

Método de detección de contornos:
implementación dinámica del modelo de
Chan-Vese y detección de contornos basado en
multirresolución

Perfecto Vidal Lloret



**UNIVERSITAT
JAUME·I**

Tutores

Dr. Vicente Candela Pomares
Dr. Vicente Martínez García

Castellón, Julio 2014

Índice

1. El origen del tratamiento de imágenes	7
2. Segmentación de imágenes y detección de contornos	11
2.1. Métodos clásicos de detección de contornos	11
2.1.1. Métodos basados en el gradiente	11
2.1.2. Métodos basados en el operador Laplaciano	12
2.1.3. Problemas de los métodos clásicos basados en los operadores	12
2.2. Enfoque moderno	13
3. Modelo Mumford-Shah	15
3.1. Modelo de adquisición de imagen estándar	15
3.2. Enfoque Variacional para la segmentación de imágenes	15
3.3. Funcional Mumford-Shah	17
3.4. Análisis de la solución del funcional	18
3.5. Problema de partición mínima	18
3.6. Formulación del funcional de Mumford-Shah mediante conjuntos de nivel	19
3.7. Ecuaciones de Euler Lagrange	21
3.8. Implementación	23
4. Modelo de Lie-Lysaker-Tai “Piecewise Constant Level Set Methods and Image Segmentation”	27
4.1. Conceptos para la representación de subdominios	27
4.2. Minimización de L respecto a c	28
4.3. Minimización de L respecto a ϕ	29
4.4. Maximización de L respecto a λ	30
4.5. Método de aproximación polinómica	30
4.6. Algoritmo	32
5. Modelo Chan-Vese	35
5.1. Relación con el funcional de Mumford-Shah	37
5.2. Formulación del modelo mediante curvas de nivel	38
5.3. Ecuaciones de Euler Lagrange	40
5.4. Implementación	40
5.4.1. Algoritmo	41
6. Implementación dinámica del modelo de Chan-Vese y mapas de contorno	43
6.1. Modificaciones del modelo Chan-Vese	43
6.2. Discretización con el parámetro de escala	44
6.3. Criterio de parada	45
6.4. Método multiresolutivo y mapa de contornos	45

7. Resultados y conclusiones	47
7.1. Pruebas numéricas	47
7.2. Métodos de 2 capas	51
7.2.1. Imagen Geométrica	51
7.2.2. Imagen Geométrica con 8.46 dB de ruido	52
7.2.3. Imagen Geométrica con 16.55 dB de ruido	53
7.2.4. Brain	54
7.2.5. Peppers	55
7.2.6. Lena	56
7.2.7. Cameraman	57
7.2.8. Satélite	58
7.2.9. Conclusiones	59
7.3. Métodos multicapa	61
7.3.1. Imagen Geométrica	61
7.3.2. Imagen Geométrica con 8.46 dB de ruido	62
7.3.3. Imagen Geométrica con 16.55 dB de ruido	63
7.3.4. Brain	64
7.3.5. Peppers	65
7.3.6. Lena	66
7.3.7. Cameraman	67
7.3.8. Satélite	68
7.3.9. Extracción de información del Mapa de contornos	69
7.3.10. Conclusiones	70
7.4. Resumen	71
8. Trabajo futuro	73

La presente memoria forma parte de las materias a superar para optar al título de Máster Universitario en Matemática Computacional por la Universitat Jaume I. Ha sido realizada por Don Perfecto Vidal Lloret, bajo la dirección de los Doctores Vicente Candela Pomares y Vicente Martínez García, los cuales firman la presente diligencia para que así conste a los efectos oportunos en Castellón a 20 de junio de 2014.

Dr. Vicente Candela Pomares
Departament de Matemàtiques
Universitat de València

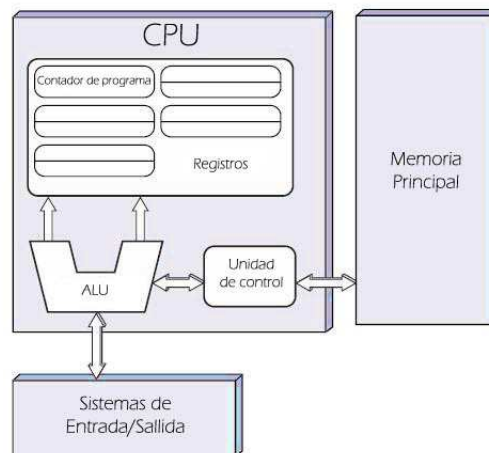
Dr. Vicente Martínez García
Departament de Matemàtiques
Universitat Jaume I

1. El origen del tratamiento de imágenes

El tratamiento de imágenes, tal y como lo conocemos hoy en día, apareció en 1920, a raíz de las necesidades en la industria periodística. Situándonos históricamente, en los años 1910, las empresas periodísticas tardaban más de una semana en enviar una imagen desde Londres a Nueva York. Por lo tanto se decidió que las imágenes fueran enviadas por cable submarino. Para esto, Harry G. Bartolomé y D. Maynard Mc Farlane inventaron el primer método de tratamiento digital, llamado *Bartlane cable picture transmission system*, que fue aplicado en 1921, consiguiendo transmitir una imagen a través de un cable, cruzando el Atlántico en tres horas. Este método tenía un gran inconveniente, la calidad con la que se recibían las imágenes, ya que sólo disponía cinco niveles distintos de gris. El método fue mejorado en 1929 permitiendo un incremento de niveles de gris, hasta un total de quince. También se consiguió una mejora en el proceso de reproducción de imágenes, con la introducción de un sistema para el desarrollo de una placa de película, a través de haces de luz moduladas por una cinta de imagen codificada.

Debido a la fuerte relación existente entre el tratamiento digital y el desarrollo de los ordenadores, ya que las grandes limitaciones en la capacidad de almacenamiento y el potencial de cálculo sesgo su evolución hasta el cambio de tecnología. Por este motivo es interesante hacer un pequeño repaso a los momentos claves en la historia de la computación, las cuales nos llevaron a conocer los ordenadores como lo son hoy en día.

Uno de los personajes más importantes fue el matemático húngaro John Von Neumann (1903-1957), el cual introdujo un concepto clave que revolucionó el desarrollo de los computadores: la arquitectura Von Neumann, que aún hoy en día utilizan prácticamente todos los procesadores, y que se ilustra en la siguiente figura.



A partir de las ideas de John Von Neumann, hubo una serie de importantes avances que llevaron a la potencia suficiente, para ser utilizados en el procesamiento de imágenes. Entre estos avances podemos destacar los siguientes:

1. La invención del transistor (Diciembre 1947 por John Bardeen, Walter Houser y William Brandford, en los laboratories Bell), que sustituyó a la válvula termoiónica de tres electrodos.
2. El desarrollo de los lenguajes de programación de alto nivel (COBOL [1959] y FORTRAN [1957]), permitiendo ser usados en cualquier ordenador y facilitando la creación de programas más complicados.
3. Circuitos integrados (1958, por Texas Instruments).
4. Sistemas Operativos (en la década de los 60).
5. El desarrollo de microprocesadores (1971 por Intel. Como curiosidad el primer microprocesador fue el 4004 que tenía 2300 transistores, con una potencia de 700 KHz y arquitectura de 4 bits).
6. La creación del Personal Computer (1981 IBM).
7. Miniaturización de componentes.

Paralelamente se mejoró la capacidad de almacenamiento de datos en los computadores, pudiendo almacenar imágenes cada vez con más calidad.

Finalmente, no se puede pasar por alto un descubrimiento muy importante, que tuvo lugar en el año 1961 con la tesis doctoral de Ivan E. Sutherland, quien demostró que se podía emplear un ordenador para diseños interactivos de trazado de dibujos, utilizando un sencillo método de representación gráfica mediante el uso de tubos de rayos catódicos y comandos auxiliares de control de entrada de datos. Esta idea no apareció de la nada: a principios de los años cincuenta, otros investigadores lograron conectar los tubos de rayos catódicos para generar sencillos *output displays*, pero hasta que Sutherland desarrolló su sistema de generación interactiva de imágenes hombre-máquina, no se pudo admirar el gran potencial que ofrecían las imágenes digitales.

Tras estos antecedentes, nos centramos ahora en el tratamiento de imágenes, tal y como lo conocemos en la actualidad. Podemos considerar que este apareció, fruto del inicio del programa espacial, en el trabajo sobre el uso de técnicas informáticas para la mejora de imágenes de la sonda espacial Ranger7 (Laboratorio espacial de propulsión a chorro [Pasadena, California 1964]).



Esta fue la primera imagen de la luna desde el espacio, gracias a la Ranger7.

El desarrollo obtenido en el campo del tratamiento de imágenes, a partir del trabajo con esta sonda, sirvió de base para el avance de los métodos utilizados para la recuperación y mejora de las imágenes, consiguiendo en los siguientes años, con las misiones espaciales, topografiar los planetas del sistema solar.

Otra importante vertiente que apareció durante la década de 1960 y principios de los 1970, es el tratamiento de imágenes médicas, creando un invento importante en el área de la medicina, la tomografía axial computarizada, o como más comúnmente se llama, TAC, que ha permitido un avance más que notable en la detección y diagnóstico de enfermedades. Esta tecnología consiste en un proceso en el que un anillo de detectores rodea un objeto (o paciente) y una fuente de rayos X (concéntricos con el anillo detector) gira alrededor del objeto. Los rayos X pasan a través del objeto y se recogen en el extremo opuesto. La tomografía consiste en utilizar los datos detectados para la construcción de una imagen que representa un “corte” a través del objeto. La novedad es que, al rotar, el algoritmo nos permite producir un conjunto de cortes tales que constituyen una versión tridimensional del interior del objeto.

Aparte de las aplicaciones en medicina y en el programa espacial, las nuevas técnicas de tratamiento de imágenes se han aplicado en muchas otras áreas. Por ejemplo, los procedimientos informáticos que se utilizan para realzar el contraste o los niveles de intensidad del color, facilitan la interpretación en las radiografías, en medicina y ciencias biológicas. También los geógrafos utilizan las mismas o similares técnicas para estudiar los patrones de contaminación, a partir de imágenes aéreas o de satélites. Otro campo importante es la utilización de la mejora de imágenes y restauración de estas; se utiliza para procesar imágenes que se han degradado con el tiempo, o resultados experimentales demasiado caros para volver a ser reproducidos. En el campo de la arqueología se usan métodos de tratamiento de imágenes que han restaurado con éxito imágenes borrosas, las cuales eran las únicas disponibles de raros artefactos perdidos o

dañados después de ser fotografiadas. En la física y campos afines, estas técnicas mejoran las imágenes de experimentos en áreas de microscopía electrónica. De manera parecida encontramos aplicaciones de gran éxito, en la astronomía, medicina nuclear y en la industria, entre otras.

Los ejemplos anteriores ilustran los resultados del tratamiento de imágenes destinado a la interpretación humana. Otro gran problema es el procesamiento de imágenes destinado a la percepción de máquinas. En este ámbito el interés se centra en la extracción de la información de la imagen, en un formato adecuado para el tratamiento informático. A menudo esta información tiene poco o ningún parecido con las características visuales de los humanos, que se utilizan para interpretar el contenido de una imagen. Algunos ejemplos de tipos de datos que se utilizan en la percepción de las máquinas son:

- Datos estadísticos.
- Coeficientes de la transformada de Fourier.
- Medidas multidimensionales de la distancia.

Los problemas típicos en la percepción de la máquina son las técnicas de reconocimiento de caracteres, procesamiento industrial para el montaje e inspección, reconocimiento militar, reconocimiento de huellas digitales, detección de rayos X y automatización del procesamiento de imágenes aéreas y de satélite, ya sea para la predicción del tiempo como para la evaluación ambiental.

Para concluir este pequeño resumen, digamos que la disminución de precios en los ordenadores y el aumento de rendimiento, ha facilitado el desarrollo de este campo; si, además le sumamos la creación de redes de comunicación (Internet) y la automatización en campos de reconocimiento, se han creado oportunidades sin precedentes para el continuo crecimiento del procesamiento digital de imágenes.

2. Segmentación de imágenes y detección de contornos

El tratamiento de imágenes matemático, es un tópico de investigación que alberga muchas especialidades. Entre ellas, este trabajo se basa en la segmentación de imágenes y la detección de contornos. Aunque estos dos conceptos puedan parecer independientes, realmente están fuertemente relacionados, debido que uno es considerado la frontera de los objetos y el otro el interior.

Este campo tiene una gran importancia tanto en aplicaciones teóricas, donde plantea nuevos enfoques teóricos, como desde el punto de vista aplicado, que nos permite afrontar una gran cantidad de problemas actuales.

Destacamos los siguientes:

- Realidad disminuida.
- Reconstrucción 3D mediante técnicas “Carving”.
- Detección de tumores en pruebas no invasivas (TAC, resonancia magnética, rayos X).
- Seguimiento y control del tráfico.

Todos estos problemas trabajan en imágenes bidimensionales digitales que definiremos matemáticamente de la siguiente manera:

Definición 2.1. Sea Ω un subconjunto cerrado y conexo de \mathbb{R}^2 , al que llamaremos **dominio de una imagen**, definiremos una **imagen** como una aplicación:

$$u_0 : \Omega \rightarrow \mathbb{R}$$

2.1. Métodos clásicos de detección de contornos

Nos centraremos en los métodos basados en el análisis numérico y no en aquellos que se basan en método estadísticos como puede ser el de histograma o métodos por *quadtrees* también conocidos como estructuras de árbol.

2.1.1. Métodos basados en el gradiente

Dada una imagen suave, sin ruido, se observa visualmente que los contornos se corresponden con las zonas de cambio, donde se produce una variación. Trasladando este concepto a las funciones reales unidimensionales, podemos identificar este concepto de contorno, como aquellos valores donde la derivada no existe o cambia de forma drástica. Por tanto podemos establecer el concepto de contornos de una imagen, que son aquellos lugares donde el módulo del gradiente supera un cierto umbral predefinido.

Definición 2.2. Sea u una imagen. Definimos su gradiente como:

$$\nabla u = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix}$$

Dado que una imagen digital es una discretización de la definición matemática, solamente conocemos con conjunto discreto de pares, por este motivo no podemos calcular el valor del gradiente como función continua, sino una aproximación discreta al mismo. Produciendo que diferentes discretizaciones dan lugar a distintos resultados.

La ventaja de estos métodos son su velocidad, ya que son muy simples desde el punto de vista computacional, pero tienen una gran cantidad de desventajas, son muy sensibles al ruido, dan muchos falsos positivos y falsos negativos.

2.1.2. Métodos basados en el operador Laplaciano

Estos tipos de métodos se basan en trabajar con el operador Laplaciano, que está definido de la siguiente manera:

Definición 2.3. Dada $f(x, y)$ una función suficientemente regular, definimos su Laplaciano como el operador:

$$\Delta f = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Este operador valdrá cero si la función es constante y el cambio de signo nos indicará que se anula en algún punto de ese intervalo, por tanto nos indicará la presencia de un supuesto perfil. Este operador es invariante por rotación, es decir, el Laplaciano y la rotación conmutan para cualquier imagen.

Al igual que ocurre con el operador gradiente, este operador es muy sensible al ruido. Por este motivo se suele aplicar un suavizado inicial para posteriormente aplicar la detección de contornos. Si el suavizado es mediante una función gaussiana, estaremos aplicando el operador LoG.

