

# Tecnológico Nacional de México

Centro Nacional de Investigación  
y Desarrollo Tecnológico

## Tesis de Maestría

Extracción de Superpíxeles y Fusión Local Mediante un  
Algoritmo de Agrupamiento Difuso Intuitivo

presentado por  
**Ing. David Domínguez López**

como requisito para la obtención del grado de  
**Maestro en Ciencias de la Computación**

Director de tesis  
**Dr. Dante Mújica Vargas**

Codirector de tesis  
**Dr. Raúl Pinto Elías**

Cuernavaca, Morelos, México. Junio de 2019.



"2019, Año del Caudillo del Sur, Emiliano Zapata"

Cuernavaca, Morelos a 12 de junio del 2019  
OFICIO No. DCC/047/2019

Asunto: Aceptación de documento de tesis

**DR. GERARDO V. GUERRERO RAMÍREZ**  
**SUBDIRECTOR ACADÉMICO**  
**PRESENTE**

Por este conducto, los integrantes de Comité Tutorial del Ing. David Domínguez López, con número de control M17CE030, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado "EXTRACCIÓN DE SUPERPIXELES Y SU FUSIÓN LOCAL MEDIANTE UN ALGORITMO DE AGRUPAMIENTO DIFUSO INTUITIVO" y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS

Dr. Dante Mújica Vargas  
Doctor en Comunicaciones y  
Electrónica  
09131756

CO-DIRECTOR

Dr. Raúl Pinto Elías  
Doctor en Ciencias en la  
Especialidad de Ingeniería  
Eléctrica  
3890453

REVISOR 1

Dra. Andrea Magadán Salazar  
Doctorado en Ciencias  
Computacionales  
10654097

REVISOR 2

Dr. Manuel Mejía Lavalle  
Doctor en Ciencias  
Computacionales  
8342472

C.p. M.E. Guadalupe Garrido Rivera - Jefa del Departamento de Servicios Escolares.  
Estudiante  
Expediente

NACS/lmz



**SEP**  
SECRETARÍA DE  
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Centro Nacional de Investigación y Desarrollo Tecnológico

"2019, Año del Caudillo del Sur, Emiliano Zapata"

Cuernavaca, Mor., 12 de junio de 2019  
OFICIO No. SAC/217/2019

Asunto: Autorización de impresión de tesis

**ING. DAVID DOMÍNGUEZ LÓPEZ**  
**CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS**  
**DE LA COMPUTACIÓN**  
**PRESENTE**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "EXTRACCIÓN DE SUPERPIXELES Y SU FUSIÓN LOCAL MEDIANTE UN ALGORITMO DE AGRUPAMIENTO DIFUSO INTUITIVO", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

**ATENTAMENTE**

**Excelencia en Educación Tecnológica®**  
*"Conocimiento y tecnología al servicio de México"*

**DR. GERARDO VICENTE GUERRERO RAMÍREZ**  
**SUBDIRECTOR ACADÉMICO**

C.p. Mtra. Guadalupe Garrido Rivera .- Jefa del Departamento de Servicios Escolares.  
Expediente

GVGR/mcr



SEP TecNM  
CENTRO NACIONAL  
DE INVESTIGACIÓN  
Y DESARROLLO  
TECNOLÓGICO  
SUBDIRECCIÓN  
ACADÉMICA

**cenidet**<sup>®</sup>  
Centro Nacional de Investigación  
y Desarrollo Tecnológico

Interior Internado Palmira S/N, Col. Palmira, C. P. 62490, Cuernavaca, Morelos.  
Tel. (01) 777 3 62 77 70, ext. 4106, e-mail: dir\_cenidet@tecnm.mx

[www.tecnm.mx](http://www.tecnm.mx) | [www.cenidet.edu.mx](http://www.cenidet.edu.mx)



## Agradecimientos

Agradezco a Dios porque ha estado conmigo a cada paso que doy, cuidándome y dándome fortaleza para continuar.

A mis padres, quienes me han brindado su apoyo y orientación, y me han impulsado a seguir adelante.

A mi director de tesis, el Dr. Dante Mújica Vargas, por su paciencia y sus consejos durante el desarrollo de esta tesis, y principalmente, por la oportunidad que me dio para trabajar en esta investigación.

A mi codirector de tesis, el Dr. Raúl Pinto Elías, por sus sugerencias y observaciones, las cuales ayudaron a mejorar esta tesis.

A los miembros del comité revisor, la Dra. Andrea Magadán Salazar, y el Dr. Manuel Mejía Lavalle, por su tiempo para la revisión de esta investigación, y por sus valiosos consejos para mejoramiento de este trabajo.

Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), por darme la oportunidad de realizar los estudios de posgrado.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por el apoyo económico que me brindó durante mis estudios de maestría.

## Resumen

En este trabajo se propone un método eficiente para la extracción de superpíxeles utilizando un enfoque basado en Agrupamiento Difuso Intuitivo (*Intuitionistic Fuzzy C-Means*, IFCM). Considerando las características de este enfoque, para el manejo de la incertidumbre, se aborda el problema del tratamiento de imágenes complejas, en específico, las imágenes de resonancia magnética cerebral. Este tipo de imágenes son difíciles de tratar debido a que usualmente presentan condiciones inherentes como la presencia de ruido, bajo contraste, inhomogeneidad de la intensidad, etc.

