a ϕ , encontrando su derivada variacional, obtenemos:

$$\frac{\delta \mathcal{E}_\epsilon^{re}}{\delta \phi} = \delta_\epsilon(\phi)(\lambda_1 e_1 - \lambda_2 e_2) \tag{4.4}$$

donde e_1, e_2 son las funciones:

$$e_i(x) = \int_{\Omega} K_\sigma(y-x) |I(x) - f_i(y)|^2 dy, \quad i = 1, 2$$

y la función de frontera está dado por:

4.3 Método numérico

$$\partial \mathcal{E}_\epsilon^{re} = 0$$

Usando la ecuación 2.14, obtenemos el siguiente problema de evolución:

$$\begin{cases} \frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 e_1 + \lambda_2 e_2 \right] & \text{en } \Omega, \\ \frac{\partial \phi}{\partial n} = 0 & \text{en } \partial \Omega. \end{cases}$$

En la siguiente sección derivaremos la aproximación numérica del modelo y discutiremos la existencia de métodos usados para resolver la evolución del problema.

4.3. Método numérico

Discreticemos el problema de evolución que tenemos,

$$\begin{cases} f_i(\phi) = \frac{K_\sigma(x) * [M_i^\epsilon(\phi(x))I(x)]}{K_\sigma(x) * M_i^\epsilon(\phi(x))}, \quad i = 1, 2 \\ \frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 e_1 + \lambda_2 e_2 \right] & \text{en } \Omega, \\ \frac{\partial \phi}{\partial n} = 0 & \text{en } \partial \Omega. \end{cases}$$

Dado que el sistema tiene dos funciones además de la ϕ , primero calcularemos los valores de las funciones f_1 y f_2 en Ω , y luego sustituiremos estos valores en la ecuación de evolución, una vez encontrada la nueva ϕ , actualizamos los valores de f_1 y f_2 y continuamos así sucesivamente.

Para la discretización de las f_i , usaremos la convolución usual en el procesamiento de imágenes, en donde la matriz k es una máscara de $(2w + 1) \times (2w + 1)$, donde w es 2σ , $i_0 = i - w$ y $j_0 = j - w$.

$$[k * I]_{ij} = \sum_{u=0, v=0}^{w, w} k_{u,v} \cdot I_{u+i_0, v+j_0} \quad (4.5)$$

Usando la ecuación 2.15 obtenemos:

4.3 Método numérico

$$\begin{aligned}
& \phi_{ij}^{k+1} [1 + \Delta t \mu \delta_\epsilon(\phi_{ij}^k)(A_1/h_x^2 + A_2/h_x^2 + A_3/h_y^2 + A_4/h_y^2)] \\
= & \phi_{ij}^k + \Delta t \mu \delta_\epsilon(\phi_{ij}^k) [(A_1/h_x^2 \phi_{i+1,j}^{k+1} + A_2/h_x^2 \phi_{i-1,j}^{k+1} + A_3/h_y^2 \phi_{i,j+1}^{k+1} + A_4/h_y^2 \phi_{i,j-1}^{k+1})] \\
- & \Delta t \delta_\epsilon(\phi_{ij}^k)(\lambda_1 e_1 - \lambda_2 e_2).
\end{aligned} \tag{4.6}$$

El mayor costo computacional es el calculo de f_1 y f_2 y $\lambda_1 e_1 - \lambda_2 e_2$ en la evolución de conjuntos de nivel. Para las f_i s hay en total cuatro convoluciones en el numerador y denominador. Sin embargo notemos que f_2 puede ser escrito como

$$f_2 = \frac{K_\sigma * I - K_\sigma * [H_\epsilon(\phi)I]}{K_\sigma * \mathbf{1} - K_\sigma * H_\epsilon(\phi)} \tag{4.7}$$

donde $\mathbf{1}$ es la función constante con valor 1. En donde los segundos términos del numerador y denominador son los mismos que el de f_1 , mientras los términos $K_\sigma * I$ en el numerador y $K_\sigma * \mathbf{1}$ en el denominador no dependen en la evolución de conjuntos de nivel de ϕ . Además, las dos convoluciones pueden ser calculados solo una vez antes de las iteraciones. Durante las iteraciones, solo dos convoluciones son calculadas para la evolución de ϕ . El término $\lambda_1 e_1 - \lambda_2 e_2$ puede ser expresado como una combinación de tres convoluciones, con una convolución independiente de ϕ , la cual puede ser calculado solo una vez antes de las iteraciones, además usaremos el hecho de que la convolución con un kernel gaussiano de dimensión dos, puede ser visto como la convolución con dos kernels gaussianos de dimensión 1.

El algoritmo descrito se podría resumir de la siguiente manera:

1. Inicializar ϕ_0 , en esta tesis tomamos la ϕ inicial como $\phi(x, y) = r - \sqrt{(x - x_0)^2 + (y - y_0)^2}$.
2. Calcular f_1 y f_2 , de acuerdo a las ecuaciones 4.5 y 4.7.
3. Resolver la ecuación diferencial parcial del modelo, por medio de la ecuación 4.6 para obtener la nueva ϕ^{n+1} .
4. Verificar si la solución es estacionaria, en nuestro caso calculamos la proporción de el número de pixeles que cambiaron de región, entre el tamaño de la imagen y si no es menor que una tolerancia dada, regresamos al paso 2.

4.4. Resultados

En esta sección presentamos los principales resultados del modelo de la minimización de la energía de ajuste de la región escalable para la segmentación de imágenes. En los primeros tres ejemplos mostramos ocho imágenes, en la primera fila tenemos cuatro imágenes en donde se puede observar la evolución de la curva con respecto al paso de las iteraciones y en la segunda fila vemos la segmentación de la imagen con respecto a la curva en la imagen arriba de ella. Estos ejemplos prueban como el modelo es robusto frente a la no homogeneidad en las imágenes, lo cual no pueden resolver modelos como el Chan-Vese. En las siguientes dos imágenes se presentan las aplicaciones del modelo a la segmentación de parásitos en imágenes de muestras de sangre, en la fila de arriba se encuentran las curvas finales de la evolución en cada caso y en la segunda fila se encuentra su respectiva segmentación. En cada ejemplo, los parámetros usados fueron distintos, sin embargo los que se mantuvieron fueron $\lambda_1 = \lambda_2 = 1$ y se usó una iteración de Jacobi, se hicieron pruebas con un mayor número de iteraciones, pero no aumentaba significativamente la razón de convergencia, en cambio si alentaba el algoritmo.