2.1.3. Problemas de los métodos clásicos basados en los operadores

Estos métodos tienen en común una serie de problemas que no los hace adecuados:

- Es necesario elegir un valor de corte y el orden de las diferentes discretizaciones.
- Generan falsos positivos y falsos negativos.
- Son muy sensibles al ruido.

2.2. Enfoque moderno

Como hemos visto hasta ahora, buscar directamente los contornos nos producen una serie de problemas que pueden ser solucionados relativamente. Cambiando el enfoque de la siguiente manera, en vez de buscar los conjuntos de píxeles de transición de un objeto a otro, directamente buscamos aquellos conjuntos de píxeles que forman un objeto. Este es el origen de la segmentación, buscar aquellos conjuntos de píxeles conexos que mantienen una fuerte relación que indica que todos ellos forman un mismo objeto. Obviamente la intersección entre estos conjuntos corresponden a los contornos. Vamos a definir matemáticamente este nuevo enfoque:

Definición 2.4. *Sea Ω el dominio de una imagen, definimos un objeto de la imagen como $\Omega_i \subset \Omega$ abierto, donde $u_0(\Omega_i)$ corresponde a todos los píxeles de este objeto.*

Por tanto podemos definir la segmentación de la imagen como una partición del dominio de la siguiente manera:

$$\Omega = (\cup_{i=1}^n \Omega_i) \cup \Gamma$$

Donde nuestros contornos buscados serán Γ .

En este trabajo presentaremos cuatro formas de encontrar el valor de Γ .

- Método de Mumford-Shah.
- Método de Lie-Lysaker-Tai.
- Método de Chan-Vese.
- Modificación dinámica del método de Chan-Vese.

3. Modelo Mumford-Shah

En el año 1989 los investigadores D. Mumford y J. Shah describieron uno de los funcionales más utilizados en la segmentación de imágenes. Empezamos describiendo e interpretando los problemas previos que dieron paso a la formulación de Mumford y Shah, para poder analizar después dicha formulación y los problemas que presenta su discretización, finalizando con el estudio del comportamiento de la solución para usarlo como herramienta en la segmentación de imágenes.

3.1. Modelo de adquisición de imagen estándar

El modelo de adquisición de imágenes estándar viene dado por:

$$g = Af + n \quad (1)$$

donde g es la imagen corrupta (la imagen que recibimos), f es la imagen original, n es el ruido aditivo, el cual suponemos que es aleatorio siguiendo una distribución uniforme, con un valor medio conocido, varianza σ^2 y A es un operador lineal que representa la influencia del sistema óptico (usualmente se considera A como un operador gaussiano).

Por tanto nuestro problema se basa en encontrar f , conociendo el valor de g y una estimación de A y σ^2 . Aunque aparentemente simple a primera vista, este tipo de problemas es bastante complicado ya que el operador A generalmente no va a ser invertible, su inversa no se podrá calcular con exactitud, o dicha inversa será inestable (y, por tanto, muy sensible a los errores). Así pues, una idea mejor, sería encontrar una aproximación de f cumpliendo:

$$\text{a) } \int_{\Omega} Af - g = 0.$$

$$\text{b) } \int_{\Omega} |Af - g|^2 = \sigma^2.$$

3.2. Enfoque Variacional para la segmentación de imágenes

Para poder estudiar el enfoque variacional de D.Mumford y J.Shah es necesario estudiar primero el enfoque estadístico propuesto por S.Geman y D.Geman, que trata de regularizar correctamente el problema inverso discreto y restaurar correctamente los bordes de una imagen. Este enfoque es reformulado por Mumford y Shah, efectuando una versión continua y aplicándolo al problema de la segmentación de imágenes.

La idea fundamental de S. Geman y D. Geman es considerar un modelo discreto basado en considerar la imagen observada, G como una matriz

$G = (g_{i,j})_{1 \leq i,j \leq n}$ con valores en tonos de gris entre 0 y 1, la imagen original f dada por $F = (f_{i,j})_{1 \leq i,j \leq n}$ y por último el ruido aditivo gaussiano $N = (n_{i,j})_{1 \leq i,j \leq n}$ donde cada $n_{i,j}$ son independientes y tienen una media nula y varianza de σ^2 .

Además fueron los que consideraron que la imagen es suave a trozos (en cada trozo no hay grandes variaciones en sus valores), con posibles discontinuidades, que llamaron bordes. Por esto introdujeron un conjunto de bordes¹ L , donde $L = \{(l_{i+\frac{1}{2},j})_{1 \leq i,j \leq n}, (l_{i,j+\frac{1}{2}})_{1 \leq i,j \leq n}\}$ toma valores en el conjunto $\{0, 1\}$ de la siguiente manera:

$$l_{i+\frac{1}{2},j} = \begin{cases} 1, & \text{si existe una discontinuidad entre } (i, j) \text{ y } (i+1, j), \\ 0, & \text{en otro caso.} \end{cases}$$

$$l_{i,j+\frac{1}{2}} = \begin{cases} 1, & \text{si existe una discontinuidad entre } (i, j) \text{ y } (i, j+1), \\ 0, & \text{en otro caso.} \end{cases}$$

Con todo lo anterior definieron una ley de probabilidad para F y L :

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

Donde los parámetros λ y μ son dos pesos positivos y Z es un coeficiente de normalización para que $\sum_{F,L} P(F, L) = 1$

Por tanto ahora nos encontramos con el problema de elegir, entre todas las posibles imágenes F y L existentes, aquellas que tienen la probabilidad más alta $P(F, L|G)$

$$P(F, L|G) \sim e^{-E(F,L,G)} \quad (3)$$

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

$$A_{i,j}(F, L) = \lambda((1 - l_{i+\frac{1}{2},j})(f_{i+1,j} - f_{i,j})^2 + (1 - l_{i,j+\frac{1}{2}})(f_{i,j+1} - f_{i,j})^2), \quad (4)$$

$$B_{i,j}(L, G) = \mu(l_{i+\frac{1}{2},j} + l_{i,j+\frac{1}{2}}) + \frac{1}{2\sigma^2}(g_{i,j}), \quad (5)$$

$$E(F, L, G) = \sum_{i,j} A_{i,j}(F, L) + B_{i,j}(L, G). \quad (6)$$

siendo G los datos conocidos. Por consiguiente, G puede obviarse quedando solo $E(F,L)$.

¹notese que estos bordes son horizontales y verticales

Así si minimizamos el funcional de la energía $E(G, L)$ lo que estaremos haciendo es maximizar $P(F, L|G)$ y como consecuencia obtendremos la imagen g del modelo de obtención de imágenes, que es constante a trozos, sabiendo que en aquellos lugares donde tiene cambios serán los contornos.

3.3. Funcional Mumford-Shah

A partir del modelo de la sección anterior D. Mumford y J. Shah decidieron reformular la ecuación (4) de manera continua. Para ello definieron Ω como un subconjunto abierto acotado de \mathbb{R}^2 (que es el dominio de la imagen) y la imagen como una función:

$$g : \Omega \longrightarrow [0, 1]$$

Siguiendo con la idea de Geman, crearon una partición del espacio Ω de la siguiente manera

$$\Omega = R \cup \Gamma$$

donde

$$R = \left(\bigcup_{i=1}^n R_i \right).$$

Consideraron Γ como el conjunto de posibles discontinuidades entre regiones de Ω denotados por R_i . Además, observaron que la variable L , en el enfoque de Geman, describe las discontinuidades o conjunto de saltos $\Gamma \subset \Omega$ de una función regular a trozos $f(x, y)$ donde $(x, y) \in \Omega$. Otra característica es que las diferencias entre $f_{i+1,j} - f_{i,j}$ y $f_{i,j+1} - f_{i,j}$ son aproximaciones de las derivadas parciales $\frac{\partial f}{\partial x}(x, y)$ y $\frac{\partial f}{\partial y}(x, y)$. De esta manera Mumford y Shah propusieron el siguiente funcional, que es una versión continua del modelo discreto de Geman:

$$E(f, \Gamma) = \nu \int_{\Omega \setminus \Gamma} (f - g)^2 + \lambda \int_{\Omega \setminus \Gamma} |\nabla f|^2 + \mu |\Gamma|, \quad (7)$$

donde Ω es el dominio, g es la imagen original en escala de grises, Γ denota el conjunto de bordes, f es la imagen suavizada pero discontinua a lo largo de Γ , $|\Gamma|$ es la longitud del conjunto de bordes y finalmente ν , α y μ son pesos.

Por tanto, la propuesta de Mumford-Shah radica en la minimización de este funcional, compuesto de tres términos que pueden interpretarse como:

- $\int_{\Omega \setminus \Gamma} (f - g)^2$, indica el nivel de fidelidad, es decir, controla cuánto se aproxima la imagen suave f a la imagen original g .
- $\int_{\Omega \setminus \Gamma} |\nabla f|^2$ será pequeño si f cambia lentamente en las regiones.
- $|\Gamma|$ controla el tamaño de la longitud del conjunto de bordes, que deberá ser pequeño para impedir que toda la imagen sea ocupada por ellos.

3.4. Análisis de la solución del funcional

Para simplificar el problema obviaremos en este apartado las variables ν , α y μ , cuya función es la de dar pesos a cada elemento del funcional, quedando de la siguiente manera:

$$E(f, \Gamma) = \int_{\Omega \setminus \Gamma} (f - g)^2 + \int_{\Omega \setminus \Gamma} |\nabla f|^2 + \int_{\Gamma} d\sigma. \quad (8)$$

Teorema 3.1. *Sea (f, Γ) una solución al problema (6) que satisface:*

- *Primera conjetura de Mumford-Shah (C_1)*
 Γ consiste en un número finito de curvas $C^{1,1}$, γ_i , que llegan a la frontera de Ω , es decir, parte de ellas están sobre $\partial\Omega$ y que, si intersectan a otras curvas, lo hacen sólo en puntos.
- *Segunda conjetura de Mumford-Shah*
 f es C^1 sobre cada componente conexa de $\Omega - \Gamma$.

Entonces:

- $\Delta f = f - g$ sobre Ω
- $\frac{\partial f}{\partial N} = 0$ sobre $\partial\Omega$ y a ambos lados γ_i^\pm de cada γ_i
- $e(f^+) - e(f^-) + \text{curv}\gamma_i = 0$ sobre γ_i .

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

Demostración. La demostración de este teorema la podemos encontrar en [1].

Con este teorema podemos observar que el funcional de Mumford-Shah involucra la medición de la curvatura como cantidad geométrica, hecho que nos permitirá más adelante representar este funcional por medio de conjuntos de nivel.

3.5. Problema de partición mínima

El funcional (5) propuesto por Mumford y Shah considera que f es una aproximación suave a trozos de la imagen dada g , es decir f varía suavemente dentro de las regiones de los objetos. Sin embargo Mumford y Shah también propusieron un caso límite del funcional (5):

$$E(f, \Gamma) = \nu \int_{\Omega - \Gamma} (f - g)^2 + \mu |\Gamma|, \quad (9)$$

donde f es una aproximación constante a trozos de g , es decir, $f = a_i$ constante en cada R_i .

A este caso tan restrictivo del funcional original, Mumford y Shah lo denominaron *problema de partición mínima*

3.6. Formulación del funcional de Mumford-Shah mediante conjuntos de nivel

Con el objetivo de facilitar el desarrollo supondremos que f es una función a trozos y constante, es decir que $f = c_i$ (constante) dentro de cada región R_i , donde $c_i = \text{media}(f)$ en las componentes conexas R_i , de $\Omega \setminus C$, siendo C la curva ∂R . Con todas estas consideraciones hemos obtenido un *problema de partición mínima*.

Dada una curva $C = \partial R$, le asociaremos una función de conjunto de nivel ϕ_i con cada fase R_i , donde entendemos fase como el conjunto de regiones con el mismo tono de gris. Entonces una fase R_i queda definida por:

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

y los bordes de cada fase son definidos por la unión de los conjuntos de nivel cero ϕ_i .

Dado que se pueden producir problemas de vacíos o superposiciones, es necesario establecer que los R_i son disjuntos y además que $\bigcup_i R_i = \Omega$; esto se logra si cumple la siguiente condición:

$$\sum_{i=1}^n H(\phi_i) = 1, \forall (x, y) \in \Omega, \quad (10)$$

donde H es la función de Heaviside unidimensional definida por:

$$H(z) = \begin{cases} 1, & \text{si } z \geq 0, \\ 0, & \text{en otro caso.} \end{cases}$$

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

$$\int_{\Omega} (\sum_{i=1}^n H(\phi_i) - 1) dx dy.$$

Este término nos puede ayudar a resolver el problema de transición de fases, para un número pequeño de regiones; sin embargo, esto no suele ocurrir normalmente, pues por lo general existen varias fases. Por lo tanto, para mejorarlo, vamos a considerar que para cualquier imagen sólo serán necesarios $\log(n)$ conjuntos de nivel, para representar n fases.

Sea $m = \log(n)$ el número de funciones de conjuntos de nivel $\phi_i : \Omega \rightarrow R$ necesaria para pintar n colores.

Sea $\Phi = (\phi_1, \dots, \phi_m)$ el vector de funciones de conjuntos de nivel y $H(\Phi) = (H(\phi_1), \dots, H(\phi_m))$ el vector de funciones de Heaviside, cuyas componentes son 0 ó 1. Además definiremos las fases del dominio Ω : dos píxeles (x_1, y_1) y (x_2, y_2) pueden pertenecer a la misma fase, si y sólo si $H(\Phi(x_1, y_1)) = H(\Phi(x_2, y_2))$, en

resumen, cada fase está determinada por la función de conjuntos de nivel $H(\phi)$. De esta manera, las fases forman una descomposición disjunta y a su vez, su unión provee el dominio completo.

Si etiquetamos cada fase con I , donde $1 \leq I \leq 2^m = n$, introducimos el concepto de vector constante de promedios $c = (c_1, c_2, \dots, c_n)$, donde cada $c_I = \text{media}(f)$ en la clase I ; y si consideramos χ_I la función característica del subconjunto I , el funcional puede escribirse como:

$$F_n(c, \Phi) = \sum_{1 \leq I \leq n} \nu \int_{\Omega} |f - c_I|^2 \chi_I dx dy + \frac{1}{2} \sum_I \mu \int_{\Omega} |\nabla \chi_I|, \quad (11)$$

donde el término longitud viene dado por:

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

Se puede simplificar como:

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

Por tanto la energía que minimizaremos viene dada por:

$$F_n(c, \Phi) = \sum_{1 \leq I \leq 2^m} \nu_I \int_{\Omega} |f - c_I|^2 \chi_I dx dy + \sum_i \mu_i \int_{\Omega} |\nabla H(\phi_i)|. \quad (12)$$

Obviamente, el conjunto de bordes C está representado por la unión de los conjuntos de nivel cero.

Si tomamos una imagen que contenga un solo objeto, existirán $n=2$ fases (color objeto y fondo), por tanto $m=1$, pero en cambio si consideramos una imagen con $n=4$ fases, entonces requeriríamos de $m=2$ funciones de nivel. Debido a la dificultad técnica de aumentar la cantidad de fases hemos decidido trabajar con sólo $m=2$ funciones de conjuntos de nivel, así podremos identificar cuatro fases análogas a los cuatro colores formados por funciones de nivel disjuntas $\{\phi_1 > 0, \phi_2 > 0\}, \{\phi_1 > 0, \phi_2 < 0\}, \{\phi_1 < 0, \phi_2 > 0\}$ y $\{\phi_1 < 0, \phi_2 < 0\}$. Además, consideraremos que los bordes son las regiones que forman la partición $\{\phi_1 = 0\} \cup \{\phi_2 = 0\}$, lo cual conforma el conjunto de contornos de C .