El método propuesto es una adaptación del algoritmo IFCM para generar superpíxeles, considerando las propiedades básicas de los superpíxeles, como la adaptación a los bordes naturales de la imagen y la homogeneidad en los superpíxeles. La inicialización del método propuesto se basó en algoritmo SLIC (*Simple Linear Iterative Clustering*) al limitar el espacio de búsqueda en regiones proporcionales al tamaño del superpíxel. La reducción del espacio de búsqueda permite que el algoritmo sea de rápida ejecución.

El método propuesto se comparó con cinco algoritmos para extraer superpíxeles con implementaciones de código abierto. Para la evaluación se utilizó dos estudios de resonancia magnética cerebral, obtenidos del simulador BrainWeb, con diferentes porcentajes de ruido. Adicionalmente, se contempló el conjunto de datos BSDS500 para la evaluación con imágenes a color. La eficiencia del método propuesto se evaluó con respecto a las métricas de recuperación de los límites como: *Rec* (Recuerdo de límites), *UE* (Error de subsegmentación) y *EV* (Variación Explicada). Los resultados cuantitativos y cualitativos, muestran que el enfoque propuesto genera superpíxeles con alta adherencia a los bordes naturales de la imagen, teniendo un rendimiento similar y en algunos casos mejor que los algoritmos considerados para la evaluación.

---

## Abstract

In this work an efficient method for extracting superpixels using an algorithm based on the Intuitionistic Fuzzy C-Means (IFCM) approach is proposed. Considering the characteristics of this approach, for the management of uncertainty, the problem of the treatment of complex images, specifically, magnetic resonance imaging of the brain, is addressed. These types of images are difficult to treat because they usually present inherent conditions such as the presence of noise, low contrast, inhomogeneity of intensity, etc.

The proposed method is an adaptation of the IFCM algorithm to generate superpixels, considering the basic properties of the superpixels, such as the adaptation to the natural edges of the image and the homogeneity in the superpixels. The initialization of the proposed method was based on the Simple Linear Iterative Clustering algorithm (SLIC) by limiting the search space in regions proportional to the size of the superpixel. The reduction of the search space allows the algorithm to be quickly executed.

The proposed method was compared with five algorithms to extract superpixels with open source implementations. For the evaluation, two brain magnetic resonance studies were used, obtained from the BrainWeb simulator, with different percentages of noise. Additionally, the BSDS500 data set was considered for the evaluation with color images. The efficiency of the proposed method was evaluated with respect to the limit recovery metrics such as: Boundary Recall (*Rec*), Undersegmentation Error (*UE*) and Explained Variation (*EV*). The quantitative and qualitative results show that the proposed approach generates super pixels with high adherence to the natural edges of the image, having a similar performance and in some cases better than the algorithms considered for the evaluation.

# Índice General

<b>Resumen</b>	<b>I</b>
<b>Índice de Figuras</b>	<b>V</b>
<b>Índice de Tablas</b>	<b>VII</b>
<b>Nomenclatura</b>	<b>VIII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del Problema . . . . .	2
1.1.1. Delimitación del problema . . . . .	3
1.1.2. Complejidad del problema . . . . .	3
1.2. Objetivos . . . . .	3
1.2.1. Objetivo general . . . . .	3
1.2.2. Objetivos específicos . . . . .	4
1.3. Alcances y Limitaciones . . . . .	4
1.3.1. Alcances . . . . .	4
1.3.2. Limitaciones . . . . .	4
1.4. Justificación . . . . .	5
1.5. Organización de la tesis . . . . .	5
<b>2. Marco Teórico</b>	<b>7</b>
2.1. Segmentación . . . . .	7
2.2. Sobresegmentación . . . . .	9
2.2.1. Superpíxeles . . . . .	10
2.2.2. Algoritmos para generar superpíxeles . . . . .	13
2.3. Agrupamiento Difuso Intuitivo . . . . .	16
2.4. Imagen de Resonancia Magnética Cerebral . . . . .	23
2.5. Discusión . . . . .	26

<b>3. Estado del Arte</b>	<b>27</b>
3.1. Antecedentes . . . . .	27
3.2. Trabajos relacionados . . . . .	30
3.3. Discusión . . . . .	48
<b>4. Metodología</b>	<b>56</b>
4.1. IFCM Superpíxeles . . . . .	56
4.2. Descripción del método . . . . .	57
4.3. Discusión . . . . .	68
<b>5. Experimentación y resultados</b>	<b>69</b>
5.1. Implementaciones . . . . .	69
5.1.1. Repositorio . . . . .	70
5.1.2. Algoritmos . . . . .	72
5.1.3. Conjunto de datos . . . . .	72
5.1.4. Métricas . . . . .	74
5.2. Pruebas con el algoritmo . . . . .	76
5.2.1. Pruebas con el parámetro $\alpha$ . . . . .	76
5.2.2. Optimización de parámetros . . . . .	79
5.3. Evaluación y Comparación . . . . .	80
5.4. Resultados con el conjuntos de datos Brain Web . . . . .	81
5.4.1. RM cerebral en modo T1 . . . . .	81
5.4.2. RM del cerebro en modo T1 con 5% de ruido . . . . .	93
5.4.3. Resultados con BSDS500 . . . . .	101
5.5. Discusión . . . . .	109
<b>6. Conclusiones</b>	<b>111</b>
6.1. Objetivos y alcances logrados . . . . .	111
6.2. Resultados del trabajo . . . . .	113
6.2.1. Productos . . . . .	113
6.2.2. Aportaciones . . . . .	114
6.2.3. Conclusiones . . . . .	114
6.2.4. Trabajo futuro . . . . .	116
<b>A. Documentos correspondientes a participaciones</b>	<b>123</b>