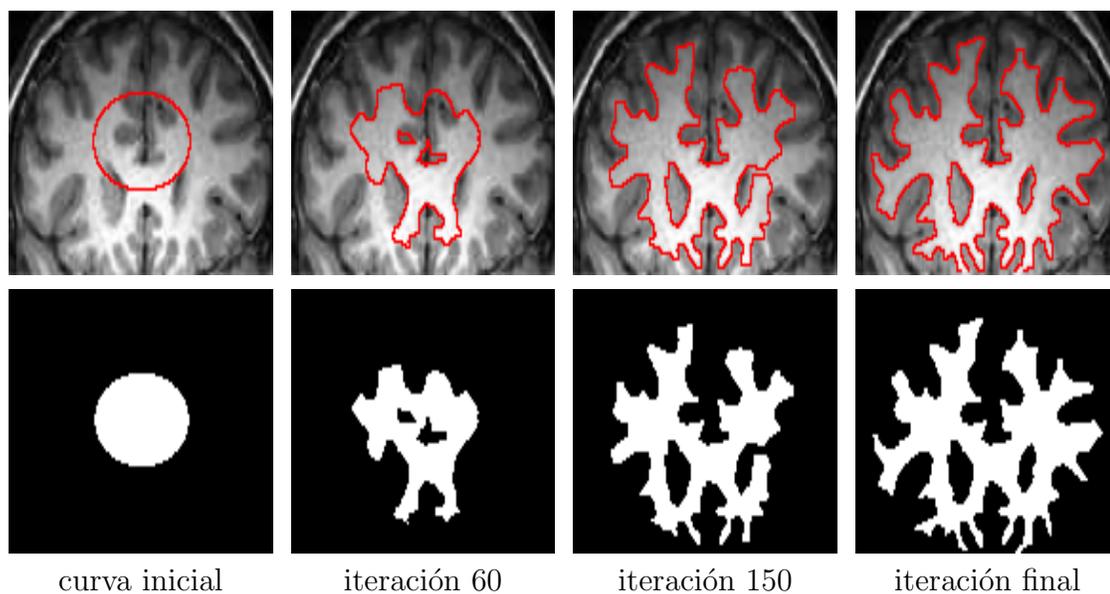


Figura 4.4: Segmentación de la materia gris de un cerebro, el número de iteraciones para esta imagen fue de 269

4.4 Resultados

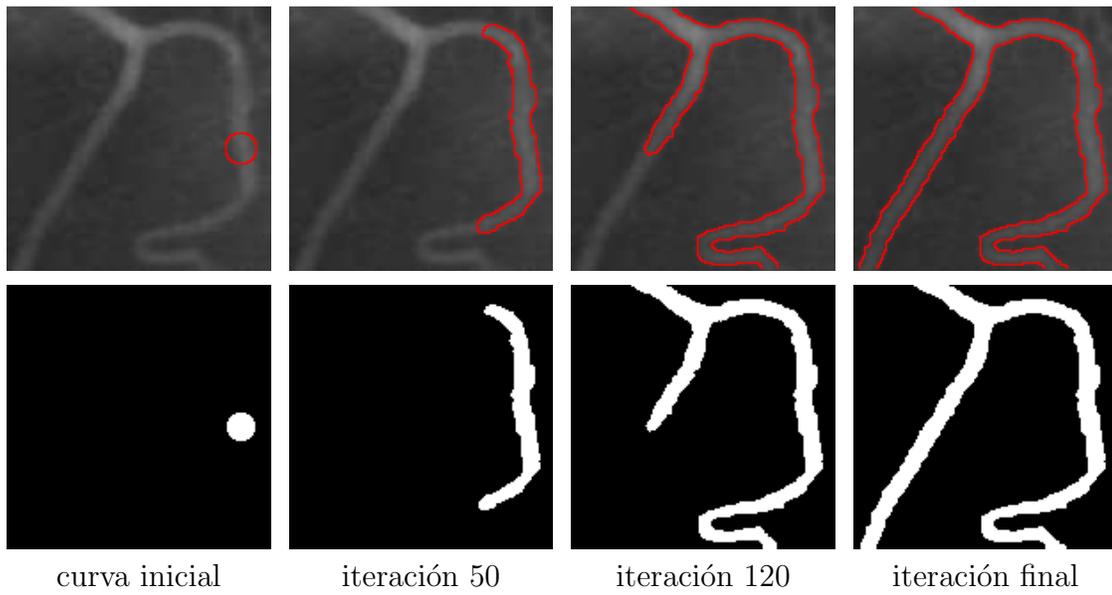


Figura 4.5: Segmentación de vasos sanguíneos, el número de iteraciones para esta imagen fue de 237

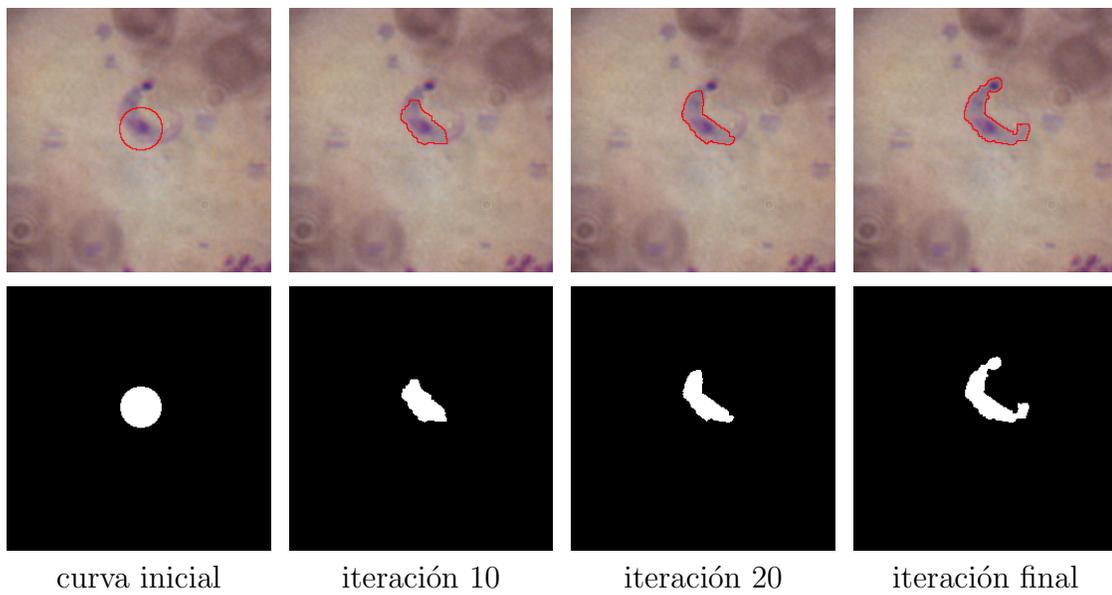


Figura 4.6: Segmentación del Trypanosoma Cruzi, el número de iteraciones para esta imagen fue de 71