Llegados a este punto, podemos plantear el funcional (10), considerando dos funciones de conjuntos de nivel, para lograr cuatro fases:

$$F_4(C, \Phi) = \nu_1 \int_{\Omega} |g(x, y) - c_{1,1}|^2 H(\phi_1) H(\phi_2) dx dy \\ + \nu_2 \int_{\Omega} |g(x, y) - c_{1,0}|^2 H(\phi_1) (1 - H(\phi_2)) dx dy$$

$$\begin{aligned}
& +\nu_3 \int_{\Omega} |g(x, y) - c_{0,1}|^2 (1 - H(\phi_1)) H(\phi_2) dx dy \\
& +\nu_4 \int_{\Omega} |g(x, y) - c_{0,0}|^2 (1 - H(\phi_1)) (1 - H(\phi_2)) dx dy \\
& +\mu_1 \int_{\Omega} |\nabla H(\phi_1)| + \mu_2 \int_{\Omega} |\nabla H(\phi_2)|.
\end{aligned}$$

Donde $c = (c_{1,1}, c_{1,0}, c_{0,1}, c_{0,0})$ es un vector de constantes y $\Phi = (\phi_1, \phi_2)$.

Nota: Por simplicidad en nuestros cálculos teóricos, obviaremos los parámetros $\nu_1, \nu_2, \nu_3, \nu_4, \mu_1$ y μ_2 , para luego considerarlos en la solución final.

3.7. Ecuaciones de Euler Lagrange

Para obtener el mínimo de este funcional aplicaremos las Ecuaciones de Euler Lagrange que nos permitirán minimizarlo según va creciendo el tiempo.

Definición 3.1. Sea $H(z)$ la función de Heaviside, definimos la delta de Dirac como:

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

Teorema 3.2. Sea $\hat{F}(\phi) = \int_{\Omega} |\nabla H(\phi)|$, y $\phi = \phi(x, t)$, $x \in \mathbb{R}^2$, entonces la ecuación de Euler-Lagrange asociada es:

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \nabla \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \quad (13)$$

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

Demostración. La demostración de este teorema la podemos encontrar en [1].

Teorema 3.3. Sea:

$$\check{F}(\phi_1) = \int_{\Omega} |g(x, y) - c_{1,1}|^2 H(\phi_1) H(\phi_2) dx dy + \int_{\Omega} |g(x, y) - c_{1,0}|^2 H(\phi_1) (1 - H(\phi_2)) dx dy$$

$$+ \int_{\Omega} |g(x, y) - c_{0,1}|^2 (1 - H(\phi_1)) H(\phi_2) dx dy + \int_{\Omega} |g(x, y) - c_{0,0}|^2 (1 - H(\phi_1)) (1 - H(\phi_2)) dx dy$$

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

$$\frac{\partial \phi_1}{\partial t} = -\delta(\phi_1) \left[[(g - c_{1,1})^2 - (g - c_{0,1})^2] [H(\phi_2)] + [(g - c_{1,0})^2 - (g - c_{0,0})^2] [1 - H(\phi_2)] \right]$$

Demostración. La demostración de este teorema la podemos encontrar en [1].

Teorema 3.4. *Sea:*

$$\begin{aligned} \check{F}(\phi_2) &= \int_{\Omega} |g(x, y) - c_{1,1}|^2 H(\phi_1) H(\phi_2) dx dy + \int_{\Omega} |g(x, y) - c_{1,0}|^2 H(\phi_1) (1 - \\ &H(\phi_2)) dx dy \\ &+ \int_{\Omega} |g(x, y) - c_{0,1}|^2 (1 - H(\phi_1)) H(\phi_2) dx dy + \int_{\Omega} |g(x, y) - c_{0,0}|^2 (1 - H(\phi_1)) (1 - H(\phi_2)) dx dy \end{aligned}$$

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

$$\frac{\partial \phi_2}{\partial t} = -\delta(\phi_2) \left[[(g - c_{1,1})^2 - (g - c_{0,1})^2] [H(\phi_1)] + [(g - c_{1,0})^2 - (g - c_{0,0})^2] [1 - H(\phi_1)] \right]$$

Demostración. La demostración de este teorema la podemos encontrar en [1].

En las ecuaciones anteriores, los funcionales se derivaron en función de ϕ , manteniendo fijas las constantes $c_{1,1}, c_{1,0}, c_{0,1}, c_{0,0}$. Ahora mantenemos fija ϕ y minimizaremos la energía de nuestro funcional respecto a las constantes $c_{i,j}$.

Teorema 3.5. *Sea $g = g(x), x \in \mathbb{R}^2$, entonces:*

$$H(z) = \begin{cases} c_{1,1} = \text{media}(g) \text{ en } \{\phi_1 > 0, \phi_2 > 0\}, \\ c_{1,0} = \text{media}(g) \text{ en } \{\phi_1 > 0, \phi_2 < 0\}, \\ c_{0,1} = \text{media}(g) \text{ en } \{\phi_1 < 0, \phi_2 > 0\}, \\ c_{0,0} = \text{media}(g) \text{ en } \{\phi_1 < 0, \phi_2 < 0\}. \end{cases} \quad (14)$$

Demostración. La demostración de este teorema la podemos encontrar en [1].

Después de haber obtenido las ecuaciones de Euler Lagrange de los funcionales \hat{F}, \check{F} , vamos a unir los resultados para deducir las ecuaciones diferenciales que rigen el comportamiento de ϕ_1 y ϕ_2

La ecuación diferencial para ϕ_1 :

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

bajo las condiciones:

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

La ecuación diferencial para ϕ_2 :

$$\begin{aligned} \frac{\partial \phi_2}{\partial t} = & \delta(\phi_2) \left\{ \mu_2 \nabla \left(\frac{\nabla \phi_2}{|\nabla \phi_2|} \right) - [(\nu_1(g - c_{1,1})^2 - \nu_2(g - c_{1,0})^2)H(\phi_1) \right. \\ & \left. + (\nu_3(g - c_{0,1})^2 - \nu_4(g - c_{0,0})^2)(1 - H(\phi_1))] \right\} \text{ en } \Omega \times (0, \infty). \end{aligned}$$

bajo las condiciones:

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

Por tanto, podemos expresar la función que representa a la imagen f como:

$$\begin{aligned} f = & c_{1,1}H(\phi_1)H(\phi_2) + c_{1,0}H(\phi_1)(1 - H(\phi_2)) \\ & + c_{0,1}(1 - H(\phi_1))H(\phi_2) + c_{0,0}(1 - H(\phi_1))(1 - H(\phi_2)). \end{aligned} \quad (15)$$

3.8. Implementación

Después de todo lo anterior, refinaremos algunos aspectos antes de desarrollar el algoritmo de implementación.

En primer lugar, la función de Heaviside es una función discontinua, y como la función delta de Dirac se concibe como la derivada de Heaviside, ésta no está bien definida. Por tanto, para solucionar esto vamos a regularizarlas. La regularización de funciones es un campo amplio en sí mismo, pero a nivel ilustrativo, pondremos como ejemplos los que siguen:

Definición 3.2. *Función de Heaviside regularizada*

$$H_\varepsilon(z) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan \frac{z}{\varepsilon} \right). \quad (16)$$

Es fácil ver que H_ε converge a $H(Z)$ si ε tiende a 0.

Definición 3.3. *Función Delta de Dirac regularizada*

$$\delta_\varepsilon(z) = \frac{\varepsilon}{\pi} \left(\frac{1}{\varepsilon^2 + z^2} \right). \quad (17)$$

Ahora sólo nos falta discretizar las ecuaciones en ϕ_1 y ϕ_2 ; para ello utilizamos las diferencias finitas, definiendo h como el paso de espacio, Δt el paso del tiempo y (x_i, y_j) la malla de puntos, para $1 \leq i \leq N$ y $1 \leq j \leq M$, donde $N \times M$ son las dimensiones de la imagen. Además $\phi_{i,j}^n = \phi(x_i, y_j, n \Delta t)$ es una aproximación de $\phi(x, y, t)$, con $n \geq 0$, $\phi^0 = \phi_0$.

Por tanto, las diferencias finitas son:

$$\Delta_-^x \phi_{i,j} = \phi_{i,j} - \phi_{i-1,j}, \quad (18)$$

$$\Delta_+^x \phi_{i,j} = \phi_{i+1,j} - \phi_{i,j}, \quad (19)$$

$$\Delta_-^y \phi_{i,j} = \phi_{i,j} - \phi_{i,j-1}, \quad (20)$$

$$\Delta_+^y \phi_{i,j} = \phi_{i,j+1} - \phi_{i,j}. \quad (21)$$

Con todo esto, ya podemos discretizar las ecuaciones diferenciales, para facilitar la notación $\phi = \phi_1$ y $\varphi = \phi_2$.

$$\begin{aligned} \frac{\phi_{i,j}^{t+1} - \phi_{i,j}^t}{\Delta t} = & \delta_\varepsilon(\phi_{i,j}^t) \left[\frac{\mu_1}{h^2} \Delta_-^x \left(\frac{\Delta_+^x \phi_{i,j}^{t+1}}{\sqrt{\frac{(\Delta_+^x \phi_{i,j}^t)^2}{h^2} + \frac{(\phi_{i,j+1}^t - \phi_{i,j-1}^t)^2}{(2h)^2}}} \right) \right. \\ & + \frac{\mu_1}{h^2} \Delta_-^y \left(\frac{\Delta_+^y \phi_{i,j}^{t+1}}{\sqrt{\frac{(\phi_{i+1,j}^t - \phi_{i-1,j}^t)^2}{(2h)^2} + \frac{(\Delta_+^y \phi_{i,j}^t)^2}{h^2}}} \right) \\ & - [(\nu_1(g - c_{1,1}(\phi^t, \varphi^t))^2 - (\nu_3(g - c_{0,1}(\phi^t, \varphi^t))^2) H(\varphi^t) \\ & \left. + (\nu_2(g - c_{1,0}(\phi^t, \varphi^t))^2 - (\nu_4(g - c_{0,0}(\phi^t, \varphi^t))^2) (1 - H(\varphi^t)))] \right] \end{aligned}$$

$$\begin{aligned} \frac{\varphi_{i,j}^{t+1} - \varphi_{i,j}^t}{\Delta t} = & \delta_\varepsilon(\varphi_{i,j}^t) \left[\frac{\mu_2}{h^2} \Delta_-^x \left(\frac{\Delta_+^x \varphi_{i,j}^{t+1}}{\sqrt{\frac{(\Delta_+^x \varphi_{i,j}^t)^2}{h^2} + \frac{(\varphi_{i,j+1}^t - \varphi_{i,j-1}^t)^2}{(2h)^2}}} \right) \right. \\ & + \frac{\mu_2}{h^2} \Delta_-^y \left(\frac{\Delta_+^y \varphi_{i,j}^{t+1}}{\sqrt{\frac{(\varphi_{i+1,j}^t - \varphi_{i-1,j}^t)^2}{(2h)^2} + \frac{(\Delta_+^y \varphi_{i,j}^t)^2}{h^2}}} \right) \\ & - [(\nu_1(g - c_{1,1}(\phi^t, \varphi^t))^2 - (\nu_2(g - c_{1,0}(\phi^t, \varphi^t))^2) H(\phi^t) \\ & \left. + (\nu_3(g - c_{0,1}(\phi^t, \varphi^t))^2 - (\nu_4(g - c_{0,0}(\phi^t, \varphi^t))^2) (1 - H(\phi^t)))] \right]. \end{aligned}$$

Finalmente planteamos el algoritmo:

Entrada: Imagen g que queremos segmentar y el número máximo de iteraciones.

Salida: Imagen f segmentada.

- 1: Inicializamos ϕ_1^0 y ϕ_2^0
 - 2: $i = 0, \dots$ **hacer**
 - 3: Calcular todos los $c_{i,j}$
 - 4: Obtener ϕ_1^{i+1} y ϕ_2^{i+1}
 - 5: **fin hacer**
 - 6: Devolver f de la manera presentada en (15)
-

4. Modelo de Lie-Lysaker-Tai “Piecewise Constant Level Set Methods and Image Segmentation”

Motivados por la idea de Mumford-Shah, los autores del artículo solventaron el problema de la segmentación por un camino diferente. Introdujeron la función de conjuntos ϕ para zonas constantes, teniendo una única ϕ conseguiremos segmentar todas las regiones, es decir en lugar de usar la función de nivel cero para representar la división de subdominios. Estas estarán implícitamente representadas por las discontinuidades de un conjunto de funciones características ψ_i .

4.1. Conceptos para la representación de subdominios

Para cada Ω_i le asociaremos una función característica ψ_i de manera que

$$\psi_i(x, y) = \begin{cases} 1, & \text{si } x \in \Omega_i, \\ 0, & \text{en otro caso.} \end{cases}$$

más adelante estudiaremos la elección de este ψ_i .

Es obvio que los valores de ψ_i donde no se anula es una unión de conjuntos compactos, por tanto ψ_i tiene soporte compacto en Ω_i , y si además suponemos que el dominio de la imagen es la unión disjunta de Ω_i , podemos construir una función u constante en cada Ω_i , por lo tanto esta función u será la suma ponderada de funciones características.

Si tomamos $c = \{c_i\}_{i=1}^n$, conjunto de escalares, que ponderan las funciones características, tenemos que:

$$u = \sum_{i=1}^n c_i \psi_i. \quad (22)$$

Vamos a considerar el siguiente modelo de adquisición de imágenes, que viene dado por:

$$u_0 = u + \eta,$$

donde η es ruido gaussiano.

Estamos suponiendo que nuestra imagen de entrada va a ser una función u constante a trozos a la que se le ha añadido ruido gaussiano. Así, se puede aproximar esta imagen por (22) si se eligen de manera óptima los valores de c y las funciones características $\{\psi_i\}_{i=1}^n$; la elección de estos se hará mediante la resolución de un problema de minimización, sujeto a una restricción correspondiente a la elección de las funciones características, esta restricción será necesaria ya que controla la estructura de las posibles soluciones.

La estructura de las funciones características nos brindan la oportunidad de medir ([6]) la longitud de las curvas alrededor de cada Ω_i y el área de cada región Ω_i .

$$\text{Longitud} = |\partial\Omega_i| = \int_{\Omega} |\nabla\psi_i(x)| dx.$$

$$\text{Área} = |\Omega_i| = \int_{\Omega} \psi_i(x) dx.$$

notamos que cada $x \in \Omega$.

Volviendo al problema, tenemos una imagen u_0 que queremos segmentar, para conseguir u , constante a trozos y que aproxima a u_0 . Lo podemos formular como la minimización del siguiente funcional :

$$F(\psi, c) = \frac{1}{2} \int_{\Omega} |u(x) - u_0(x)|^2 dx + \beta \sum_{i=1}^n \int_{\Omega} |\nabla \psi_i(x)| dx \quad \beta > 0. \quad (23)$$

El primer término del funcional se acercará a cero, si u aproxima u_0 , por lo tanto al minimizar lograremos que u se parezca a u_0 , por otro lado el segundo término es un término regularizador que mide la curva que separa los subdominios, para que sea lo más pequeña posible.

Este funcional tiene un claro parecido con el funcional de Mumford-Shah ([14]), que fueron los pioneros de esta teoría.