# Índice de Figuras

2.1. Segmentación y sobresegmentación de imágenes. . . . .	10
2.2. Imagen tomada del conjunto de datos BSDS500, sobresegmentada en aproximadamente 400 superpíxeles. . . . .	11
2.3. Comparación de T1-Ponderado, T2-Ponderado y Flair en una RM cerebral. . . . .	24
2.4. Tejidos principales en una RM cerebral . . . . .	25
3.1. Representación del equilibrio en los algoritmos SLIC y CW. . . . .	36
3.2. Segmentación de imágenes utilizando SLIC y agrupación espectral. . . . .	37
3.3. Extracción de superpíxeles con el algoritmo ETPS. . . . .	38
3.4. Flujo del algoritmo para la generación de superpíxeles. . . . .	39
3.5. Aceleración del algoritmo SLIC a través de submuestreo. . . . .	44
4.1. Reducción de espacio de búsqueda. . . . .	59
4.2a. Extracción de superpíxeles con el algoritmo IFCM (a). . . . .	61
4.2b. Extracción de superpíxeles con el algoritmo IFCM (b). . . . .	62
5.1. Imagen de resonancia magnética cerebral con las diferentes representaciones de superpíxeles. . . . .	70
5.2. Adquisición de las imágenes de RM cerebral utilizando el simulador BrainWeb. . . . .	73
5.3. Imagen tomada del conjunto de datos BSDS500- <i>test</i> -2018, la imagen original y sus 5 segmentaciones <i>ground truth</i> . . . . .	74
5.4. Resultados de la experimentación con el parámetro $\alpha$ . . . . .	78
5.5. Esquema de experimentación para la evaluación comparativa. . . . .	80
5.6. Cálculo de las métricas <i>Rec</i> y <i>UE</i> . . . . .	82
5.7. Resultados de la evaluación con <i>Rec</i> , en el conjunto de datos BrainWeb T1. . . . .	84

5.8. Resultados de la evaluación con <i>UE</i> , en el conjunto de datos BrainWeb T1. . . . .	85
5.9. Resultados de la evaluación con <i>EV</i> , en el conjunto de datos BrainWeb T1.. . . .	87
5.10. Resultados del tiempo de ejecución en segundos, en el conjunto de datos BrainWeb T1. . . . .	88
5.11. Extracción de superpíxeles con el conjunto de datos de BrainWeb. . . . .	89
5.12. Resultados cualitativos de la evaluación, en el conjunto de datos de BrainWeb T1 con $K \approx 200$ . . . . .	91
5.13. Resultados cualitativos de la evaluación, en el conjunto de datos de BrainWeb T1 con $K \approx 800$ . . . . .	92
5.14. Resultados de la evaluación con <i>Rec</i> , en el conjunto de datos BrainWeb T1 + 5% de ruido. . . . .	94
5.15. Resultados de la evaluación con <i>UE</i> , en el conjunto de datos BrainWeb T1 + 5% de ruido. . . . .	95
5.16. Resultados de la evaluación con <i>EV</i> , en el conjunto de datos BrainWeb T1 + 5% de ruido. . . . .	97
5.17. Resultados del tiempo de ejecución en segundos, en el conjunto de datos BrainWeb T1 + 5% de ruido. . . . .	98
5.18. Resultados cualitativos de la evaluación, en el conjunto de datos de BrainWeb T1 + 5% de ruido, con $K \approx 200$ . . . . .	99
5.19. Resultados cualitativos de la evaluación, en el conjunto de datos de BrainWeb T1 + 5% de ruido, con $K \approx 800$ . . . . .	100
5.20. Resultados de la evaluación con <i>Rec</i> , en el conjunto de datos BSDS500.102	102
5.21. Resultados de la evaluación con <i>UE</i> , en el conjunto de datos BSDS500.103	103
5.22. Resultados de la evaluación con <i>EV</i> , en el conjunto de datos BSDS500.104	104
5.23. Resultados del tiempo de ejecución en segundos, en el conjunto de datos BSDS500. . . . .	106
5.24. Resultados cualitativos de la evaluación, en el conjunto de datos de BSDS500, con $k \approx 1000$ . . . . .	107
A.1. Escuela de Inteligencia Artificial y Robótica 2018. . . . .	123
A.2. ICMEAE 2018. . . . .	124
A.3. Portada artículo publicado en la revista JCYTA VOL. 2, CENIDET 2019.125	125
A.4. Constancia del artículo publicado en la revista JCYTA VOL. 2, CENIDET 2019. . . . .	126