4.4 Resultados

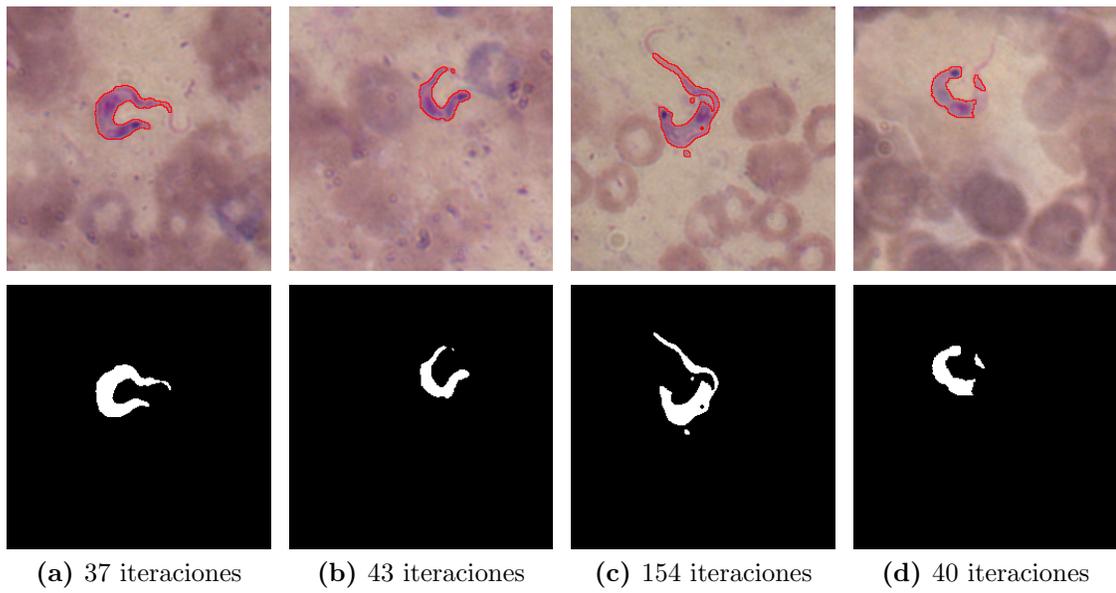


Figura 4.7: Segmentación del Trypanosoma Cruzi en varias muestras de sangre

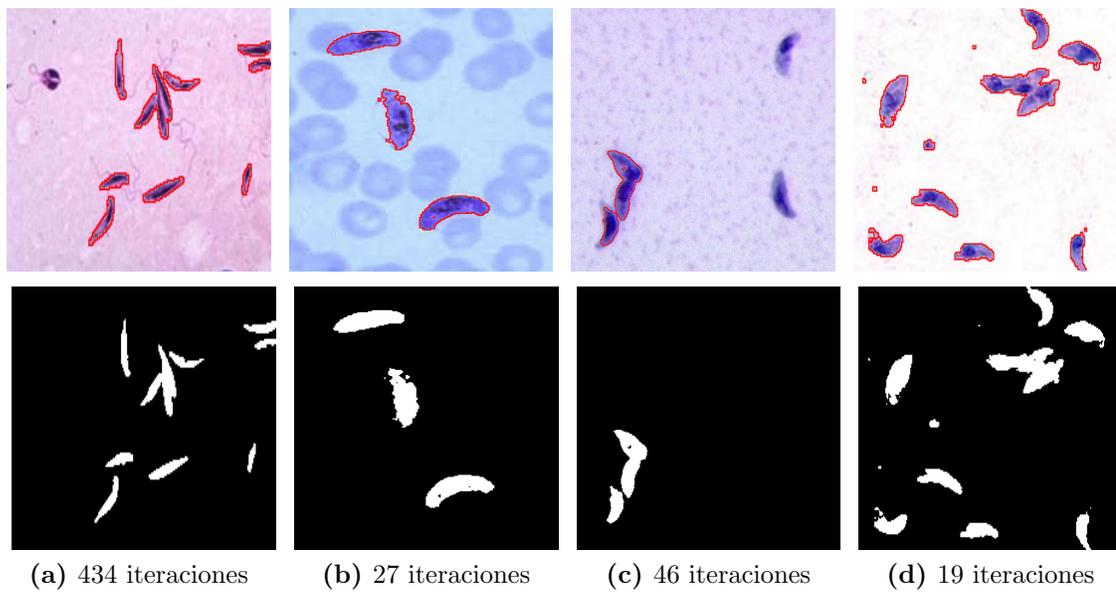


Figura 4.8: Segmentación de la leishmania (a), plasmodium (b) y el toxoplasma gondii(d) y (c)

4.5. Discusión de resultados

En las primeras tres figuras, mostramos la eficiencia de este modelo, resolviendo los problemas que observamos en el capítulo anterior con el modelo de contornos activos sin bordes, en la figura 4.4 se observa una correcta segmentación de la materia gris del cerebro, en donde a pesar de sus tonos no homogéneos, el resultado es muy bueno, en la figura 4.5 vemos también una buena segmentación de los vasos sanguíneos, en donde a pesar de que haya varios tonos en el objeto, la curva si continuó su evolución, esto es debido a que el cambio entre estos tonos es de forma suave, con lo cual al verlo en sus localidades pareciera ser el mismo, en la figura 4.6 observamos una muy buena segmentación del parásito *Trypanosoma Cruzi*, en donde a pesar de que en la imagen hay objetos con tonos parecidos a los de él, se aprovecha que en su localidad es clara la diferencia entre el parásito y el fondo.

En la figura 4.7 mostramos varios ejemplos de segmentación del parásito *Trypanosoma Cruzi*, el algoritmo en estos casos se corrió tomando como curva inicial un circulito centrado en el pixel donde esta el mínimo del canal verde, el cual coincide con el cinetoplasto (similar a un ojito), lo cual da la idea de una posible automatización en la curva inicial, en la figura observamos que en las primeras tres imágenes hay una correcta segmentación, sin embargo en la cuarta hay una parte del trypanosoma que no fue correctamente segmentado, esto es debido a que el tono que rodea al parásito, realmente en su localidad si se parece al tono de la parte faltante, por eso el algoritmo tomó esa segmentación.

En la figura 4.8 mostramos varios ejemplos en donde se pueden observar buenas segmentaciones, sin embargo, en la imagen tres se nota como el modelo es sensible a la curva inicial, ya que si hay objetos lejanos no los segmentará a menos que haya otros objetos entre ellos, de manera que durante la evolución, la curva pueda ir avanzando hasta llegar al objeto más distante.

Capítulo 5

Contornos Activos para la Segmentación de Distribuciones

El histograma de intensidad o de color de objetos específicos nos da pistas globales y robustas para segmentar objetos en imágenes, en este modelo el objetivo es segmentar una región en una imagen cuya distribución de intensidades o de color mejor se parezca a una previamente dada. Este modelo es parte de las investigaciones hechas por Daniel Freedman y Thao Zhang en [23] y [22].

Sea Z el conjunto de valores fotométricos de la imagen, por ejemplo valores enteros entre 0 y 255 para imágenes en escala de grises o vectores en \mathbb{Z}^3 con entradas entre 0 y 255 para imágenes a color, y sea $P_o(z)$ la función de densidad de probabilidad (FDP) de los valores de los píxeles de un objeto de interés en una imagen. La segmentación de la imagen se realizará encontrando una curva que encierre una región en la imagen cuya FDP mejor se parezca a la FDP dada ($P_o(z)$). Sea $P_+(z | \Gamma)$ la FDP estimada de la región dentro de la curva Γ . $P_+(z | \Gamma)$ puede ser estimada por el método de Parzen [25] como:

$$P_+(z | \Gamma) = \frac{\int_{\Omega_1} K_\sigma(z - I(x)) dx}{\int_{\Omega_1} dx}$$

donde K_σ es el kernel Gaussiano dado por la ecuación 4.2 con $n = 1$. Así por ejemplo, usando $\sigma = 3$, obtenemos que el histograma de frecuencias y el histograma por el método de Parzen, del Trypanosoma Cruzi en la imagen 4.6, son los que se grafican en la figura 5.1.