Todavía no podemos afirmar que la minimización de este funcional en cada punto de la imagen sea única, debido a que las integrales pueden compensarse y tendremos varias formas posibles de llegar a ese mínimo. Para que sólo tenga una solución, los autores proponen introducir una restricción, que será $K(\phi) = 0$ y así tendremos que solucionar un problema de optimización con restricción

$$\min_{c, \psi} F(c, \psi) \text{ sujeto a } K(\phi) = 0.$$

Para solventar una minimización con restricción, proponen utilizar el método del Lagrangiano aumentado [5]; para ello consideraremos:

$$L(c, \phi, \lambda) = F(c, \psi) + \int_{\Omega} \lambda(x) K(\phi(x)) dx + \frac{r}{2} \int_{\Omega} |K(\phi(x))|^2 dx. \quad (24)$$

donde λ es una función llamada el multiplicador de lagrange con el mismo dominio que ϕ y $r \in \mathbb{R}^+$ es un parámetro de penalización para el método del Lagrangiano aumentado.

El método del lagrangiano aumentado nos indica que el mínimo de nuestro funcional está localizado en el punto de silla de la ecuación (24). En este punto:

$$\frac{\partial L}{\partial \phi} = 0, \quad \frac{\partial L}{\partial c_i} = 0, \quad \frac{\partial L}{\partial \lambda} = 0. \quad (25)$$

Esencialmente el algoritmo consistirá en minimizar L respecto c y ϕ y maximizaremos L respecto λ .

4.2. Minimización de L respecto a c

Para minimizar L respecto a c , calcularemos $\frac{\partial L}{\partial c_i} = 0$, ya que en este caso podremos hacerlo, como veremos a continuación:

$$0 = \frac{\partial L}{\partial c_i} = \frac{\partial F}{\partial c_i} = \frac{1}{2} \int_{\Omega} 2(u(x) - u_0(x))\psi_i(x)dx = \int_{\Omega} (u(x) - u_0(x))\psi_i(x)dx \quad i = 1, \dots, n,$$

así:

$$0 = \int_{\Omega} (u(x) - u_0(x))\psi_i(x)dx \rightarrow \int_{\Omega} u\psi_i = \int_{\Omega} u_0\psi_i.$$

Sustituimos u por su definición (22) quedando el siguiente sistema:

$$\sum_{j=1}^n \int (\psi_j(x)\psi_i(x))c_j = \int_{\Omega} u_0(x)\psi_i(x)dx. \quad (26)$$

Este sistema se puede escribir de manera lineal como:

$$\begin{bmatrix} \int_{\Omega} \psi_1\psi_1 & \int_{\Omega} \psi_2\psi_1 & \cdots & \int_{\Omega} \psi_{n-1}\psi_1 & \int_{\Omega} \psi_n\psi_1 \\ \int_{\Omega} \psi_1\psi_2 & \int_{\Omega} \psi_2\psi_2 & \cdots & \int_{\Omega} \psi_{n-1}\psi_2 & \int_{\Omega} \psi_n\psi_2 \\ \vdots & & & & \vdots \\ \int_{\Omega} \psi_1\psi_{n-1} & \int_{\Omega} \psi_{n-1}\psi_1 & \cdots & \int_{\Omega} \psi_{n-1}\psi_{n-1} & \int_{\Omega} \psi_n\psi_{n-1} \\ \int_{\Omega} \psi_1\psi_n & \int_{\Omega} \psi_2\psi_n & \cdots & \int_{\Omega} \psi_{n-1}\psi_n & \int_{\Omega} \psi_n\psi_n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} \int_{\Omega} u_0\psi_1 \\ \int_{\Omega} u_0\psi_2 \\ \vdots \\ \int_{\Omega} u_0\psi_{n-1} \\ \int_{\Omega} u_0\psi_n \end{bmatrix} \quad (27)$$

Por tanto ahora hemos liberalizado el problema a $Ac = b$, donde queremos saber la c ; este cambio nos asegura la existencia de solución, que sabemos calcularla de una manera numérica eficiente. Además si ψ_i esta bien definida y las particiones de Ω son disjuntas, esta matriz es diagonal.

4.3. Minimización de L respecto a ϕ

La minimización de L respecto a ϕ la calculamos usando el método iterativo del gradiente[7], el cual nos indica que la dirección más rápida para el descenso de L respecto ϕ es $-\frac{\partial L}{\partial \phi}$, por tanto queremos que $\frac{\partial \phi}{\partial t} = -\frac{\partial L}{\partial \phi}$, es decir, de manera discretizada, queremos que:

$$\frac{\phi^{new} - \phi^{old}}{\Delta t} = -\frac{\partial L}{\partial \phi}(c, \phi^{old}, \lambda),$$

así si lo haremos evolucionar de la siguiente manera para que minimice a L respecto ϕ :

$$\phi^{new} = \phi^{old} - \Delta t \frac{\partial L}{\partial \phi}(c, \phi^{old}, \lambda), \quad (28)$$

donde Δt es un número entero positivo pequeño, que estará determinado por el error de la malla.

4.4. Maximización de L respecto a λ

Para esto, el autor nos propone un método iterativo de primer orden, que depende de la restricción $K(\phi)$. El método es el siguiente:

$$\lambda^{new} = \lambda^{old} + rK(\phi). \quad (29)$$

Sustituyendo la parte de L relacionada con los multiplicadores de Lagrange, se maximiza L respecto λ

$$\int_{\Omega} \lambda^{new}(x)K(\phi(x)) = \int_{\Omega} \lambda^{old}(x)K(\phi(x))dx + r \int_{\Omega} K(\phi(x))^2 dx, \quad (30)$$

y una vez se cumpla la restricción ya no seguirá creciendo.

4.5. Método de aproximación polinómica

Una vez planteados los conceptos básicos y el desarrollo del modelo PCLSM, estudiaremos como llevarlo a cabo utilizando una aproximación polinómica.

Supongamos que necesitamos encontrar n regiones, en consecuencia tendremos $\{\Omega_i\}_{i=1}^n$ como partición de Ω . Con el fin de identificar cada región Ω_i hemos de encontrar una función constante a trozos, que tome los siguientes valores:

$$\phi(x) = i \quad \forall x \in \Omega_i,$$

con este enfoque sólo necesitaremos una función para identificar todas las zonas constantes, a las que se suele llamar fases.

Las funciones características de cada partición ψ_i asociadas con ϕ son definidas de la siguiente manera:

$$\psi_i(x) = \frac{1}{\alpha_i} \prod_{j=1, j \neq i}^n (\phi(x) - j), \quad \alpha_i = \prod_{k=1, k \neq i}^n (i - k). \quad (31)$$

Vamos a analizar esta función para ver si cumple nuestras expectativas. Tomando $x \in \Omega$, podemos suponer que $x \in \Omega_i \rightarrow \phi(x) = i$, veremos que valdrá $\psi_i(x)$ y $\psi_l(x)$ para $l \neq i$.

$$\begin{aligned} \psi_i(x) &= \frac{\prod_{j=1, j \neq i}^n (\phi(x) - j)}{\prod_{k=1, k \neq i}^n (i - k)} = \frac{\prod_{j=1, j \neq i}^n (i - j)}{\prod_{k=1, k \neq i}^n (i - k)} = 1, \\ \psi_l(x) &= \frac{\prod_{j=1, j \neq l}^n (\phi(x) - j)}{\prod_{k=1, k \neq l}^n (i - k)} = \frac{\prod_{j=1, j \neq l}^n (i - j)}{\prod_{k=1, k \neq l}^n (i - k)} = 0. \end{aligned}$$

Queda de la siguiente manera

$$\psi_i(x) = \begin{cases} 0, & x \notin \Omega_i, \\ 1, & x \in \Omega_i. \end{cases}$$

Cumple los requisitos del apartado dos, por tanto podemos decir que

$$u = \sum_{i=1}^n c_i \psi_i(x).$$

Además, tal como está planteado, tenemos que la función u es un polinomio de orden $n - 1$.

Para asegurarnos que la ecuación $u = \sum_{i=1}^n c_i \psi_i(x)$ nos da una única representación de u , introduciremos una función restricción:

$$K(\phi(x)) = \prod_{i=1}^n (\phi(x) - i). \quad (32)$$

Esto nos lleva a la conclusión que si elegimos funciones $\phi : \Omega \rightarrow \mathbb{R}$ que satisfacen $K(\phi) = 0$ tendremos una única $i \in \{1, 2, \dots, n\}$ para cada $x \in \Omega$ de manera que $\phi(x) = i$, es decir que cada punto sólo podrá encontrarse en un Ω_i si $K(\phi) = 0$. Además esta condición nos asegura que no pueden existir zonas que se solapen.

Una vez llegados a este punto ya podemos hacer uso de la construcción del apartado 2, usando las funciones básicas (31), la restricción (32) y la representación (22) de u .

Para minimizar el funcional (23) necesitaremos encontrar el punto de silla de (24), para ello el autor propone utilizar el algoritmo de Uzawa-type, que presentaremos a continuación, cuya convergencia esta estudiada en ([13]).

4.6. Algoritmo

Entrada: Imagen u_0 que queremos segmentar.

Salida: Imagen u segmentada.

- 1: Elegimos los valores iniciales ϕ^0 y λ^0
- 2: $k = 1, \dots$ **hacer**
- 3: Encontrar c^k de

$$L(c^k, \phi^{k-1}, \lambda^{k-1}) \quad (33)$$

{Calcular las variables c del paso k }

- 4: Actualizamos u mediante (22)
- 5: Encontrar ϕ^k de

$$L(c^k, \phi^k, \lambda^{k-1}) = \min_{\phi} L(c^k, \phi, \lambda^{k-1}) \quad (34)$$

- 6: Actualizamos u mediante (22)
- 7: Actualizar los multiplicadores de Lagrange

$$\lambda^k = \lambda^{k-1} + rK(\phi) \quad (35)$$

- 8: Si no converge, cogemos $k = k + 1$ y volver al paso 1
 - 9: **fin hacer**
-

Vamos a estudiar paso a paso como se llevaría a término este algoritmo, a partir de los apartados anteriores.

Para la elección de ϕ^0 el autor no da ningún criterio, aunque implícitamente en los ejemplos utiliza un método para la elección de esta. Sea n el número de fases y sea u_0 nuestra imagen a trabajar (sin pérdida de generalidad podemos suponer que todos los valores de u_0 son positivos):

$$\phi^0 = n * \frac{u_0}{\max(u_0)}.$$

Esta elección de ϕ^0 nos permite tener todos los valores entre $[0, n]$, facilitando la convergencia a los número enteros menores o igual que n .

La elección de los multiplicadores de Lagrange iniciales no es tan relevante, ya que irán creciendo dependiendo de como vaya cumpliendo ϕ^k la restricción impuesta.

Para calcular (33), podemos ir directamente al desarrollo del apartado 2.1, donde tenemos que existe solución exacta mediante (27). Hay que tener cuidado ya que puede estar mal condicionada; en ese caso se procederá a los métodos numéricos correspondientes para intentar paliar este problema. Tai y Yao ([18]) nos proponen una modificación de este paso, quedándonos solamente con la

diagonal de la matriz de (27), calculando c_i de la siguiente manera:

$$c_i = \frac{\int_{\Omega} \psi_i u_0}{\int_{\Omega} \psi_i^2}, \quad i = 1, 2, \dots, n. \quad (36)$$

Debido a la definición ψ_i , c_i convergerá a:

$$c_i = \frac{\int_{\Omega_i} u_0}{\text{Area}(\Omega_i)}, \quad i = 1, 2, \dots, n; \quad (37)$$

coincidiendo exactamente con la idea de que c_i tenga el valor de la media de u_0 en cada Ω_i . Esta idea no es única de este modelo, otros modelos como [2] también la utilizan, para calcular el valor en las zonas constantes.

En caso de tener muchos problemas con el mal condicionamiento de la matriz (27), es recomendable utilizar el método mencionado, ya que directamente eliminaremos este problema.

Para actualizar ϕ (5), como hemos dicho anteriormente usaremos el método del gradiente.

$$\phi^{new} = \phi^{old} - \Delta t \frac{\partial L}{\partial \phi}(c, \phi^{old}, \lambda).$$

Por motivos de convergencia Δt ha de ser menor que $\frac{1}{4}$ para que el método sea estable, por tanto cogemos $\Delta t = 0,2$.

Para calcular $\frac{\partial L}{\partial \phi}$ usaremos la regla de la cadena:

$$\frac{\partial L}{\partial \phi} = (u - u_0) \frac{\partial u}{\partial \phi} - \beta \sum_{i=1}^n \nabla \left(\frac{\nabla \psi_i}{|\nabla \psi_i|} \right) \frac{\partial \psi_i}{\partial \phi} + (\lambda + rK) \frac{\partial K}{\partial \phi}. \quad (38)$$

Para calcular todo estas sumas, vamos a ir por partes:

- $\nabla \left(\frac{\nabla \psi_i}{|\nabla \psi_i|} \right)$ es la curvatura local, que la podemos aproximar de la siguiente manera ²:

$$\nabla \left(\frac{\nabla \psi}{|\nabla \psi|} \right) = \left(\frac{\nabla \psi_x}{|\nabla \psi|} \right)_x + \left(\frac{\nabla \psi_y}{|\nabla \psi|} \right)_y$$

Lo discretizamos mediante diferencias divididas de la siguiente manera

$$\nabla \left(\frac{\nabla \psi}{|\nabla \psi|} \right)_{j,k} = \left[\frac{\left(\frac{\nabla \psi_x}{|\nabla \psi|} \right)_{j+1/2,k} - \left(\frac{\nabla \psi_x}{|\nabla \psi|} \right)_{j-1/2,k}}{\Delta x} \right] + \left[\frac{\left(\frac{\nabla \psi_y}{|\nabla \psi|} \right)_{j,k+1/2} - \left(\frac{\nabla \psi_y}{|\nabla \psi|} \right)_{j,k-1/2}}{\Delta y} \right]. \quad (39)$$

Y otra vez aplicando diferencias divididas tanto a la derecha, como hacia la izquierda y centrales, calcularemos los valores que faltan de la siguiente manera:

²para facilitar la notación $\psi = \psi_i$

$$\left(\frac{\nabla\psi_x}{|\nabla\psi|}\right)_{j+1/2,k} = \frac{(\psi_{j+1,k} - \psi_{j,k})/\Delta x}{\sqrt{[(\psi_{j+1,k} - \psi_{j,k})/\Delta x]^2 + \{1/2[(\psi_{j,k+1} - \psi_{j,k-1})/2\Delta y + (\psi_{j+1,k+1} - \psi_{j+1,k-1})/2\Delta y]\}^2}}. \quad (40)$$

$$\left(\frac{\nabla\psi_x}{|\nabla\psi|}\right)_{j-1/2,k} = \frac{(\psi_{j,k} - \psi_{j-1,k})/\Delta x}{\sqrt{[(\psi_{j,k} - \psi_{j-1,k})/\Delta x]^2 + \{1/2[(\psi_{j-1,k+1} - \psi_{j-1,k-1})/2\Delta y + (\psi_{j,k+1} - \psi_{j,k-1})/2\Delta y]\}^2}}. \quad (41)$$

$$\left(\frac{\nabla\psi_y}{|\nabla\psi|}\right)_{j,k+1/2} = \frac{(\psi_{j,k+1} - \psi_{j,k})/\Delta y}{\sqrt{\{1/2[(\psi_{j+1,k} - \psi_{j-1,k})/2\Delta x + (\psi_{j+1,k+1} - \psi_{j-1,k+1})/2\Delta x]\}^2 + [(\psi_{j,k+1} - \psi_{j,k})/\Delta y]^2}}. \quad (42)$$

$$\left(\frac{\nabla\psi_y}{|\nabla\psi|}\right)_{j,k-1/2} = \frac{(\psi_{j,k} - \psi_{j,k-1})/\Delta y}{\sqrt{\{1/2[(\psi_{j+1,k-1} - \psi_{j-1,k-1})/2\Delta x + (\psi_{j+1,k} - \psi_{j-1,k})/2\Delta x]\}^2 + [(\psi_{j,k} - \psi_{j,k-1})/\Delta y]^2}}. \quad (43)$$

(44)

- Cálculo $\frac{\partial K}{\partial \phi}$

$$\frac{\partial K}{\partial \phi} = \sum_{l=1}^n \prod_{i=1, i \neq l}^n (\phi - i). \quad (45)$$

- Cálculo $\frac{\partial \psi_i}{\partial \phi}$

$$\frac{\partial \psi_i}{\partial \phi} = \frac{1}{\alpha_i} \sum_{l=1}^n \prod_{\substack{j=1 \\ j \neq l \\ j \neq i}}^n (\phi - j). \quad (46)$$

- Cálculo $\frac{\partial u}{\partial \phi}$

$$\frac{\partial u}{\partial \phi} = \sum_{i=1}^n c_i \frac{\partial \psi_i}{\partial \phi}. \quad (47)$$

Con todo este apartado ya tenemos todos los ingredientes para poder programarlo.