# Índice de Tablas

2.1. Resumen de los algoritmos para extraer superpíxeles. . . . .	15
2.2. Las secuencias de IRM más comunes y sus tiempos aproximados de TR y TE. . . . .	24
3.1. Resumen del Estado del Arte. . . . .	49
5.1. Resumen de los conjuntos de datos compatibles con en el repositorio.	71
5.2. Resumen de los algoritmos considerados para la evaluación. . . . .	72
5.3. Resultados de la evaluación con la métrica <i>Rec</i> . . . . .	76
5.4. Resultados de la evaluación con la métrica <i>UE</i> . . . . .	77
5.5. Resultados de la evaluación con la métrica <i>EV</i> . . . . .	77
5.6. <i>Rec</i> , en el conjunto de datos BrainWeb T1. . . . .	83
5.7. <i>UE</i> , en el conjunto de datos BrainWeb T1. . . . .	84
5.8. <i>EV</i> , en el conjunto de datos BrainWeb T1. . . . .	86
5.9. Tiempo de ejecución en segundos para diferentes valores superpíxeles <i>K</i> en el conjunto de datos de BrainWeb IRM T1. . . . .	88
5.10. <i>Rec</i> , en el conjunto de datos BrainWeb T1 + 5% ruido. . . . .	93
5.11. <i>UE</i> , en el conjunto de datos BrainWeb T1 + 5% ruido. . . . .	95
5.12. <i>EV</i> , en el conjunto de datos BrainWeb T1 + 5% ruido. . . . .	96
5.13. Tiempo de ejecución en segundos para diferentes valores superpíxeles <i>K</i> en el conjunto de datos de BrainWeb IRM T1 + 5% de ruido. . . . .	97
5.14. Resultados de la evaluación con <i>Rec</i> , en el conjunto de datos BSDS500.101	
5.15. Resultados de la evaluación con <i>UE</i> , en el conjunto de datos BSDS500.103	
5.16. Resultados de la evaluación con <i>EV</i> , en el conjunto de datos BSDS500.104	
5.17. Tiempos de ejecución en segundos para diferentes valores superpíxeles <i>K</i> en el conjunto de datos BSDS500 . . . . .	105
5.18. Comparación final de los algoritmos para extraer superpíxeles. . . . .	108
6.1. Objetivos y alcances realizados. . . . .	111

# Nomenclatura

$m$	Coefficiente difuso ( <i>Fuzzifier</i> ).
$c$	Número de centroides.
$\varepsilon$	Variable de precisión.
$\lambda$	Factor para la transformación a conjuntos difusos intuitivos.
$W$	Ancho de la imagen.
$H$	Largo de la imagen.
$N$	Número de píxeles en la imagen.
$Y$	Segmentación de superpíxeles.
$K$	Número de superpíxeles.
$w$	Factor para controlar la compacidad de los superpíxeles.
$n$	Número de píxeles en un región.
$R_x$	Número de columnas en una región.
$R_y$	Número de filas en una región.
$S$	Conjunto de superpíxeles.
$s_i$	$i$ -ésimo superpíxel.
$\alpha$	Umbral para la fusión de superpíxeles.
$L$	Vector de etiquetas.
$d^{IFS}$	Distancia euclidiana difusa intuitiva.
$Rec$	Recuerdo de límites ( <i>Boundary Recall</i> ).
$UE$	Error de subsegmentación ( <i>Undersegmentation Error</i> ).
$EV$	Variación explicada ( <i>Explained Variation</i> ).
$TP$	Verdadero positivo ( <i>True Positive</i> ).
$FN$	Falso negativo ( <i>False Negative</i> ).
$G$	Conjunto de segmentos ( <i>ground truth</i> ).
$g_i$	$i$ -ésimo segmento <i>ground truth</i> .
$\mu_I$	Media de la imagen.
$\mu_S$	Media del superpíxel.

# Capítulo 1

En este Capítulo se describe el punto de partida para esta tesis, donde en cada sección se aborda los aspectos fundamentales que servirán de soporte, tales como: la descripción, delimitación y complejidad del problema, los objetivos, alcances y limitaciones, así como la justificación.

## Introducción

La sobresegmentación de imágenes mediante superpíxeles tiene por objeto representar a una imagen con precisión a través de regiones de píxeles agrupados por algún criterio de similaridad (intensidad, textura, color, relación espacial, etc.). Los superpíxeles reducen la complejidad de una imagen al representarla con solo un par de cientos de segmentos en lugar de decenas de miles de píxeles. La reducción de la complejidad permite que los superpíxeles sean de gran utilidad en diversas aplicaciones de visión por computadora y procesamiento de imágenes. Los superpíxeles son utilizados principalmente como un preprocesamiento para reducir el costo computacional en etapas de procesamiento posteriores (Tasli et al., 2015).

Algunas propiedades deseadas en los algoritmos para generación de superpíxeles son: adherencia a los límites, velocidad de segmentación, precisión de segmentación, entre otras (Moore et al., 2008). *Simple Linear Iterative Clustering* (SLIC), es uno de los algoritmos para generación de superpíxeles más eficientes que existe en la literatura, utiliza un enfoque basado en *k-means* para generar superpíxeles, tiene una velocidad de segmentación alta y ofrece la posibilidad de controlar la cantidad de superpíxeles. Una desventaja para muchos algoritmos de segmentación basados en superpíxeles, incluyendo SLIC, es la necesidad de ajustar sus parámetros por lo que puede reducir la eficiencia práctica del algoritmo (Achanta et al., 2012). En SLIC la adherencia a los bordes de la imagen se ve influenciada por sus parámetros (el número de superpíxeles deseados  $K$  y el factor de compacidad  $w$ ), así como una relación espacial insuficiente.

Otro factor que puede influir en la adherencia de los bordes de la imagen es el uso del algoritmo *k-means* para la agrupación de los píxeles, debido a que este hace asignaciones totalmente binarias (Han, 2017). Una posible solución para mejorar la adherencia a los bordes de la imagen es considerar el algoritmo de Agrupamiento Difuso Intuitivo, el cual podría realizar una mejor asignación de los píxeles a las regiones.