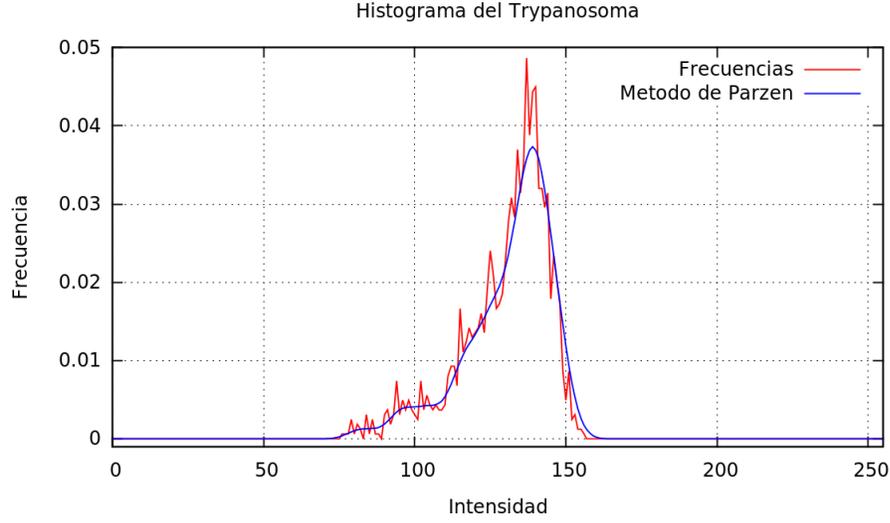


Figura 5.1: Histograma de frecuencias e histograma con el método de Parzen

Por la gráfica podemos notar que el método de Parzen parece solo suavizar el histograma de frecuencias, para ver esto, notemos las siguientes igualdades en la discretización:

$$\frac{\sum_{\Omega_1} K_\sigma(z - I(x))}{\sum_{\Omega_1} 1} = \sum_{u=0}^{255} K_\sigma(z - u) Hist(u),$$

siendo $Hist$ el histograma de frecuencias. Notemos que el lado derecho es la discretización de $K_\sigma * Hist$, lo cual nos muestra que efectivamente el método de Parzen es en realidad un suavizamiento del histograma de frecuencias.

Para medir la diferencia entre FDPs usaremos el coeficiente de Bhattacharyya [26], el cual para dos FDPs $p(z), q(z)$ con dominio en Z , está definido como sigue:

$$\mathcal{B}(p, q) = \sum_Z \sqrt{p(z)q(z)} \quad (5.1)$$

Notar que los valores de \mathcal{B} están en $[0, 1]$, donde 0 indica que no hay superposición, y 1 indica una perfecta superposición. Así el término de ajuste estaría dado por:

$$\mathcal{E}_{\mathcal{B}} = -\mathcal{B}(P_o(\cdot), P_+(\cdot | \Gamma))$$

5.1 Formulación en conjuntos de nivel

cuyo mínimo daría la curva que encierra la región en una imagen cuya FDP (P_+) más se parece a la FDP dada (P_o). Usando la ecuación 5.1 el término de ajuste quedaría:

$$\mathcal{E}_{\mathcal{B}}(\Gamma) = - \sum_Z \sqrt{P_o(z)P_+(z | \Gamma)} \quad (5.2)$$

5.1. Formulación en conjuntos de nivel

En la formulación en conjuntos de nivel, un contorno $C \in \Omega$ es representado por un conjunto de nivel cero de una función Lipschitz $\phi : \Omega \rightarrow \mathbb{R}$, tal como fue definido en el capítulo uno y tomando a H como la función de Heaviside, entonces la función de densidad dentro de la curva sera estimada por:

$$P_+(z|\phi) = \frac{\int_{\Omega} K_{\sigma}(z - I(x))H(\phi)dx}{\int_{\Omega} H(\phi)dx}$$

Así el término de ajuste en conjuntos de nivel, quedaría

$$\mathcal{E}_{\mathcal{B}}(\phi) = - \sum_Z \sqrt{P_o(z)P_+(z | \phi)} \quad (5.3)$$

5.2. Minimización

Para minimizar el funcional, tomamos las versiones suaves de la función de Heaviside y la delta de Dirac dadas por las ecuaciones 2.11 y 2.10, así obtenemos:

$$P_+(z|\phi) = \frac{\int_{\Omega} K_{\sigma}(z - I(x))H_{\epsilon}(\phi)dx}{\int_{\Omega} H_{\epsilon}(\phi)dx}$$

$$\mathcal{E}_{\epsilon}^{\mathcal{B}}(\phi) = - \sum_Z \sqrt{P_o(z)P_+(z | \phi)}$$

Usando la ecuación 2.12 , obtenemos que su derivada variacional es

$$\frac{\delta \mathcal{E}_{\mathcal{B}}}{\delta \phi} = - \frac{\delta(\phi)}{2 \int_{\Omega} H(\phi)dx} \sum_Z \sqrt{\frac{P_o(z)}{P_+(z | \phi)}} (K_{\sigma}(z - I) - P_+(z | \phi)) dz \quad (5.4)$$

y la función de frontera está dado por:

5.3 Método numérico

$$\partial \mathcal{E}_\epsilon^{\mathcal{B}} = 0$$

Usando la ecuación 2.14, obtenemos el siguiente problema de evolución:

$$\begin{cases} \frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \frac{\delta(\phi)}{2 \int_\Omega H(\phi) dx} \sum_Z \sqrt{\frac{P_o(z)}{P_+(z | \phi)}} (K_\sigma(z - I) - P_+(z | \phi)) \right] & \text{en } \Omega, \\ \frac{\partial \phi}{\partial n} = 0 & \text{en } \partial\Omega. \end{cases}$$

5.3. Método numérico

Usando la ecuación 2.15 obtenemos la discretización de nuestro problema de evolución:

$$\begin{aligned} & \phi_{ij}^{k+1} [1 + \Delta t \mu \delta_\epsilon(\phi_{ij}^k) (A_1/h_x^2 + A_2/h_x^2 + A_3/h_y^2 + A_4/h_y^2)] \\ = & \phi_{ij}^k + \Delta t \mu \delta_\epsilon(\phi_{ij}^k) [(A_1/h_x^2 \phi_{i+1,j}^{k+1} + A_2/h_x^2 \phi_{i-1,j}^{k+1} + A_3/h_y^2 \phi_{i,j+1}^{k+1} + A_4/h_y^2 \phi_{i,j-1}^{k+1})] \quad (5.5) \\ + & \Delta t \delta_\epsilon \frac{\delta(\phi^k)}{2 \int_\Omega H(\phi^k) dx} \sum_Z \sqrt{\frac{P_o(z)}{P_+(z | \phi^k)}} (K_\sigma(z - I) - P_+(z | \phi^k)). \end{aligned}$$

El algoritmo descrito se podría resumir de la siguiente manera:

1. Inicializar ϕ_0 , en esta tesis tomamos la ϕ inicial como $\phi(x, y) = r - \sqrt{(x - x_0)^2 + (y - y_0)^2}$.
2. Resolver la ecuación diferencial parcial del modelo, por medio de la ecuación 5.5 para obtener la nueva ϕ^{n+1} .
3. Verificar si la solución es estacionaria, en nuestro caso calculamos la proporción de el número de pixeles que cambiaron de región, entre el tamaño de la imagen y si no es menor que una tolerancia dada, regresamos al paso 2.