Nota: Algunos coeficientes a la hora de programarlos se pueden agrupar, con el fin de ahorrar cálculos y hacerlo de una manera más óptima.

5. Modelo Chan-Vese

A partir de la idea del funcional de Mumford-Shah, Chan y Vese desarrollaron un funcional que al minimizarlo detectaría los contornos de las imágenes que denotaremos por u_0 .

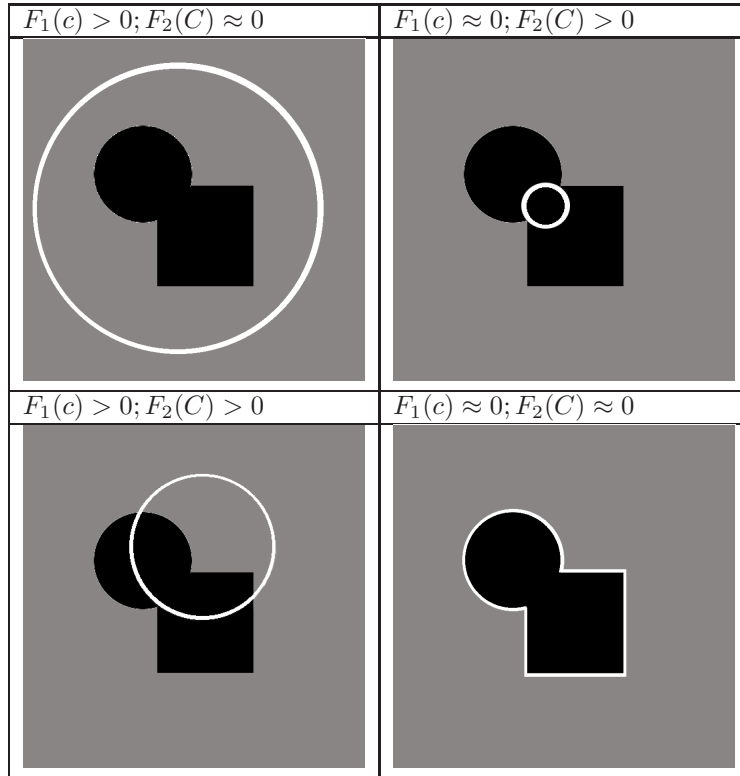
Definimos C como la curva inicial, que puede tomar cualquier forma excepto la nula y estar situada en cualquier zona de la imagen, ya que el resultado final será el mismo. Este detalle es muy positivo a la hora de trabajar ya que dependiendo del objeto que queramos segmentar podemos tomar una curva inicial genérica o algo más especializada, para que así converja con mayor velocidad. Tampoco es importante si la curva está dentro del objeto a segmentar, fuera o en ambas situaciones, ya que este método alcanzará el óptimo sin importar todos estos matices.

Consideraremos:

$$F_1(C) + F_2(C) = \int_{Dentro(C)} |u_0(x, y) - c_1|^2 dx dy + \int_{fuera(C)} |u_0(x, y) - c_2|^2 dx dy, \quad (48)$$

donde c_1, c_2 son constantes que vienen dadas por la media de la imagen dentro y fuera de la curva respectivamente.

Para simplificar la explicación, supondremos que el objeto es uniforme y el fondo también. Por tanto, si la curva está fuera del objeto, $F_1(c) > 0$ y $F_2(C) \approx 0$; del mismo modo si la curva está dentro del objeto, $F_1(c) \approx 0$ pero $F_2(C) > 0$; y por último si la curva C está dentro y fuera del objeto $F_1(c) > 0$ y $F_2(C) > 0$. Como podemos ver en las siguientes imágenes el contorno se detectará si minimizamos $F_1(C) + F_2(C)$.



Como hemos visto, este funcional nos permite obtener en casos simples una buena segmentación, pero no tenemos control de la curva, por este motivo fue modificado de tal manera que podamos tener un mayor control de la longitud y el área de la curva. Para ello añadiremos los parámetros de la longitud y el área, quedando de la siguiente manera:

$$\begin{aligned}
 F(c_1, c_2, C) &= \mu \cdot \text{long}(C) + \nu \cdot \text{Área}(\text{dentro}(C)) + \lambda_1 F_1(c_1, C) + \lambda_2 F_2(c_2, C); \\
 F_1(c_1, C) &= \int_{\text{Dentro}(C)} |u_0(x, y) - c_1|^2 dx dy, \\
 F_2(c_2, C) &= \int_{\text{fuera}(C)} |u_0(x, y) - c_2|^2 dx dy.
 \end{aligned}
 \tag{49}$$

donde $\mu, \nu \geq 0$ y $\lambda_1, \lambda_2 > 0$ son parámetros prefijados elegidos al tomar la imagen.

Nuestro problema consiste en calcular el ínfimo de F respecto a (c_1, c_2, C) .

5.1. Relación con el funcional de Mumford-Shah

Si escribimos el funcional de Mumford-Shah con la nomenclatura anterior tenemos que:

$$F^{MS}(u, C) = \mu \cdot \text{long}(C) + \lambda \int_{\Omega} |u_0(x, y) - u(x, y)|^2 dx dy + \int_{\Omega \setminus C} |\nabla u(x, y)| dx dy, \quad (50)$$

donde u_0 es la imagen dada, λ y μ parámetros positivos y u es la solución obtenida por minimización de este funcional. Por tanto el funcional de Chan-Vese es una generalización del funcional de Mumford-Shah, ya que si tomamos en el funcional de Chan-Vese $\nu = 0$, $\lambda_1 = \lambda_2 = \lambda$ obtenemos el de Mumford-Shah.

Además notamos que es un caso particular del problema de partición mínima, en el cual vemos que la mejor aproximación de u a partir de u_0 es la siguiente función:

$$u(x, y) = \begin{cases} \text{promedio}(u_0), & \text{dentro de } C, \\ \text{promedio}(u_0), & \text{fuera de } C. \end{cases}$$

Este caso particular del problema de partición mínima lo tenemos formulado y solucionado usando conjuntos de nivel. Para más información sobre este problema, remitimos a [15].

5.2. Formulación del modelo mediante curvas de nivel

Consideremos $C = \{(x, y) \in \Omega : \phi(x, y) = 0\}$, donde $\phi : \Omega \rightarrow \mathbb{R}$ es una función de Lipschitz. Por tanto:

$$Dentro(C) = \omega = \{(x, y) \in \Omega : \phi(x, y) > 0\},$$

$$Fuera(C) = \Omega \setminus \bar{\omega} = \{(x, y) \in \Omega : \phi(x, y) < 0\},$$

siendo $\omega \subset \Omega$ abierto y por tanto $C = \partial\omega$.

Para reescribir el funcional mediante curvas de nivel, vamos a usar, al igual que con Mumford-Shah, la función de Heavyside

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

y también la función delta

$$\delta_0(z) = \frac{d}{dz}H(z).$$

Con todo esto podemos empezar a reformular los términos del funcional:

$$long(\phi = 0) = \int_{\Omega} |\nabla H(\phi(x, y))| dx dy = \int_{\Omega} \delta_0(\phi(x, y)) |\nabla \phi(x, y)| dx dy, \quad (51)$$

$$\text{Área}(\phi \geq 0) = \int_{\Omega} H(\phi(x, y)) dx dy, \quad (52)$$

$$\int_{\phi > 0} |u_0(x, y) - c_1|^2 dx dy = \int_{\Omega} |u_0(x, y) - c_1|^2 H(\phi(x, y)) dx dy, \quad (53)$$

$$\int_{\phi < 0} |u_0(x, y) - c_2|^2 dx dy = \int_{\Omega} |u_0(x, y) - c_2|^2 (1 - H(\phi(x, y))) dx dy. \quad (54)$$

El funcional queda de la siguiente manera:

$$\begin{aligned}
F(c_1, c_2, \phi) = & \mu \int_{\Omega} \delta_0(\phi(x, y)) |\nabla \phi(x, y)| dx dy + \nu \int_{\Omega} H(\phi(x, y)) dx dy \\
& + \lambda_1 \int_{\Omega} |u_0(x, y) - c_1|^2 H(\phi(x, y)) dx dy \\
& + \lambda_2 \int_{\Omega} |u_0(x, y) - c_2|^2 (1 - H(\phi(x, y))) dx dy.
\end{aligned}$$

Como observamos antes, u está definida de la siguiente manera:

$$u(x, y) = \begin{cases} \text{promedio}(u_0), & \text{dentro de } C, \\ \text{promedio}(u_0), & \text{fuera de } C. \end{cases}$$

Por consiguiente, es un caso particular del problema de partición mínima y en conclusión

$$u(x, y) = c_1 H(\phi(x, y)) + c_2 (1 - H(\phi(x, y))),$$

donde los valores de c_1 y c_2 en función de ϕ , son:

$$c_1(\phi) = \frac{\int_{\Omega} u_0(x, y) H(\phi(x, y)) dx dy}{\int_{\Omega} H(\phi(x, y)) dx dy}, \quad (55)$$

$$c_2(\phi) = \frac{\int_{\Omega} u_0(x, y) (1 - H(\phi(x, y))) dx dy}{\int_{\Omega} (1 - H(\phi(x, y))) dx dy}. \quad (56)$$

Una vez formulado mediante conjuntos de niveles, esperamos la existencia del mínimo de la energía de $F(c_1, c_2, C)$. Esto se cumple, ya que nuestro modelo es un caso particular del problema de partición mínima, para el cual la existencia está demostrada ([14]), suponiendo que u_0 es continuo en Ω . Para u_0 más general está demostrado en [9][10].

5.3. Ecuaciones de Euler Lagrange

Con el fin de encontrar el mínimo utilizaremos las ecuaciones de Euler Lagrange de la función desconocida ϕ , para poder aplicar estas ecuaciones consideraremos la versión regularizada de la función de Heavyside y delta regularizada, que denotaremos por H_ϵ y δ_ϵ , que cumplen:

$$\lim_{\epsilon \rightarrow 0} H_\epsilon = H,$$

$$\lim_{\epsilon \rightarrow 0} \delta_\epsilon = \delta_0.$$

En conclusión, fijando c_1 y c_2 , reduciremos $F_\epsilon(c_1, c_2, C)$ respecto a ϕ . Si parametrizamos en dirección de descenso para un tiempo artificial $t \geq 0$, la ecuación es $\phi(t, x, y)$ (siendo $\phi_0(x, y) = \phi(0, x, y)$ que define el contorno inicial), se deduce la siguiente ecuación de Euler Lagrange:

$$\begin{cases} \frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) [\mu \cdot \operatorname{div}(\frac{\nabla \phi}{|\nabla \phi|}) - \nu - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2] = 0 & \text{en } (0, \infty) \times \Omega, \\ \phi(0, x, y) = \phi_0(x, y) & \text{en } \Omega, \\ \frac{\delta_\epsilon(\phi)}{|\nabla \phi|} \frac{\partial \phi}{\partial \vec{n}} = 0 & \text{en } \partial \Omega. \end{cases} \quad (57)$$

donde \vec{n} denota la normal exterior de la frontera $\partial \Omega$, y $\frac{\partial \phi}{\partial \vec{n}}$ es la derivada normal de ϕ respecto a la frontera.

5.4. Implementación

Con el fin de poder calcular numéricamente tenemos que discretizarlo. Para ello usaremos exactamente el mismo procedimiento que en Mumford-Shah utilizando diferencias divididas y obteniendo como resultado:

$$\begin{aligned} \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta t} = & \delta_h(\phi_{i,j}^n) \left[\frac{\mu}{h^2} \cdot \Delta_x \left(\frac{\Delta_+^x \phi_{i,j}^{n+1}}{\sqrt{\frac{(\Delta_+^x \phi_{i,j}^n)^2}{h^2} + \frac{(\phi_{i,j+1}^n - \phi_{i,j-1}^n)^2}{(2h)^2}}} \right) \right. \\ & + \frac{\mu}{h^2} \cdot \Delta_y \left(\frac{\Delta_+^y \phi_{i,j}^{n+1}}{\sqrt{\frac{(\Delta_+^y \phi_{i,j}^n)^2}{h^2} + \frac{(\phi_{i+1,j}^n - \phi_{i-1,j}^n)^2}{(2h)^2}}} \right) - \nu \\ & \left. - \lambda_1(u_{0_{i,j}} - c_1(\phi^n))^2 + \lambda_2(u_{0_{i,j}} - c_2(\phi^n))^2 \right]. \end{aligned} \quad (58)$$

5.4.1. Algoritmo

Entrada: Imagen u_0 que queremos segmentar, curva inicial ϕ_0 y número máximo de iteraciones n .

Salida: ϕ

- 1: $k = 1 : n$ **hacer**
 - 2: Calcular c_1 y c_2 respecto ϕ_k
 - 3: Actualizamos ϕ_{k+1} mediante (58).
 - 4: **fin hacer**
-

6. Implementación dinámica del modelo de Chan-Vese y mapas de contorno

Los modelos anteriormente presentados son modelos estáticos, en el sentido que las constantes se ajustan a partir de las condiciones iniciales, estos modelos pueden discriminar ciertos contornos que al minimizar el funcional no se consideran importantes. Además todos estos modelos presentan una serie de problemas que hemos intentado resolver con el modelo que introducimos:

- Dificultad a la hora de elegir que parámetro modificar para obtener un buen resultado.
- Complejidad numérica a la hora de aumentar el número de curvas.
- Velocidad de computacional.

El modelo desarrollado se basa en introducir un parámetro dinámico de escala con el objeto de conseguir distintas regularizaciones del funcional a minimizar, que servirán para detectar contornos de diferente regularidad. Este parámetro simplifica la elección todos los parámetros para centrarnos en una única variable, obteniendo los diferentes contornos con diferente grado de regularidad. Finalmente nos permitirá definir y construir un mapa de contornos, que será una versión multiresolutoria del modelo de Chan-Vese, solucionando así la complejidad numérica, ya que no necesitaremos aumentar el número de curvas y mediante una implementación con resolvedores rápidos tipo Fourier, conseguimos aumentar la velocidad en el cómputo.