En esta tesis, se presenta un método para la extracción de superpíxeles que se basa en la etapa de inicialización del algoritmo SLIC y la adaptación del algoritmo de agrupamiento difuso intuitivo, para un agrupamiento local. Esta adaptación permite extraer superpíxeles con alta adherencia a los bordes de la imagen en los conjuntos de datos considerados.

### 1.1. Descripción del Problema

La sobreesegmentación de imágenes utilizando superpíxeles son de gran importancia en las aplicaciones de visión artificial debido a que proporciona una representación más simple y compacta de una imagen. Aunque los métodos de extracción de superpíxeles se aplicaron exitosamente en diversas áreas de análisis y procesamiento de imágenes con resultados satisfactorios, muchos de estos algoritmos requieren de un ajuste de parámetros para dar buenos resultados.

Esta sintonización de parámetros afecta la eficiencia práctica del algoritmo principalmente en la adaptación de los bordes de la imagen (Boemer et al., 2018). La relación espacial entre píxeles también es un factor importante a considerar debido a que influye directamente en la compacidad y regularidad de los superpíxeles, características importantes que son deseables en estos métodos (Fu et al., 2014). Otro inconveniente de estos métodos es la poca eficiencia al tratar imágenes con ruido. Al considerar el algoritmo de agrupamiento difuso intuitivo, así como la relación espacial entre los píxeles, podría ser una solución factible para la extracción eficiente de superpíxeles.

**Hipótesis:** Se puede realizar una mejor extracción de superpíxeles con mayor adherencia a los bordes de la imagen al considerar los grados de pertenencia, no pertenencia e indecisión, propios de un algoritmo de agrupamiento difuso intuitivo. El cual debe incluir información espacial de los píxeles.

### 1.1.1. Delimitación del problema

Este trabajo está enfocado en la extracción de superpíxeles utilizando un algoritmo de agrupamiento difuso intuitivo, el cual permita trabajar con imágenes de resonancia magnética del cerebro. Considerando los beneficios de este algoritmo para el manejo de la incertidumbre, es posible realizar una extracción de superpíxeles con alta adherencia a los bordes de la imagen.

### 1.1.2. Complejidad del problema

Las complejidad del problema radica en los siguientes puntos:

- La modificación del algoritmo de agrupamiento difuso intuitivo para realizar la agrupación en una región limitada.
- Las características inherentes presentes en las imágenes de resonancia magnética como la presencia de ruido, inhomogeneidad de intensidad, bajo contraste, etc; hacen que el procesamiento sea complicado.
- Complejidad del algoritmo IFCM. En términos generales, el algoritmo IFCM tiene un complejidad computacional  $O(p(n + c) + cn)$ , donde  $p$  es el número de muestras en los datos,  $n$  es el número de conjuntos difusos intuitivos (IFS) en una muestra, y  $c$  es el número de grupos. En la mayoría de los casos  $n$  y  $c$  son mucho más pequeños que  $p$ , por lo cual, IFCM se puede ver como un algoritmo lineal en el tamaño de muestra  $p$ .

En la adaptación para extraer superpíxeles, se espera conservar esta característica, además, al implementar la estrategia de reducir el espacio de búsqueda, se podría disminuir el requisito de tiempo.

## 1.2. Objetivos

### 1.2.1. Objetivo general

Implementar y adaptar el algoritmo de Agrupamiento Difuso Intuitivo para una extracción eficiente de superpíxeles en imágenes de resonancia magnética del cerebro.

### 1.2.2. Objetivos específicos

- Estudiar y comprender los algoritmos de Agrupamiento Difuso (Intuitivo) y su utilización para segmentar imágenes.
- Estudiar y comprender el algoritmo SLIC para la extracción de superpíxeles.
- Adaptar el algoritmo de agrupamiento difuso intuitivo para extraer superpíxeles utilizando imágenes en escala de grises.
- Evaluar el desempeño del algoritmo implementado utilizando bases de datos especializadas.
- Comparar el desempeño del algoritmo implementado con otras técnicas existentes en la literatura.

## 1.3. Alcances y Limitaciones

### 1.3.1. Alcances

- Analizar el algoritmo SLIC para extracción de superpíxeles.
- Analizar el algoritmo de agrupamiento difuso intuitivo.
- Implementación de una metodología para la extracción de superpíxeles basado en SLIC utilizando el algoritmo de agrupamiento difuso intuitivo.
- Evaluar el algoritmo con las métricas definidas en la literatura.
- Evaluar el algoritmo con bases de datos especializadas en imágenes de resonancia magnética del cerebro.

### 1.3.2. Limitaciones

- Trabajar con imágenes en escala de grises.
- Generar superpíxeles en imágenes de resonancia magnética cerebral.

### 1.4. Justificación

Las técnicas de segmentación basadas en la extracción de superpíxeles son cada vez más utilizadas para diversas aplicaciones en diferentes áreas del procesamiento de imágenes. El beneficio principal de utilizar superpíxeles es una representación más compacta y simplificada de una imagen, permitiendo mejorar el rendimiento del procesamiento.

Sin embargo, la eficiencia de los algoritmos de extracción de superpíxeles puede verse afectada por diversos factores como: una mala inicialización de sus parámetros, relación espacial entre píxeles insuficiente, etc. El algoritmo de Agrupamiento Difuso Intuitivo (*Intuitionistic Fuzzy C-Means* IFCM) es más robusto para la agrupación debido a que considera los grados de pertenencia, no pertenencia e indecisión para cada elemento. Implementar este enfoque se podría mejorar la extracción de superpíxeles con mayor adherencia a los bordes de la imagen.