5.4. Resultados

En esta sección presentamos los principales resultados del modelo basado en la distancia de Bhattacharyya para la segmentación de imágenes. En cada ejemplo, los parámetros usados fueron distintos, se uso una iteración de Jacobi, se hicieron pruebas con un mayor número de iteraciones, pero no aumentaba significativamente la razón de convergencia, en cambio si alentaba el algoritmo.

Se muestran tres ejemplos con imágenes sintéticas, en donde se puede ver la utilidad de este modelo para encontrar objetos en una imagen, cuya distribución sea similar a una dada previamente. También damos un ejemplo de aplicación en la segmentación de parásitos en imágenes de muestras de sangre, en donde el parásito es el toxoplasma gondii.

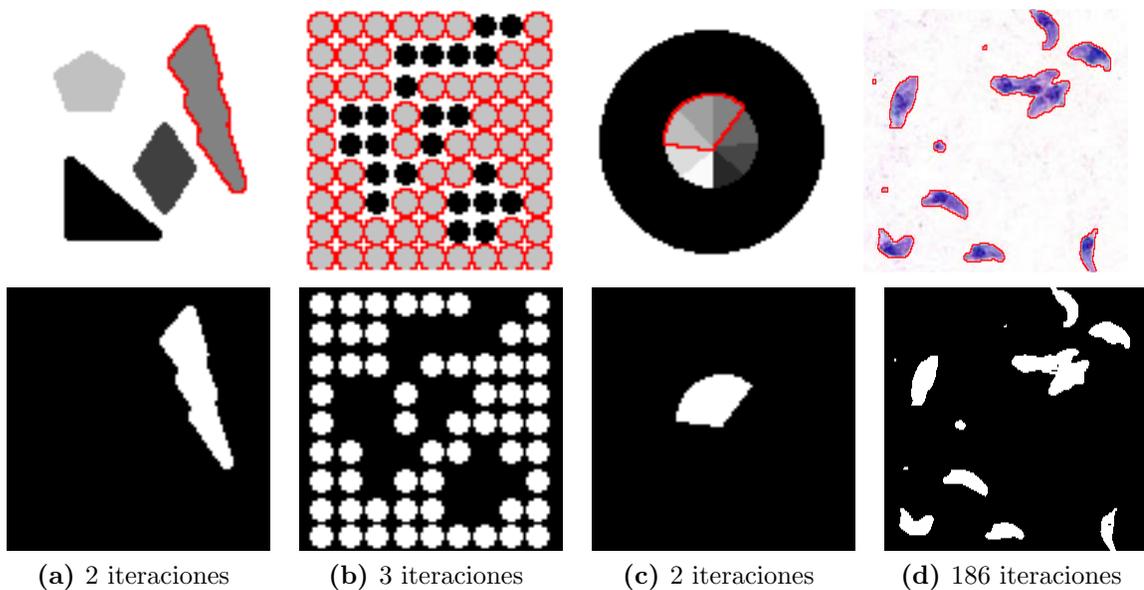


Figura 5.2: Ejemplos de segmentación usando la distancia de Bhattacharyya para 3 imágenes sintéticas (a), (b) y (c), y para el parásito toxoplasma gondii (d)

5.5. Discusión de resultados

En la figura 5.2 se muestra la utilidad que puede tener el modelo anteriormente visto, sobre todo cuando se quiere segmentar objetos específicos en imágenes.

5.5 Discusión de resultados

En la primera imagen 5.2a, se observan varias figuras con distintos tonos de gris, al algoritmo se le paso la distribución de los tonos de gris del objeto que se observa en la segmentación obtenida, por lo que vemos la eficacia del modelo.

En la imagen 5.2b, le pasamos al algoritmo la distribución de los tonos de gris de las bolitas grises, y obtuvimos buenos resultados, ya que exactamente solo esas segmentó.

En la imagen 5.2c, le pasamos una distribución de tonos de gris de la región segmentada, en donde notamos al menos tres tonalidades de gris, y vemos que el algoritmo sigue funcionando muy bien al segmentar exactamente la región deseada de la imagen.

En la figura 5.2d mostramos un ejemplo de la aplicación de este modelo a la segmentación de parásitos en imágenes de muestras de sangre, en donde vemos que obtuvo buenos resultados, sin embargo, en imágenes más complicadas como la del trypanosoma, el algoritmo no obtuvo una buena segmentación debido a la presencia de las mismas tonalidades en otros objetos distintos al parásito.

Capítulo 6

Comentarios finales

6.1. Resultados de la paralelización

En esta sección presentamos los resultados obtenidos de la paralelización del modelo Chan - Vese, la cual fue realizada en GPUs usando la librería CUDA y la librería de plantillas TRUST en C++, notemos la gran mejoría que tiene en la velocidad con respecto al algoritmo secuencial, lo cual sugiere este método como una forma eficaz de mejorar el tiempo de ejecución de nuestros algoritmos basados en modelos de contornos activos.

| Tipo | Tamaño | Secuencial | Paralelo | Mejora |
|-----------|-----------------------|------------|----------|---------|
| figuras | $216px \times 166px$ | 0.78 s | 0.05 s | 14.85 × |
| tornillos | $289px \times 226px$ | 6.74 s | 0.34 s | 20× |
| tumor | $184px \times 167px$ | 0.59 s | 0.05 s | 12.87 × |
| caballo | $1024px \times 768px$ | 41.11 s | 1.18 s | 34.76 × |
| mat. gris | $139px \times 91px$ | 0.49s | 0.73s | 6.71 × |
| vasos | $149px \times 149px$ | 0.63 s | 0.06 s | 10.27 × |
| chagas | $256px \times 256px$ | 14.89 s | 0.75 s | 19.84 × |

Los experimentos se realizaron en una computadora con las siguientes características:

- Nombre del modelo de procesador : Intel(R) Core(TM) i7 CPU 930
- CPU GHz : 2.80
- Hay 1 dispositivos soportando CUDA
- Dispositivo 0: "GeForce GTS 450"

6.2 Conclusiones

- Multiprocesadores x Núcleos/MP = Núcleos: 3 (MP) x 8 (Núcleos/MP) = 24 (Núcleos)
- Frecuencia de reloj: 1579000 kilohertz

6.2. Conclusiones

Por los resultados obtenidos en la aplicación de los modelos de contornos activos en la segmentación de parásitos en imágenes de muestras de sangre, podemos concluir que son modelos robustos (en cuanto a la presencia de ruido), obtienen una buena segmentación en las imágenes en donde los parásitos presentan un mismo tono en su totalidad, aunque sea difuso o no presente contornos bien definidos y tienen mucha flexibilidad para su formulación al poder elegir que es lo que se desea segmentar cambiando el término de ajuste de la curva.