6.1. Modificaciones del modelo Chan-Vese

Como hemos visto en el apartado anterior, tenemos la ecuación de Euler-Lagrange asociada al funcional:

$$\left\{ \begin{array}{l} \frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) [\mu \cdot \operatorname{div}(\frac{\nabla \phi}{|\nabla \phi|}) - \nu - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2] = 0 \text{ en } (0, \infty) \times \Omega, \\ \phi(0, x, y) = \phi_0(x, y) \text{ en } \Omega, \\ \frac{\delta_\epsilon(\phi)}{\nabla \phi} \frac{\partial \phi}{\partial \bar{n}} = 0 \text{ en } \partial \Omega. \end{array} \right. \quad (59)$$

Nuestro primer cambio de enfoque, se basa en cambiar la resolución del modelo, pasando de una resolución local, como son las diferencias divididas, a una global, como es la transformada discreta de Fourier, quedando de la siguiente manera:

$$\phi^{n+1} = \phi^n + \Delta t \delta_h(\phi^n) \left[\frac{\mu}{h^2} \cdot A - \nu - \lambda_1(u_0 - c_1(\phi^n))^2 + \lambda_2(u_0 - c_2(\phi^n))^2 \right], \quad (60)$$

donde A es un discretización de $div\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$ mediante la transformada de fourier:

$$A = F_x\left(\frac{F_x(\phi)}{\sqrt{F_x^2(\phi) + F_y^2(\phi)}}\right) + F_y\left(\frac{F_y(\phi)}{\sqrt{F_x^2(\phi) + F_y^2(\phi)}}\right), \quad (61)$$

siendo $F_x(\phi)$ la derivada de ϕ respecto la variable x , que calculamos mediante el uso de transformada de Fourier; de la misma manera respecto a la variable y .

De esta forma reutilizamos varios cálculos de manera muy simple y al ser un método global, al contrario que los locales, cuanto mayor tamaño tiene la imagen, más rápido se comportará el algoritmo,relativamente, en comparación a las diferencias divididas, obteniendo, por consiguiente, un mejor rendimiento. También hemos de destacar que la transformada de Fourier tiende a regularizar demasiado, pero esto lo contrarrestaremos en la siguiente modificación que nos permitirá modificar la escala de precisión.

6.2. Discretización con el parámetro de escala

Una vez mejorado la velocidad de computo mediante resolvedores rápidos de Fourier, vamos a cambiar la discretización propiamente dicha. Esta no afectaría a la convergencia del modelo, ya que el funcional no ha sido modificado y ,por lo tanto, sigue siendo convergente, la diferencia es que dependiendo del valor del parámetro escala, nos devolverá un tipo de contornos u otros.

Para esta modificación tomaremos la función gaussiana como una aproximación de la delta de Dirac:

$$\delta(\phi) \approx G_\beta(\phi) = \frac{1}{\sqrt{\pi}}e^{-\left(\frac{\phi}{\beta}\right)^2}. \quad (62)$$

Este cambio permite obtener aproximaciones de los contornos según el parámetro β , que lo consideramos un parámetro de escala. Aunque no parezca sustancial a simple vista, en el apartado conclusiones comprobaremos la importancia de estas modificaciones. Además de dinamizar la implementación, ya no es fundamental definir el límite de la aproximación a la δ dado que cada β nos aportará información relevante que utilizaremos posteriormente para obtener el mapa de contornos.

Aplicando todos estos cambios el funcional queda de la siguiente manera:

$$\begin{aligned} \phi^{n+1} = & \phi^n + \Delta t \cdot G_\beta(\phi^n) \left[\frac{\mu}{h^2} \cdot F_x\left(\frac{F_x(\phi)}{\sqrt{F_x^2(\phi) + F_y^2(\phi)}}\right) + F_y\left(\frac{F_y(\phi)}{\sqrt{F_x^2(\phi) + F_y^2(\phi)}}\right) \right. \\ & \left. - \nu - \lambda_1(u_0 - c_1(\phi^n))^2 + \lambda_2(u_0 - c_2(\phi^n))^2 \right]. \end{aligned} \quad (63)$$

6.3. Criterio de parada

Todo algoritmo debe tener un criterio de parada, que depende de la tolerancia al error, fijado inicialmente. A continuación describimos el criterio propuesto en el algoritmos nuestro.

Podemos suponer que nos hallamos en el paso I , obtenemos la curva del paso I , que llamaremos C_{new} , a partir de la curva del paso $I - 1$, la cual a partir de ahora nombraremos como C_{old} . Denotaremos M como el número de píxeles totales de la imagen que estamos tratando.

$$Q = \frac{\left\| \sum_i |C_{new}(i, :) - C_{old}(i, :)| \right\|_2}{M}. \quad (64)$$

En resumen sumamos en cada fila, todos los elementos de la diferencia, en valor absoluto, entre la curva vieja y la nueva, calculamos la norma 2 y para normalizar lo dividimos entre el número de píxeles, obteniendo un valor real que llamaremos Q . Cuando el valor de Q sea menor que la tolerancia que elijamos, pararemos el algoritmo. Por tanto, consideraremos la convergencia en la norma euclidea por filas, podríamos haber considerado por columnas o global, el resultado sería equivalente.

Este criterio de parada se cumple, debido a la convergencia del modelo. Esto significa que como vaya creciendo el número de iteraciones, la curva irá convergiendo al contorno de la imagen; así, a partir de cierta iteración la diferencia entre C_{old} y C_{new} se aproximará a 0.

6.4. Método multiresolutivo y mapa de contornos

Como su propio nombre indica, la multiresolución consiste en estudiar un problema, en diferentes escalas o resoluciones, para encontrar la solución. Esta forma de buscar la solución del problema es un enfoque totalmente diferente, ya que en vez de atacarlo directamente, se busca información en diferentes escalas, consiguiendo aportar cada una de ellas información no redundante, con el fin de facilitar la solución del problema a partir de estas. En la actualidad hay muchos métodos multiresolutivos, desde los wavelets hasta los basados en Fourier, pasando por técnicas interpolatorias a la Harten, entre otras. Gracias a la dinámica de nuestra implementación, conseguimos información dependiendo el factor escala, que utilizaremos para poder unir todas las imagenes en una obteniendo un mapa de contornos

Definición 6.1. *Dada una imagen a segmentar, tomamos el conjunto $C = \{C_{\beta_1}, \dots, C_{\beta_n}\}$, donde cada elemento C_{β_i} es el resultado de detectar los contornos utilizando el parámetro de escala β_i . Definimos el mapa de contornos del conjunto C , como:*

$$S = \sum_{i=1}^n \frac{\|C_{\beta_i}\|_2}{\|\sum_{j=1}^n C_{\beta_j}\|_2} C_{\beta_i}.$$

Es decir, el mapa de contornos consiste en una suma ponderada de todos los contornos obtenidos.

En la siguiente sección veremos que aquellos contornos que se repiten muchas veces, serán los contornos más fuertes y aquellos que no se repiten mucho, serán débiles, pero no los habremos perdido. Además, también veremos que en pocas iteraciones podremos obtener suficiente información, ya que la información se acumula con el parámetro escala, y esto nos permite reducir el coste computacional (en imágenes complejas será más rápido evaluar varias veces con pocas iteraciones, que muchas iteraciones en una evaluación). Este método, además tiene una ventaja añadida que se ajusta a los trabajos en matemática computacional actuales, es totalmente paralelizable mediante hebras de computación, permitiendo lanzar en cada hebra una evaluación, y así reduciríamos drásticamente el cómputo.

7. Resultados y conclusiones

Hasta ahora hemos desarrollado diferentes modelos de segmentación basadas en curvas activas, empezando por el modelo de Mumford-Shah, que luego fue modificado por Chan y Vese, el cual nos hemos basado para crear nuestra modificación y por último hemos desarrollado otro modelo creado por Lie, Lysaker y Tai, donde tienen una idea inicial muy parecida, pero cambian el enfoque totalmente.

Dado que el modelo de Mumford-Shah y Chan-Vese son muy parecido, hemos realizado la diferencia de montar el primero para dos curvas (que fácilmente es interpolable a múltiples) y el segundo solamente centrado con una curva de nivel, por tanto con el de Mumford-Shah nos dará como resultado cuatro capas de segmentación y Chan-Vese solamente dos.

7.1. Pruebas numéricas

Para las pruebas numéricas hemos elegido un conjunto de imágenes bastante amplio, para poder así comparar los métodos en las diferentes situaciones. La elección de estas imágenes viene caracterizada por estos detalles:

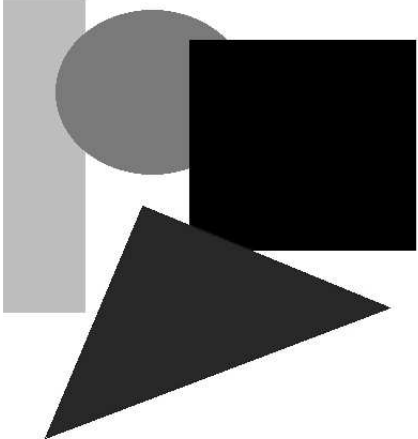
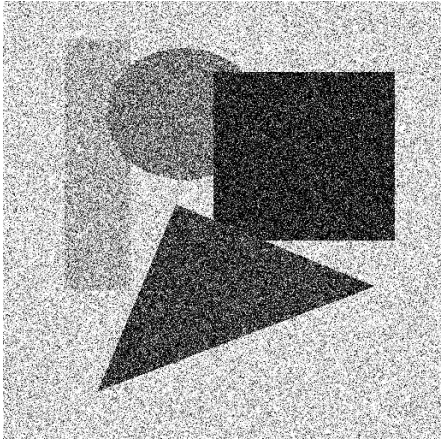
- **Imágenes Geométricas:** La importancia de estas imágenes viene caracterizadas por su simplicidad, ya que son constante a trozos, se conoce la solución, sus perfiles son rectas o curvas muy suaves y tienen esquinas bien definidas. Además veremos como se comportan los diferentes algoritmos con ruido.
- **Brain:** Es una imagen médica de poca resolución y muy poca textura.
- **Peppers:** Una imagen muy difícil de segmentar debido a que sus perfiles están poco definidos y su contraste es bajo.
- **Lena:** Una de las imágenes más recurrentes en el tratamiento de imágenes, debido a sus propiedades de textura, ya que tenemos zonas planas y zonas de mucha textura, tiene diferentes niveles de profundidad, que como más lejos están los objetos mayor es el desenfoque.
- **Cameraman:** Al contrario que la foto anterior, también tiene diferentes niveles de profundidad pero sin desenfoque. Su fondo esta poco realzado y tiene pocos niveles de grises, salvo el personaje principal.
- **Satélite:** Una imagen donde la importancia reside en que tiene un fondo negro homogéneo y sobre el hay una serie de detalles muy importantes como son las antenas o las placas interiores.

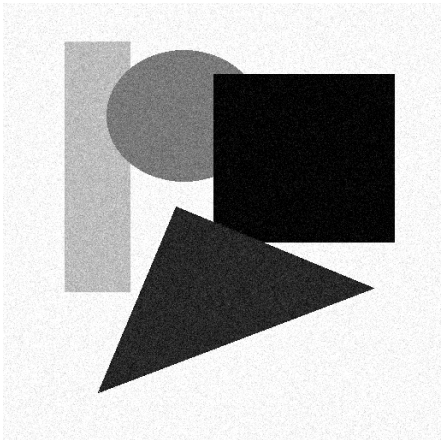

Nota técnica:

Todos los experimentos numéricos han sido desarrollado íntegramente en un portátil Satellite L850-138 con las siguientes características técnicas:

- S.O.: Ubuntu 14.04.4 LTS.
- Kernel Linux: 3.2.0-64-Generíc.
- Ram: 4 Gb.
- Cpu: i5-2450 2.50 GHz, 4 cores de computación.

Los algoritmos han sido implementados en Matlab y se ha realizado una tarea de optimización para que los tiempos de cómputo sean comparables de manera adecuada.

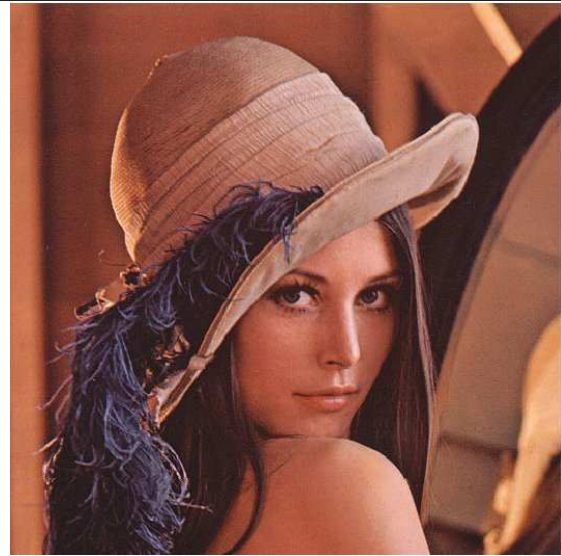
Geométrica	Geométrica Ruido:8.46 dB
	

Geométrica Ruido:16.55 dB	Brain
	

Peppers



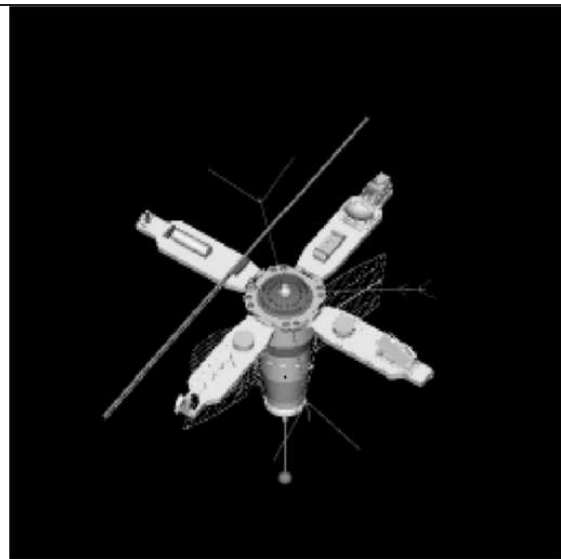
Lena



Cameraman



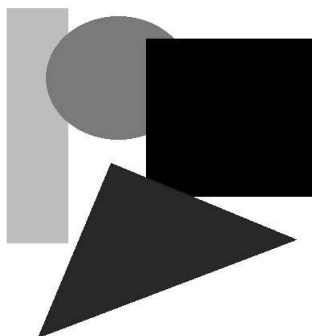
Satélite


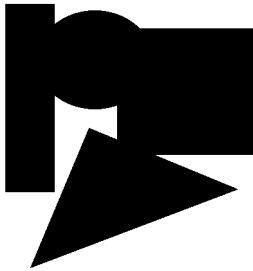
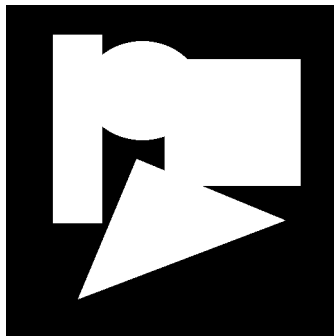


7.2. Métodos de 2 capas

Procedemos a analizar estas imágenes utilizando los modelos de manera que segmente en dos capas.