### 1.5. Organización de la tesis

La presente tesis se organiza de la siguiente manera: En el Capítulo 2 se describen los conceptos de segmentación y sobresegmentación de imágenes, donde se analiza las características particulares de estas técnicas. También se proporciona una breve introducción de los métodos de sobresegmentación de imágenes basado en algoritmos para generación de superpíxeles.

La descripción de estos métodos se realiza de acuerdo al enfoque que utilizan. Posteriormente, se presenta una descripción del algoritmo de agrupamiento difuso intuitivo, donde se estudian y analizan sus propiedades y características. Además, se da una descripción de las imágenes de resonancia magnética cerebral, analizando su importancia en la imagenología médica.

En el Capítulo 3, se presenta un análisis de los trabajos recientes, relacionados con las aplicaciones de los algoritmos para generación de superpíxeles en imágenes médicas, así como en imágenes naturales. Este análisis es base para la evaluación de los resultados de esta tesis.

El Capítulo 4 describe a detalle el método propuesto para generar superpíxeles utilizando el algoritmo de agrupamiento difuso intuitivo. Como principal contribución, el algoritmo de agrupamiento difuso intuitivo se adapta para generar superpíxeles, considerando las características básicas de estos algoritmos, como: adaptación a los bordes de la imagen y homogeneidad de los segmentos. El enfoque propuesto, se aplica para generar superpíxeles en imágenes en escala de grises. Adicionalmente se hizo una extensión para imágenes a color, realizando modificaciones pequeñas al algoritmo propuesto.

En el Capítulo 5 se detallan las implementaciones y se muestra el desempeño del método propuesto comparado con 5 algoritmos para generación de superpíxeles clásicos en la literatura. En la evaluación se contempló un subconjunto de imágenes de resonancia magnética cerebral, obtenida del simulador *BrainWeb: Simulated Brain Database*. Para este conjunto de datos se consideraron diferentes condiciones de ruido.

Las pruebas adicionales con imágenes a color, se realizaron con el conjunto de datos BSDS500 (*Berkeley Segmentation Data Set*). Finalmente, en el Capítulo 6, se exponen las conclusiones obtenidas, recomendaciones para trabajos futuros, así como un análisis de los objetivos completados y de los alcances planteados en esta investigación.

# Capítulo 2

## Marco Teórico

En este capítulo se presenta diversos conceptos básicos, complementarios y específicos relacionados con la presente investigación. De igual forma, se da una breve introducción de los algoritmos para la sobresegmentación de imágenes con superpíxeles tomando en cuenta los diferentes enfoques que existen en la literatura; así como una descripción general de las imágenes de resonancia magnética cerebral, su complejidad y el impacto que tienen en la imagenología médica.

### 2.1. Segmentación

La segmentación se define como el proceso de dividir una imagen en áreas homogéneas (segmentos) con características similares (es decir, color, contraste, brillo, textura y otras características de bajo nivel) basadas en un criterio predefinido (El Aziz et al., 2016). El proceso de segmentación puede ser visto como el problema de la partición  $R$  en una serie de subregiones  $R_1, R_2, \dots, R_N$  tales que

$$R = \bigcup_{i=1}^N R_i \quad (2.1)$$

$$R_i \cap R_j = \emptyset \text{ para } i \neq j \quad (2.2)$$

donde  $R$  es la región espacial ocupada por la imagen y  $R_i$  es un conjunto conectado,  $i = 1 \dots N$ . La expresión (2.1) se refiere a que la segmentación debe estar completa; es decir, cada píxel debe pertenecer a una región y los puntos de una región deben estar conectados en algún sentido predefinido, la expresión (2.2) indica que las regiones deben estar disjuntas (Caponetti and Castellano, 2017).

Las técnicas populares utilizadas para la segmentación de imágenes son: métodos basados en fronteras, métodos basados en la región y métodos basados en la intensidad de los píxeles. Todas estas técnicas son diferentes entre sí, las cuales se describen con mas detalle de la siguiente manera.

### **Métodos basados en fronteras**

Estas técnicas se basan en la ubicación de las fronteras de la imagen. Las fronteras proporcionan información importante de los límites del objeto e indican la extensión física de los objetos en la imagen (He et al., 2013). Se caracterizan porque proporcionan contornos bien definidos y trabajan bien cuando las regiones en ambos lados del contorno tienen características muy diferentes. Sin embargo, esta condición no suele darse en imágenes reales. Los métodos de segmentación basada en contornos se pueden agrupar en dos categorías, las técnicas locales y globales (Davis, 1975).

- (a) Las técnicas locales de detección de bordes suelen calcular el gradiente de las máscaras de diferenciales con el fin de detectar y seguir el contorno de las regiones.
- (b) Las técnicas globales de detección de fronteras se basan en la búsqueda de heurísticas y en la programación. Suelen suavizar la información del tono y luego lo combinan con la información de los contornos extraídos en niveles de gris, obteniendo un mapa de las fronteras que descartan los efectos derivados de la iluminación.

### **Métodos basados en la región**

Las técnicas de segmentación basadas en regiones parten de la suposición de la existencia de un elemento (píxel o región) representativo de la región que se pretende segmentar: la semilla. De forma muy general, lo que se hace es estudiar aquellos elementos topológicamente conectados con la semilla (entendiendo por tales, los que estén en su entorno o con los que exista un arco, según el caso) y fusionar con ella aquellos que cumplan una determinada condición de homogeneidad.