En los resultados de la paralelización podemos notar la gran mejoría que se obtiene en cuanto al tiempo de ejecución, lo cual sugiere que la paralelización en GPUs es un método muy confiable para mejorar la velocidad de los algoritmos basados en modelos de contornos activos. Notemos además que la potencia que obtuvimos fue usando solo 24 núcleos de procesamiento GPU, lo cual sugiere que si corriéramos el programa en una maquina con un mayor número de núcleos la velocidad aumentaría aun más.

Los resultados obtenidos con la distancia de Bhattacharyya no parecen muy prometedores, ésto es debido a que usamos tonos de grises, si usáramos los tres canales de color, quizá los parásitos se distinguirían mejor.

El modelo más prometedor es el de región escalable, sin embargo, este modelo aun tiene sus problemas, ya que si la región que rodea al objeto se parece a una partecita de el, la curva que segmenta lo pondrá fuera del objeto, además como el ajuste de los valores de las f_i 's se va haciendo conforme la curva evoluciona en cada localidad, esto puede causar que en algún momento dado la curva divida a la imagen al toparse con otro objeto con una distribución distinta de la que queremos segmentar, es decir, obtener que dos objetos con la misma distribución pero muy separados, se coloquen uno dentro y otro fuera de la curva, obteniendo así una mala segmentación. Una posible solución a esto sería implementar el modelo de región escalable tomando la distancia de bhattacharyya y así conservar la

6.3 Trabajos futuros

distribución que queremos dentro de la curva.

6.3. Trabajos futuros

Algunos de los trabajos futuros que se pueden realizar a partir de esta tesis son los siguientes:

1. Mejorar la paralelización del modelo Chan - Vese incorporando la memoria de textura, ya que ésta es la que tiene el acceso más optimizado, sin embargo se tendría que analizar la forma de como hacerlo. La importancia de seguir mejorando la paralelización de este modelo, radica en que se puede adaptar a cualquier otro modelo de contornos activos.
2. Agregar las ideas del modelo para la segmentación de distribuciones al modelo de la minimización de la energía de ajuste de la región escalable, asegurando de esta manera que la región dentro de la curva tenga la misma distribución.
3. Agregar información sobre la forma de los parásitos, al modelo de la minimización de la energía de ajuste de la región escalable, ya que lo único que se ha tomado fueron los tonos del color dentro de la curva y no la forma que tendrá ésta.
4. Implementar el modelo para la segmentación de distribuciones a color, tomando en cuenta que lo más recomendable es paralelizarlo, debido al tiempo que tomará en realizar las iteraciones.

Apéndice A

Paralelización

A.1. Historia

Uno de los métodos más importante para la mejora del rendimiento en dispositivos informáticos de consumo ha sido el aumento en la velocidad a la que el reloj del procesador funciona. Basta ver que en las primeras computadoras de la década de los 80s, las unidades de procesamiento central (CPUs) corrían con relojes internos de funcionamiento alrededor de 1MHz y ahora la mayoría de los procesadores de escritorio vienen con velocidades de reloj entre 1GHz y 4GHz, casi 1000 veces más rápido que el reloj de la computadora personal original.

En los últimos años, sin embargo, los fabricantes se han visto obligados a buscar alternativas a esta fuente tradicional de una mayor potencia de cálculo. Debido a diversas limitaciones fundamentales en la fabricación de circuitos integrados, ya no es factible depender del ascenso en espiral de las velocidades de reloj del procesador como un medio para extraer energía adicional de arquitecturas existentes. Debido a las restricciones de energía y calor, así como un límite físico se acerca rápidamente al tamaño de los transistores, los investigadores y los fabricantes han empezado a buscar en otra parte.

Por otro lado las supercomputadoras también han extraído de forma similar durante décadas enormes ganancias. El rendimiento en un procesador utilizado ha subido enormemente de forma similar como en las CPUs de los ordenadores personales. Sin embargo, además de mejorar un solo procesador, los fabricantes han aumentado el rendimiento agregando más procesadores a las supercomputadoras. Con esto observamos que los superordenadores más rápidos tienen decenas o cientos de miles de núcleos de procesadores trabajando en paralelo.

A.2 CUDA

Buscando una mejora en la potencia de procesamiento adicional para los ordenadores personales, esta mejora en las supercomputadoras planteó una pregunta muy buena: ¿en vez de sólo buscar aumentar el rendimiento de un solo núcleo de procesamiento, por qué no poner más de una en una computadora personal? De esta manera, las computadoras personales podrían seguir mejorando sin la necesidad del aumento en la velocidad del reloj de procesador.

En 2005, los principales fabricantes de CPU comenzaron a ofrecer las computadoras con más de un núcleo de procesamiento. Así en los años siguientes, el desarrollo continuó con tres, cuatro, seis y ocho núcleos de unidades centrales de proceso. Algunas veces conocido como la revolución de múltiples núcleos, la cual ha marcado un gran cambio en la informática de consumo y lo cual confirma que la computación en paralelo ha llegado [27].

A.2. CUDA

CUDA son las siglas de Compute Unified Device Architecture que hace referencia tanto a un compilador como a un conjunto de herramientas de desarrollo creadas por Nvidia que permiten a los programadores usar una variación del lenguaje de programación C para codificar algoritmos en GPUs de Nvidia¹.

CUDA intenta explotar las ventajas de las GPUs frente a las CPUs de propósito general utilizando el paralelismo que ofrecen sus múltiples núcleos, que permiten el lanzamiento de un altísimo número de hilos simultáneos. Por ello, si una aplicación está diseñada utilizando numerosos hilos que realizan tareas independientes (que es lo que hacen las GPUs al procesar gráficos, su tarea natural), una GPU podrá ofrecer un gran rendimiento en campos que podrían ir desde la biología computacional a la criptografía por ejemplo.

El modelo de programación de CUDA está diseñado para que se creen aplicaciones que de forma transparente escalen su paralelismo para poder incrementar el número de núcleos computacionales. Este diseño contiene tres puntos claves, que son la jerarquía de grupos de hilos, las memorias compartidas y las barreras de sincronización [27], [28] y [29].

¹Nvidia es uno de los grandes líderes del sector de tecnologías de visualización digital y la inventora de la GPU.

A.3. Programando en CUDA

A continuación presentamos una breve introducción a la programación en CUDA, para una mejor aproximación puede consultar [27] y [28].

La estructura que se utiliza en este modelo está definida por una malla, dentro de la cual hay bloques de hilos que están formados por como máximo 512 hilos distintos.