7.2.1. Imagen Geométrica

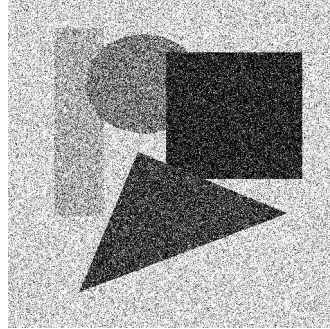


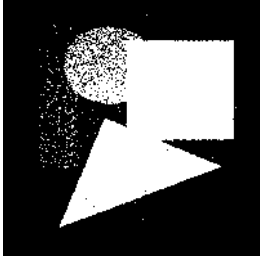
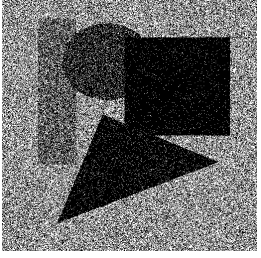
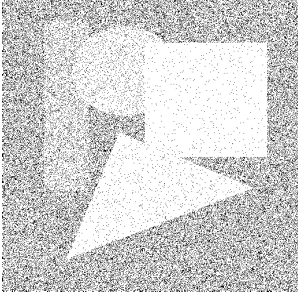
Chan-Vese	Lie-Lysaker-Tai	Método dinámico $\beta = 1$
		

Método	iteraciones	Tiempo medio proceso (s)	Coste por iteración (it/s)	
Chan-Vese	154	28.6500329	0.1860391753	
Lie-Lysaker-Tai	15	8.1298901	0.5419926733	
Método dinámico	2	0.4584991	0.22924955	$\beta = 1$

tabla 1: Tiempos y costes por iteración, tolerancia=0.0162 y tamaño de la imagen 512×512

7.2.2. Imagen Geométrica con 8.46 dB de ruido

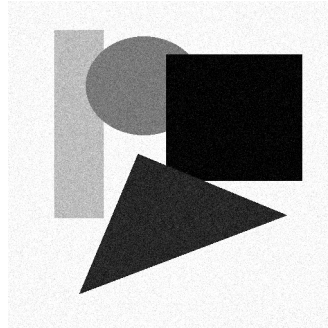


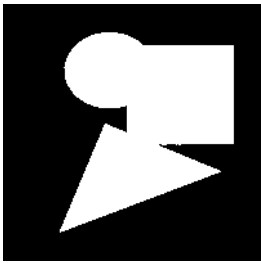
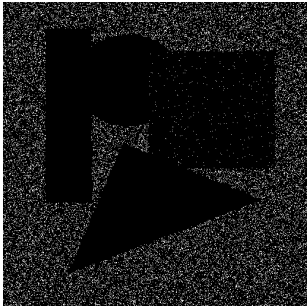
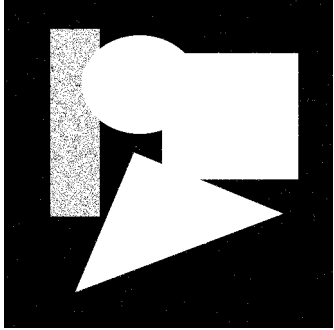
Chan-Vese	Lie-Lysaker-Tai	Método dinámico $\beta = 100$
		

Método	iteraciones	Tiempo medio proceso (s)	Coste por iteración (it/s)	
Chan-Vese	130	27.299683	0.2099975654	
Lie-Lysaker-Tai	50	27.598522	0.55197044	
Método dinámico	142	32.798503	0.2309753732	$\beta = 100$

tabla 2: Tiempos y costes por iteración, tolerancia=0.0162 y tamaño de la imagen 512× 512

7.2.3. Imagen Geométrica con 16.55 dB de ruido

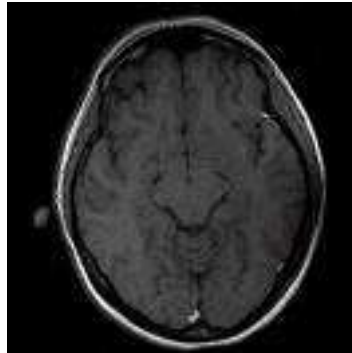





Chan-Vese	Lie-Lysaker-Tai	Método dinámico $\beta = 10$
		

Método	iteraciones	Tiempo medio proceso (s)	Coste por iteración (it/s)	
Chan-Vese	162	29.8219485	0.1840861019	
Lie-Lysaker-Tai	50	27.321732	0.54643464	
Método dinámico	4	0.9293281	0.232332025	$\beta = 10$

tabla 3: Tiempos y costes por iteración, tolerancia=0.0162 y tamaño de la imagen 512× 512

7.2.4. Brain



Chan-Vese	Lie-Lysaker-Tai	Método dinámico $\beta = 10$
		

Método	iteraciones	Tiempo medio proceso (s)	Coste por iteración (it/s)	
Chan-Vese	166	1.90049669	0.0114487753	
Lie-Lysaker-Tai	500	19.412786	0.038825572	
Método dinámico	6	0.1996024	0.0332670667	$\beta = 10$

tabla 4: Tiempos y costes por iteración, tolerancia=0.0162 y tamaño de la imagen 131×131

7.2.5. Peppers

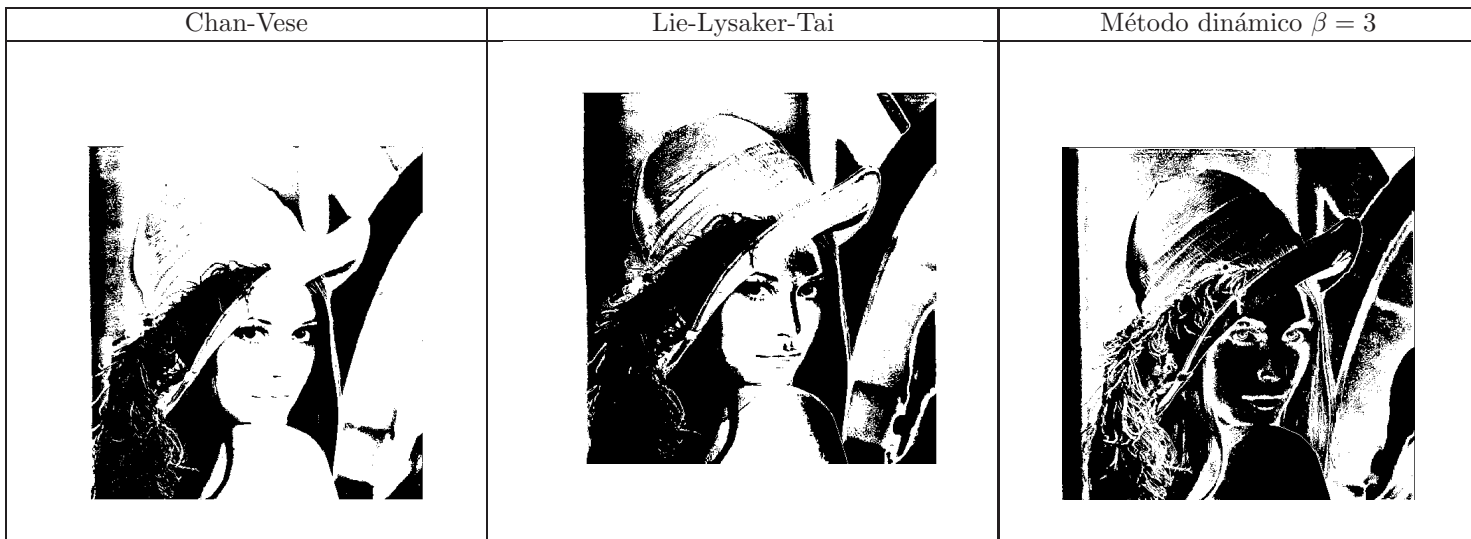


Chan-Vese	Lie-Lysaker-Tai	Método dinámico $\beta = 1$

Método	iteraciones	Tiempo medio proceso (s)	Coste por iteración (it/s)	
Chan-Vese	181	4.2258184	0.0233470633	
Lie-Lysaker-Tai	500	57.21711	0.11443422	
Método dinámico	3	0.2309237	0.0769745667	$\beta = 1$

tabla 5: Tiempos y costes por iteración, tolerancia=0.0162 y tamaño de la imagen 225× 225

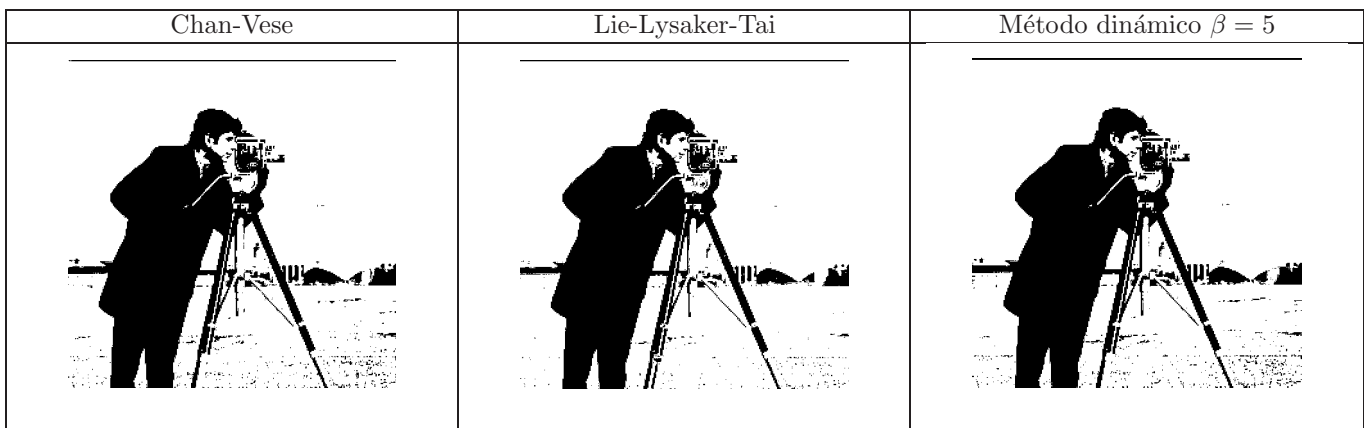
7.2.6. Lena



Método	iteraciones	Tiempo medio proceso (s)	Coste por iteración (it/s)	
Chan-Vese	368	79.9383794	0.2172238573	
Lie-Lysaker-Tai	100	65.313818	0.65313818	
Método dinámico	21	5.0884784	0.2423084952	$\beta = 3$

tabla 6: Tiempos y costes por iteración, tolerancia=0.0162 y tamaño de la imagen 532x 532

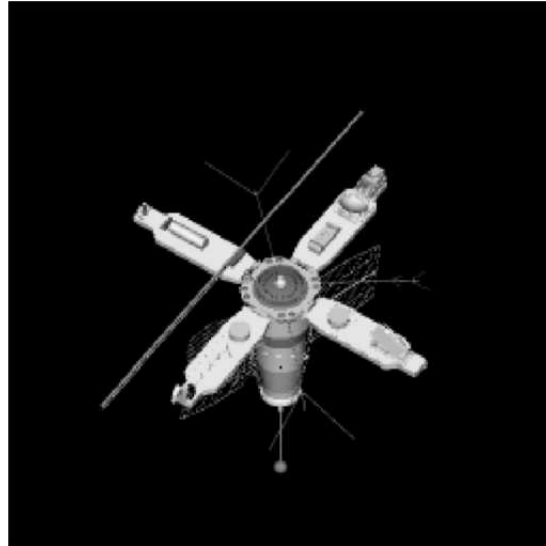
7.2.7. Cameraman

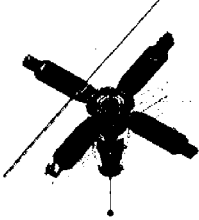
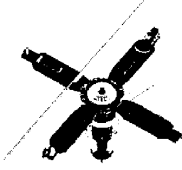



Método	iteraciones	Tiempo medio proceso (s)	Coste por iteración (it/s)	
Chan-Vese	169	36.7108318	0.2172238573	
Lie-Lysaker-Tai	100	119.254097	1.19254097	
Método dinámico	6	1.3761665	0.2293610833	$\beta = 5$

tabla 7: Tiempos y costes por iteración, tolerancia=0.0162 y tamaño de la imagen 512× 512

7.2.8. Satélite



Chan-Vese	Lie-Lysaker-Tai	Método dinámico $\beta = 1$
		

Método	iteraciones	Tiempo medio proceso (s)	Coste por iteración (it/s)	
Chan-Vese	475	83.1243015	0.1749985295	
Lie-Lysaker-Tai	64	34.5174319	0.5393348734	
Método dinámico	3	0.6982251	0.2327417	$\beta = 1$

tabla 8: Tiempos y costes por iteración, tolerancia=0.0162 y tamaño de la imagen 512× 512

7.2.9. Conclusiones

Si procedemos al análisis de todas las imágenes anteriores, encontramos una serie de detalles comunes que son interesantes de remarcar:

- El coste por iteración de Lie-Lysaker-Tai supera el doble del coste de los otros dos métodos.
- El de menor coste por iteración es el de Chan-Vese.
- El menor tiempo computacional total es el del método dinámico ya que converge más rápidamente.

Entrando en los detalles imagen a imagen vemos que:

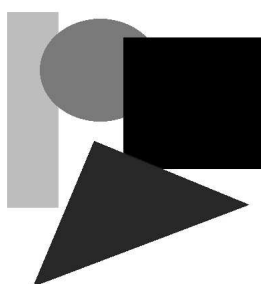
- Geométrica: Debido a que el rectángulo de color gris claro, está más cerca del color del fondo que del color del resto de objetos, produce que en el modelo de Chan-Vese no aparezca, dejando incompleta la segmentación en comparación con el resto de métodos.
- Geométrica 8.46 dB: El método menos sensible a este tipo de ruido no muy denso pero fuerte, es el método de Chan-Vese, ya que el resto de métodos cogen el ruido como parte de la segmentación.
- Geométrica 16.55 dB: A diferencia del ruido de la imagen anterior, este ruido es más denso, por tanto hace que el método pueda gestionarlo casi perfectamente.
- Brain: Esta imagen nos da que el método de Lie-Lysaker-Tai es más sensible a la textura y por tanto dando buenos detalles en la zona interior del cerebro. El método dinámico iguala la calidad en la parte exterior del cerebro que lo separa del hueso, mejorando ligeramente en la zona derecha, que por el difuminado el resto de métodos no la segmentan bien, en cambio en la zona izquierda no termina de salir del todo correctamente. En conclusión Lie-Lysaker-Tai es mejor en esta imagen, en cambio su convergencia es muy lenta.
- Peppers: Debido a la poca diferencia de tonos de grises, y que los perfiles colorimétricos no están bien definidos el resultado es más o menos parecido, con Chan-Vese y con su modificación dinámica. Lie-Lysaker-Tai tiene problemas con los reflejos.
- Lena: Esta imagen denota que cada método es bueno para ciertas cosas. El modelo de Chan-Vese saca muy bien lo que va siendo la zona del hombro y los ojos, pero considera muchas partes que son la misma produciendo una no segmentación de la misma. El método de Lie-Lysaker-Tai segmenta muy bien la zona del sombrero denotando cierta textura. Por último nuestra modificación con este parámetro β segmenta con bastante calidad los pequeños detalles de las plumas y los ojos, y también algunas zonas grandes como es el hombro.