Este proceso se repite con la nueva región obtenida, hasta obtener el resultado. Habitualmente se sigue un criterio de homogeneidad o uniformidad de color, aunque hay casos en que esta homogeneidad viene dada por otro tipo de criterios (Mújica, 2013). Un método básico entre estos es el crecimiento de la región que agrupa píxeles o subregiones en regiones más grandes en función de criterios predefinidos para el crecimiento (Caponetti and Castellano, 2017).

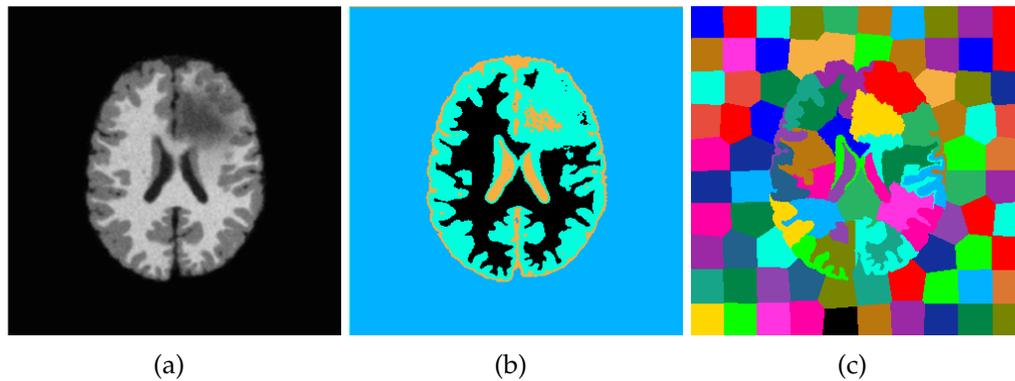
### Métodos basados en la intensidad de los píxeles

Estos métodos se basadas en las propiedades de los píxeles y su entorno. Algunas de las técnicas que se enmarcan dentro de esta categoría son las que utilizan umbrales, que constituyen las primeras propuestas, que con el tiempo fueron evolucionando para aplicarse sobre histogramas de la imagen. Las técnicas basadas en histograma construyen el histograma con la información global obtenida de una imagen dada e identifican el representante de cada grupo utilizando el histograma (Cho et al., 2014).

## 2.2. Sobresegmentación

La sobresegmentación puede definirse como el resultado de un algoritmo en el que las regiones deseadas están representadas por una gran cantidad de regiones (Fisher et al., 2013). Un ejemplo de estos métodos son los algoritmos para generación de superpíxeles, que están diseñados específicamente para sobresegmentar una imagen en regiones con características homogéneas (Xiang et al., 2010). La Figura 2.1 muestra un ejemplo de segmentación y sobresegmentación aplicados en una imagen de resonancia magnética cerebral.

La diferencia entre estos dos enfoques es que, por una parte, el objetivo de la segmentación es dividir la imagen en un conjunto de regiones semánticamente significativas. El resultado de una segmentación es fundamental para realizar tareas más específicas como la clasificación, la identificación o la descripción de estructuras u objetos. Por ejemplo, la Figura 2.1-(b) muestra la segmentación utilizando el algoritmo *Fuzzy C-Means* en cuatro regiones (toma en cuenta el fondo) que representa los tipos de tejido principales en una imagen de resonancia magnética cerebral: materia blanca, materia gris y líquido cefalorraquídeo.



**Figura 2.1:** Segmentación y sobresegmentación de imágenes; (a) imagen original, (b) segmentación utilizando el algoritmo *Fuzzy C-Means* con  $c = 4$ , (c) sobresegmentación utilizando el algoritmo SLIC, con  $k \approx 100$ .

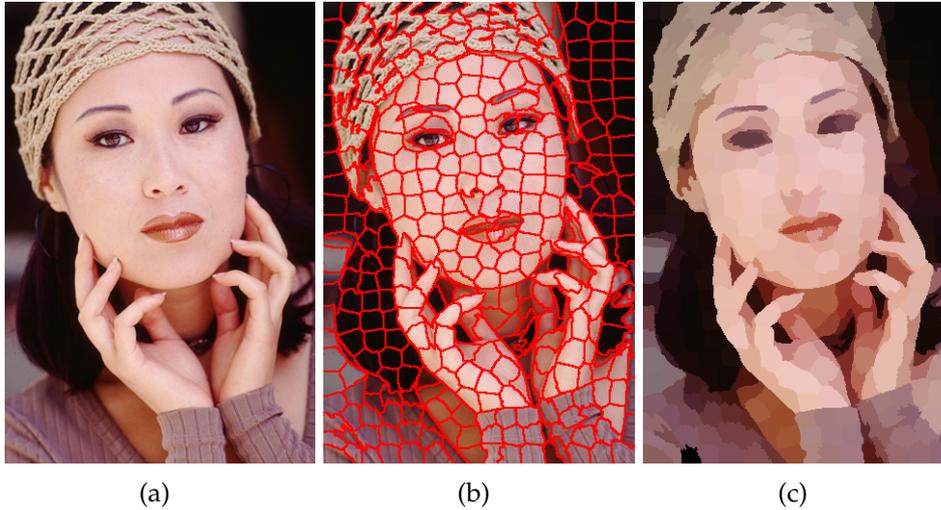
Por otra parte, un algoritmo de sobresegmentación busca dividir una imagen en una gran cantidad de regiones en lugar de segmentos a nivel de objeto. Los algoritmos de sobresegmentación están enfocados en reducir la complejidad de la imagen y comúnmente se aplican como paso de preprocesamiento para mejorar el rendimiento de las aplicaciones (Choi and Oh, 2016). La Figura 2.1-(c) muestra la imagen sobresegmentada utilizando superpíxeles con aproximadamente 100 regiones.