Cada hilo está identificado con un identificador único, que se accede con la variable `threadIdx`. Esta variable es muy útil para repartir el trabajo entre distintos hilos. `threadIdx` tiene 3 componentes (x, y, z) , coincidiendo con las dimensiones de bloques de hilos. Así, cada elemento de una matriz, por ejemplo, lo podría tratar su homólogo en un bloque de hilos de dos dimensiones.

Al igual que los hilos, los bloques se identifican mediante `blockIdx` (en este caso con dos componentes x e y). Otro parámetro útil es `blockDim`, para acceder al tamaño de bloque.

A.3.1. Kernel

Un kernel en “C for CUDA”, es una función la cual al ejecutarse lo hará en N distintos hilos en lugar de en secuencial. Se define incluyendo `__global__` en la declaración. Por ejemplo:

```
//Definicion del kernel  
__global__ void f(int a, int b, int c)  
{  
}
```

Si en nuestra función `f` queremos que calcule la diferencia entre dos vectores `A` y `B` y lo almacene en un tercero `C`:

A.3 Programando en CUDA

```
--global-- void f(int* A, int* B, int* C)
{
    int i = threadIdx.x;
    C[i] = A[i] - B[i];
}
```

El mismo ejemplo con matrices sería:

```
--global-- void f(int** A, int** B, int** C)
{
    int i = threadIdx.x; //Columna del bloque que ocupa
                        //este determinado hilo
    int j= threadIdx.y; //Fila
    C[i][j] = A[i][j] - B[i][j];
}
```

A.3.2. Invocación a un kernel

En una llamada a un kernel, se le ha de pasar el tamaño de la malla y del bloque, por ejemplo, en el programa del ejemplo anterior podríamos añadir:

```
dim3 bloque(N, N); //Definimos un bloque de hilos de NxN
dim3 grid(M, M); //Malla de dimension MxM

f<<<grid, bloque>>>(A, B, C);
```

En el momento que se invoque esta función, los bloques de una malla se enumerarán y distribuirán por los distintos multiprocesadores libres.

A.3.3. Gestión de la memoria

La memoria del GPU y la memoria del CPU son entidades distintas.

A.4 Thrust

- Punteros del GPU apuntan a la memoria del GPU.
 - Se puede pasar desde y hacia el código del CPU
 - No puede hacerse referencia a partir del código del CPU
- Punteros del CPU apuntan a la memoria del CPU.
 - Se puede pasar desde y hacia el código del GPU
 - No puede hacerse referencia a partir del código del GPU

Algunas funciones para acceder a la memoria del GPU es `cudaMalloc()`, `cudaFree()`, `cudaMemcpy()`, estas funciones trabajan de forma similar como sus equivalentes en C `malloc()`, `free()`, `memcpy()`.

A.3.4. Sincronización y memoria compartida

Cada bloque tiene una memoria compartida por todos los threads que se encuentran en ella a la cual se le conoce como memoria compartida, para poder declarar una variable en esta memoria y poder acceder a ella en el código del GPU basta con declarar la variable con la palabra `__shared__`, por ejemplo:

```
__shared__ int temp[N];
```

Si en cada thread del bloque asignamos un valor en la entrada respectiva de `temp` (en el ejemplo), y luego queremos sumarlos (tenemos que seguir en el bloque para poder hacerlo), antes tenemos que sincronizar los hilos para asegurarnos que todos los valores han sido guardados en `temp`, el comando para hacer esto es `__syncthreads()` con la que todos los hilos se esperan a que los demás lleguen a ese mismo punto.

A.4. Thrust

Thrust es una librería de plantillas en `C++` para CUDA basada en la librería de plantillas estándar (STL por sus siglas en inglés). Thrust permite implementar aplicaciones paralelas de alto rendimiento con un esfuerzo de programación mínima a través de una interfaz de

A.4 Thrust

alto nivel que es totalmente interoperable con CUDA C.

Thrust permite una colección rica de primitivas paralelas de datos tales como scan, sort, y reduce, las cuales pueden ser compuestas juntas para implementar algoritmos complejos con código fuente legible y conciso.

En esta pequeña sección mostraremos la función que usamos dentro de la librería Thrust, pero para una información más detallada se puede consultar la guía de iniciación rápida ².

A.4.1. Reducciones

Un algoritmo de reducción usa una operación binaria para reducir una secuencia de entrada a un solo valor. Por ejemplo, la suma de un arreglo de números es obtenido por reducción del arreglo con una operación suma. De manera similar, el máximo de un arreglo es obtenido reduciendo con un operador que toma dos entradas y regresa el máximo. La suma de un arreglo es implementada con `thrust::reduce` como sigue:

```
int sum = thrust::reduce (D.begin(), D.end(), (int) 0,  
thrust::plus<int>());
```

los primeros dos argumentos de `reduce` definen el rango de valores, mientras el tercero y el cuarto parámetro dan el valor inicial y el operador de reducción respectivamente. Actualmente, este tipo de reducción es tan común que es una elección por defecto cuando el valor inicial o el operador no es dado. Las siguientes líneas son equivalentes:

```
int sum = thrust::reduce (D.begin(), D.end(), (int) 0,  
thrust::plus<int>());  
int sum = thrust::reduce (D.begin(), D.end(), (int) 0);  
int sum = thrust::reduce (D.begin(), D.end());
```

²http://docs.nvidia.com/cuda/pdf/Thrust_Quick_Start_Guide.pdf

Bibliografía

- [1] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [2] R. Malladi, R. Kimmel, D. Adalsteinsson, G. Sapiro, V. Caselles, and J. A. Sethian, “A geometric approach to segmentation and analysis of 3D medical images,” in *Workshop on Mathematical Methods in Biomedical Image Analysis*, pp. 244–252, IEEE, 1996.
- [3] A. Yezzi Jr, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum, “A geometric snake model for segmentation of medical imagery,” *IEEE Transactions on Medical Imaging*, vol. 16, no. 2, pp. 199–209, 1997.
- [4] L. Jonasson, X. Bresson, P. Haggmann, O. Cuisenaire, R. Meuli, and J.-P. Thiran, “White matter fiber tract segmentation in DT-MRI using geometric flows,” *Medical Image Analysis*, vol. 9, no. 3, pp. 223–236, 2005.
- [5] A. Blake and A. Zisserman, *Visual Reconstruction*. MIT Press Cambridge, 1987.
- [6] V. Caselles, F. Catté, T. Coll, and F. Dibos, “A geometric model for active contours in image processing,” *Numerische Mathematik*, vol. 66, no. 1, pp. 1–31, 1993.
- [7] S. Osher and J. A. Sethian, “Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations,” *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.
- [8] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.
- [9] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi Jr, “Conformal curvature flows: from phase transitions to active vision,” *Archive for Rational Mechanics and Analysis*, vol. 134, no. 3, pp. 275–301, 1996.