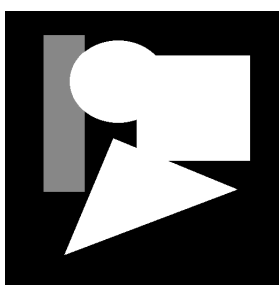
- Cameraman: En este caso los tres métodos sacan resultados muy parecidos.
- Satélite: Por último, en esta imagen vemos, que el resultado de Chan-Vese esta entre el Lie-Lysaker-Tai, que detecta algunas partes interiores y el modelo dinámico, con el que obtenemos todas las antenas exteriores. Aquí al poder elegir el parámetro de escala nos permite obtener aquellos detalles que son muy complicados de obtener.

7.3. Métodos multicapa

Una vez visto y comparado los métodos expuestos de una sola capa, vamos a comparar el método de varias capas, para ello lo que hacemos es comparar Mumford-Shah de cuatro capas con nuestro mapa de contornos. Decidimos dejar fuera el método Lie-Lysaker-Tai, que sobre tener una buena calidad, su coste por iteración junto a su velocidad de convergencia, impide que sea un método comparable a estos dos. Hemos decidido no añadir con Mumford-Shah más de 4 colores, ya que al añadir una curva nueva, el tiempo y dificultad de las ecuaciones crece de manera desmesurada.

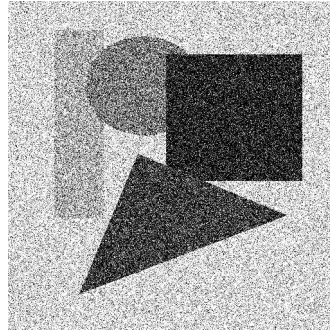
7.3.1. Imagen Geométrica



Mumford-Shah	Mapa de contornos
	

Modelo	Coste total
Mumford-shah	76.0592841989
Mapa de contornos	5.881118

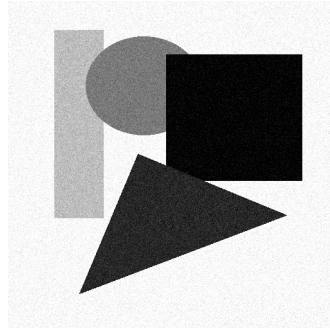
7.3.2. Imagen Geométrica con 8.46 dB de ruido



Mumford-Shah	Mapa de contornos

Modelo	Coste total
Mumford-shah	67.716790232
Mapa de contornos	13.296193

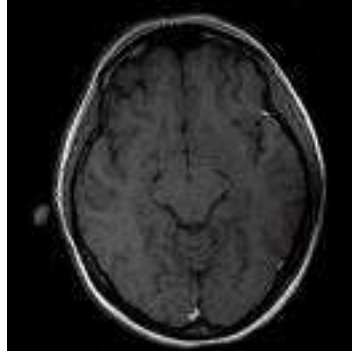
7.3.3. Imagen Geométrica con 16.55 dB de ruido





Mumford-Shah	Mapa de contornos
The original geometric image with the noise removed, showing the shapes in their original gray and black tones.	A binary image where the shapes are white and the background is black. The shapes are represented as solid white regions, with some internal noise still visible.

Modelo	Coste total
Mumford-shah	67.716790232
Mapa de contornos	13.296193

7.3.4. Brain





Mumford-Shah	Mapa de contornos
	

Modelo	Coste total
Mumford-shah	5.3707713193
Mapa de contornos	3.552349

7.3.5. Peppers




Mumford-Shah	Mapa de contornos
	

Modelo	Coste total
Mumford-shah	28.0845594383
Mapa de contornos	44.107412

7.3.6. Lena





Mumford-Shah	Mapa de contornos
	

Modelo	Coste total
Mumford-shah	112.2767495618
Mapa de contornos	63.787732

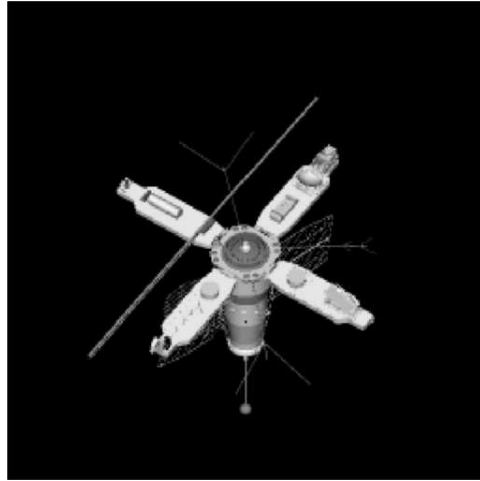
7.3.7. Cameraman

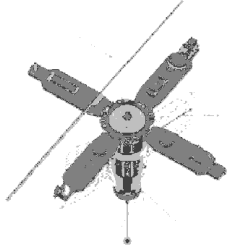
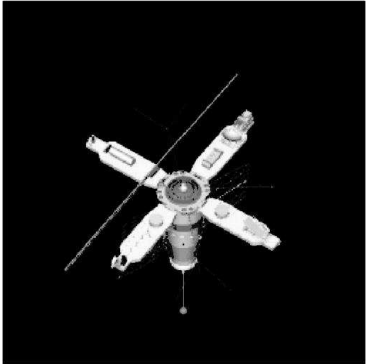


Mumford-Shah	Mapa de contornos
	

Modelo	Coste total
Mumford-shah	22.6125208354
Mapa de contornos	44.172587

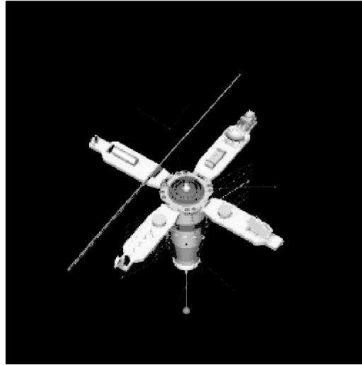
7.3.8. Satélite



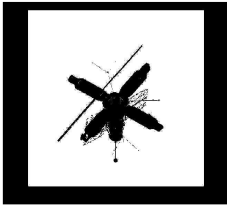
Mumford-Shah	Mapa de contornos
	

Modelo	Coste total
Mumford-shah	16.2614081462
Mapa de contornos	43.20552

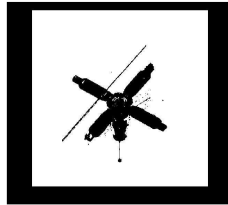
7.3.9. Extracción de información del Mapa de contornos



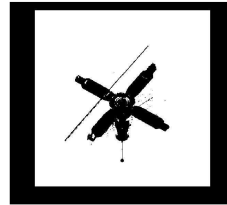
10<



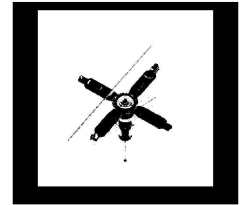
30<



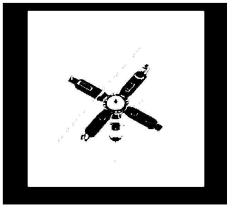
70<



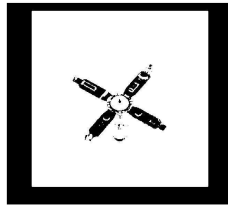
120<



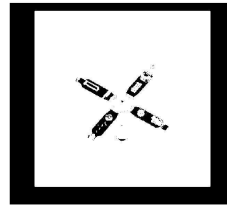
170<



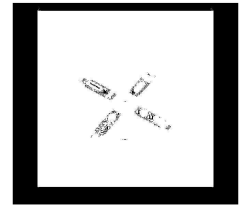
200<



230<



250<



7.3.10. Conclusiones

Vamos a comparar ambos métodos presentados en las imágenes previas. Queríamos notar que la paralelización de Matlab se realiza mediante núcleos de cpu físicos, por tanto la máquina con la que hemos realizado las pruebas ha sido de 2 núcleos físicos, en vez de cuatro hilos de computo. También destacar el hecho que si la paralelización se hubiera sido realizada mediante computación en gpu, la mejora en tiempo en comparación hubiera sido más notable, debido a que se podrían hacer más ejecuciones en paralelo y no solamente 2.

Denotamos que las cuatro primeras imágenes hemos tomado los valores de $\beta = \{1, 10, 20, 40\}$, para Peppers $\beta = [1 : 5 : 400]$ y el resto $[1 : 5 : 100]$. La gran cantidad de betas tomados y la mala paralelización realizada con Matlab, hacen que en estas últimas imágenes el tiempo de computo sea mayor en el mapa de contornos pero también mayor la calidad y cantidad de contornos obtenidos.

Concluyendo, vemos que el método Mumford-Shah funciona mejor en condiciones de ruido, ya que acumulamos contornos que realmente son ruido. En la imagen del cerebro nuestro método no mejora sustancialmente, debido a la poca resolución, en comparación al método de Mumford-Shah, no introduce falsos positivos, aunque pierde bastante detalles. Entrando en la parte de fotografías, vemos que en Peppers los reflejos afectan menos al mapa de contornos. En la imagen de Lena vemos una mayor cantidad de detalles en el mapa de contornos, como son las plumas, los ojos y el pelo. En Cameraman remarca más detalles, sobretodo los edificios del fondo que son muy difícil de segmentar, debido a sus tonos de grises tan parecidos con el fondo. Por último en el Satélite vemos el buen resultado del mapa de contornos, ya que ambos métodos detectan las antenas, pero además el mapa de contornos detecta cada uno de los detalles interiores del satélite y mejorando el resultado en la zona interior del mismo. Todos estos detalles lo podemos ver en el apartado de extracción de información del mapa de contornos.

7.4. Resumen

En este trabajo hemos presentado un modelo de segmentación basado en el funcional de Chan-Vese basado en curvas de nivel. Este modelo permite distinguir diferentes niveles de perfiles, de acuerdo con unos parámetros de regularización. Estos parámetros establecen una especie de multirresolución en la segmentación, a partir del que se puede obtener un *mapa de contornos*, es decir, una fusión de diferentes niveles de contornos.

Hemos comparado la eficiencia del modelo con otros clásicos, (Mumford-Shah, Chan-Vese, Lyseker...), confirmando que el modelo que presentamos converge más rápido en general, y, en algunos casos, presenta una mejor calidad respecto a los demás. Por contra, su calidad disminuye en presencia de ruido, puesto que confunde los perfiles de la imagen con los producidos por el propio ruido.

Entre las ventajas computacionales del modelo aquí introducido destacamos no sólo su velocidad y adaptabilidad a diferentes tipos de imágenes, sino el hecho de ser paralelizable.

8. Trabajo futuro

La línea de investigación seguida en este trabajo está basada en segmentación de imágenes. En este trabajo se han mejorado algunos resultados respecto velocidad y obtención de un método multiresolutorio. En esta línea, a medio y corto plazo pretendemos abordar las siguientes acciones:

- Incluir parámetros de texturas. Para poder estudiar las texturas en la imágenes, pensamos utilizar bases de *wavelets*. Sus coeficientes, nos pueden aportar información local de estas texturas. De esta manera se conseguirán modelos más fiables del mapa de contornos.
- Desarrollar teóricamente resultados sobre Wavelets que nos permitan trabajar con texturas como un concepto teórico y poder caracterizarla.
- Profundizar en las texturas. Sustituir las medias internas y externas, por otro tipo de medida que sea invariante respecto rotaciones o escalados, así podremos juntar aquellos objetos con la misma textura o de texturas similares.
- Añadir el factor tridimensional a la segmentación, mediante varias imágenes de un mismo escenario. En esta parte resulta factible la utilización de métodos numéricos, como pueden ser el SURF o el SIFT, para obtener modelos próximos a la realidad.
- Obtener información de los objetos desde distintos puntos de vista para tratarlos computacionalmente y obtener así una segmentación global de la imagen.

Referencias

- [1] Analí J. Alfaro Alfaro y Iván A. Sipirán Mendoza, *Diseño de un Algoritmo de Segmentación de Imágenes aplicando el funcional de Mumford-Shah para mejorar el desempeño de algoritmos clásicos de segmentación*, Computer Engineering thesis, School of Informatic, University National of Trujillo.
- [2] Chan, T. F., and Vese L. A. (2001). Active Contours without Edges . *Image Processing, IEEE Transactions on*, 10(2), 266-277.
- [3] Chan, T. F., and Vese L. A. (2001). A Level Set Algorithm for Minimizing the Mumford-Shah Functional in Image Processing. *In Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on* (pp. 161-168). IEEE.
- [4] Chan, T. F., and Vese, L. A. (2002) A Multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision* 50(3), 271–293.
- [5] Bertsekas, D. P.(1982). Constrained Optimization and Lagrange Multiplier Methods. *Computer Science and Applied Mathematics, Boston: Academic Press*, 1982, 1.
- [6] Evans, L. C. and Gariepy, R. F. (1991) *Measure theory and fine properties of functions* (Vol. 5). CRC press.
- [7] Oliva, G. H. (2006) Método clásicos de optimización para problemas no-lineales sin restricción. University of Chile
- [8] Lie, J., Lysaker, M., and Tai, X. C.(2005). Piecewise constant level set methods and image segmentation. In *Scale Space and PDE Methods in Computer Vision*, (pp. 573-584). Springer Berlin Heidelberg
- [9] Morel, J. M. and Solimni, S. (1988) Segmentation of images by variational Methods: A constructive approach. *Revista Matemática Complutense*, 1(1), 169.
- [10] Morel, J. M. and Solimni, S. (1989) Segmentation, d'images por méthode variationnelle: Une preuve constructive d'existence. *Comptes rendus de l'Académie des sciences. Serie 1, Mathématique* 308(15), 465-470.
- [11] Jorge N., Wright I., Steven J. (1999) *Numerical optimization* Springer Series in operations research.
- [12] Kato, Z. (2002). Mumford-Shah Energy Functional. Course on Variational and Level Set Methods in Image Processing.
- [13] Kunisch, K. and Tai, X. C.(1997) Sequential and parallel splitting methods for bilinear control problems in Hilbert spaces. *SIAM journal on numerical analysis*, 34(1), 91-118.

- [14] Mumford, D., and Shah, J. (1989) Optimal approximation by piecewise smooth functions and associated variational problems, *Communications on pure and applied mathematics*, 42(5), 577-685.
- [15] Osher, S., and Sethian, J. A. (1988). Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi Formulation. *Journal of computational physics*, 79(1), 12-49.
- [16] Gonzalez, R. C., and Woods, R. E. (2007). *Digital Image Processing , Third Edition*
- [17] Shah, J. (1992). Properties of energy-minimizing segmentations. *SIAM journal on control and optimization*, 30(1), 99-111.
- [18] Tai, X. C., and Yao, C. H. (2006). Image segmentation by piecewise constant Mumford-Shah model without estimating the constants. *JOURNAL OF COMPUTATIONAL MATHEMATICS-INTERNATIONAL EDITION-*, 24(3), 435.
- [19] Candela, V. F., and Marquina, A. (2008). On the Numerical Approximation of Length of (Implicit) Level Curves. *Journal of Scientific Computing*, 35(2-3), 99-113.
- [20] Vese, L., Chan, T. (2000). Image Segmentation using Levels Sets and the Piecewise Constant Mumford-Shah Model. *Kluwer Academic Publishers*.
- [21] Zao, H. K., Chan, T. , Merriman, B. and Osher, S. (1996). A variational level set approach to multiphase motion. *Journal of computation physics* ,127(1), 179-195.