BIBLIOGRAFÍA

- [10] T. F. Chan and L. A. Vese, “Active contours without edges,” *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [11] D. Mumford and J. Shah, “Optimal approximations by piecewise smooth functions and associated variational problems,” *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.
- [12] T. F. Chan, B. Y. Sandberg, and L. A. Vese, “Active contours without edges for vector-valued images,” *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, pp. 130–141, 2000.
- [13] Y. Zhai, B. Wu, D. Zhang, and J. Sun, “A new variational model for segmenting objects of interest from color images,” *Mathematical Problems in Engineering*, vol. 2010, 2010.
- [14] C. Li, C.-Y. Kao, J. C. Gore, and Z. Ding, “Minimization of region-scalable fitting energy for image segmentation,” *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1940–1949, 2008.
- [15] F. Li, C. Shen, and L. Pi, “A new diffusion-based variational model for image denoising and segmentation,” *Journal of Mathematical Imaging and Vision*, vol. 26, no. 1-2, pp. 115–125, 2006.
- [16] X. Bresson, S. Esedoğlu, P. Vandergheynst, J.-P. Thiran, and S. Osher, “Fast global minimization of the active contour/snake model,” *Journal of Mathematical Imaging and Vision*, vol. 28, no. 2, pp. 151–167, 2007.
- [17] Y. Yang, C. Li, C.-Y. Kao, and S. Osher, “Split bregman method for minimization of region-scalable fitting energy for image segmentation,” in *Advances in Visual Computing*, pp. 117–128, Springer, 2010.
- [18] J. Yuan, E. Ukwatta, X. Tai, A. Fenster, and C. Schnörr, “A fast global optimization-based approach to evolving contours with generic shape prior,” Tech. Rep. CAM 12-38, University of California Los Angeles, 2012.
- [19] V. V. Makkapati and R. M. Rao, “Segmentation of malaria parasites in peripheral blood smear images,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009*, pp. 1361–1364, 2009.

BIBLIOGRAFÍA

- [20] S. Mandal, A. Kumar, J. Chatterjee, M. Manjunatha, and A. Ray, “Segmentation of blood smear images using normalized cuts for detection of malarial parasites,” in *India Conference (INDICON), 2010 Annual IEEE*, pp. 1–4, 2010.
- [21] L. B. Damahe, R. K. Krishna, N. J. Janwe, and N. V. Thakur, “Segmentation based approach to detect parasites and rbcs in blood cell images,” *International Journal of Computer Science and Applications*, vol. 4, no. 2, pp. 71–81, 2011.
- [22] D. Freedman and T. Zhang, “Active contours for tracking distributions,” *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 518–526, 2004.
- [23] T. Zhang and D. Freedman, “Improving performance of distribution tracking through background mismatch,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 2, pp. 282–287, 2005.
- [24] R. L. Burden and J. D. Faires, *Análisis numérico*. International Thomson Editores, S. A. de C. V., séptima ed., 2002.
- [25] E. Parzen, “On estimation of a probability density function and mode,” *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.
- [26] A. Bhattacharyya, “On a measure of divergence between two statistical populations defined by their probability distributions,” *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943.
- [27] J. Sanders and E. Kandrot, *CUDA by example: an introduction to general-purpose GPU programming*. Addison–Wesley, 2010.
- [28] D. B. Kirk and W. H. Wen-meí, *Programming massively parallel processors: a hands-on approach*. Morgan Kaufmann, 2010.
- [29] R. Farber, *CUDA application design and development*. Applications of GPU computing, Elsevier, 2011.
- [30] S. Tse, L. Bradbury, J. W. Wan, H. Djambazian, R. Sladek, and T. Hudson, “A combined watershed and level set method for segmentation of brightfield cell images,” in *Society of Photo-Optical Instrumentation Engineers Medical Imaging*, vol. 7259, p. 72593G, International Society for Optics and Photonics, 2009.

BIBLIOGRAFÍA

- [31] R. Ali, M. Gooding, T. Szilágyi, B. Vojnovic, M. Christlieb, and M. Brady, “Automatic segmentation of adherent biological cell boundaries and nuclei from brightfield microscopy images,” *Machine Vision and Applications*, vol. 23, no. 4, pp. 607–621, 2012.
- [32] M. Tscherepanow, F. Zöllner, M. Hillebrand, and F. Kummert, “Automatic segmentation of unstained living cells in bright-field microscope images,” in *Advances in Mass Data Analysis of Images and Signals in Medicine, Biotechnology, Chemistry and Food Industry* (P. Perner and O. Salvetti, eds.), vol. 5108 of *Lecture Notes in Computer Science*, pp. 158–172, Springer, 2008.
- [33] C. J. Bradhurst, W. Boles, and Y. Xiao, “Segmentation of bone marrow stromal cells in phase contrast microscopy images,” in *Image and Vision Computing New Zealand, 2008. 23rd International Conference*, pp. 1–6, IEEE, 2008.
- [34] T. Riklin-Raviv, N. Kiryati, and N. Sochen, “Unlevel-sets: Geometry and prior-based segmentation,” in *Computer Vision - ECCV 2004* (T. Pajdla and J. Matas, eds.), vol. 3024 of *Lecture Notes in Computer Science*, pp. 50–61, Springer, 2004.
- [35] J. W. Demmel, *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics, 1997.
- [36] G. H. Golub and C. F. Van Loan, *Matrix Computations*. The Jonhs Hopkins University Press, third ed., 1996.
- [37] J. W. Thomas, *Numerical partial differential equations: finite difference methods*. No. 1, Springer, 1995.
- [38] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second ed., 1992.
- [39] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O’reilly, 2008.
- [40] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*. Academic Press, 2001.
- [41] G. Aubert and L. Vese, “A variational method in image recovery,” *SIAM Journal on Numerical Analysis*, vol. 34, no. 5, pp. 1948–1979, 1997.

BIBLIOGRAFÍA

- [42] L. C. Evans and R. F. Gariepy, *Measure theory and fine properties of functions*. CRC Press, 1992.
- [43] E. Rouy and A. Tourin, “A viscosity solutions approach to shape-from-shading,” *SIAM Journal on Numerical Analysis*, vol. 29, no. 3, pp. 867–884, 1992.
- [44] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.
- [45] M. Sussman, P. Smereka, and S. Osher, “A level set approach for computing solutions to incompressible two-phase flow,” *Journal of Computational physics*, vol. 114, no. 1, pp. 146–159, 1994.
- [46] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, “A variational level set approach to multiphase motion,” *Journal of computational physics*, vol. 127, no. 1, pp. 179–195, 1996.
- [47] S. Yang and M. K. Gobbert, “The optimal relaxation parameter for the SOR method applied to a classical model problem,” Tech. Rep. TR2007-6, Department of Mathematics and Statistics, University of Maryland, Baltimore County, 2007.
- [48] D. Marr, *Vision: A computational investigation into the human representation and processing of visual information*. W.H. Freeman and Company New York, 1982.
- [49] J. W. Thomas, *Numerical partial differential equations: conservation laws and elliptic Equations*. No. 2, Springer, 1999.
- [50] A. Blake and M. Isard, *Active Contours*. Springer, 1998.
- [51] S. I. Grossman, *Álgebra lineal*. Mc Graw Hill, sexta ed., 2008.