La sobresegmentación utilizando algoritmos para generación de superpíxeles se considera un subproblema de segmentación de imágenes con el objetivo de proporcionar una representación más simple, compacta y eficiente de una imagen sin perder los detalles entre píxeles (Boemer et al., 2018).

### 2.2.1. Superpíxeles

Los superpíxeles son regiones homogéneas de píxeles agrupados comúnmente por características de bajo nivel (color, textura, relación espacial, etc.) proporcionando una representación más simple, compacta y eficiente de una imagen. Los superpíxeles permiten representar una imagen con sólo un par de cientos de segmentos en lugar de decenas de miles de píxeles (Schick et al., 2014). Fueron introducidos por (Ren and Malik, 2003), como entidades más naturales y coherentes de una imagen, permitiendo al mismo tiempo disminuir sustancialmente la complejidad del gráfico y el tiempo de cálculo (Tasli et al., 2015).

En la Figura 2.2 se muestra una imagen sobresegmentada con superpíxeles utilizando el algoritmo SLIC, así como las diferentes formas de visualizar los superpíxeles (contornos de las regiones e intensidad media de las regiones).



**Figura 2.2:** Imagen tomada del conjunto de datos BSDS500, sobresegmentada en aproximadamente 400 superpíxeles utilizando el algoritmo SLIC; (a) imagen original, (b) superpíxeles representados con contornos, (c) representación de la imagen con la intensidad media de los superpíxeles.

Un método de sobresegmentación de superpíxeles intenta contraer píxeles similares de una región local más grande de igual valor de píxeles (Krig, 2014). Los algoritmos para generación de superpíxeles toman como entrada una imagen  $I : W \times H \rightarrow \mathbb{R}^3$ , si la imagen es a color, con ancho  $W$ , altura  $H$  y  $N = WH$  píxeles. La sobresegmentación se obtiene dividiendo la imagen en un conjunto de  $K$  superpíxeles

$$S = \{S_1, \dots, S_K\} \quad \text{con} \quad S_k \subseteq W \times H \quad (2.3)$$

donde  $S = \{S_1, \dots, S_K\}$  representa los componentes conectados. Además, los superpíxeles  $S_i$  son disjuntos y descomponen la imagen  $I$ :

$$S_i \cap S_j = \emptyset \quad \forall i, j \quad \text{y} \quad \bigcup_{S_i \in S} S_i = W \times H \quad (2.4)$$

Por lo tanto, se puede definir la sobresegmentación de superpíxeles como una función  $s : W \times H \rightarrow K$ , que asigna un único superpíxel a cada píxel  $x_n \in W \times H$ .

Los algoritmos para generación de superpíxeles existentes son ampliamente utilizados debido a que reducen la complejidad de la imagen con una pérdida mínima de información, además, ayudan a mejorar el rendimiento de las aplicaciones cuando se usan como preprocesamiento. Por lo cual, los algoritmos para generación de superpíxeles se han convertido en una técnica popular en las diferentes aplicaciones del procesamiento de imágenes y visión por computadora (Tasli et al., 2015).

Por ejemplo, en los trabajos de (Tighe and Lazechnik, 2010) y (Strassburg et al., 2015), los superpíxeles se utilizaron para el análisis de imágenes mediante segmentación semántica; también se han aplicado para la detección de objetos (Ghadiri et al., 2018), para el rastreo de puntos en 3D (Maghsoudi and Spence, 2017), segmentación de imágenes biomédicas (Funke et al., 2017), análisis de imágenes médicas (Verma et al., 2013), clasificación de imágenes (Shi and Pun, 2018), detección de contornos (Wang et al., 2017b), por mencionar sólo algunas aplicaciones.

### Características de los algoritmos para generar superpíxeles

En los últimos años se han propuesto un gran número de técnicas sofisticadas con diferentes enfoques, con el objetivo de mejorar la calidad de la sobresegmentación y cubrir las características esenciales que requieren los algoritmos para generar superpíxeles. En el trabajo de (Liu et al., 2014) se mencionan las siguientes propiedades deseables para la sobresegmentación utilizando superpíxeles:

- **La precisión.** Mide cómo los superpíxeles se adhieren bien a los bordes de la imagen. Para preservar más información para el procesamiento futuro en superpíxeles, la segmentación de superpíxeles debe ser lo más precisa posible.
- **Regularidad.** Mide la forma en que un superpíxel está cerca de los cuadrados. Los superpíxeles se utilizan para representar la información de los píxeles. Los superpíxeles regulares pueden conservar las propiedades típicas de los píxeles, como la topología espacial, la estructura, la homogeneidad y la información isométrica.
- **Eficiencia.** Como paso de preprocesamiento en la visión por computadora, los superpíxeles se utilizan para reducir la complejidad computacional. Debe ser rápido y eficiente en memoria para su cálculo y fácil de usar.

### 2.2.2. Algoritmos para generar superpíxeles

En esta sección, se presenta un breve análisis de los diferentes algoritmos superpíxel que fueron publicados recientemente. De acuerdo con el enfoque que utilizan para generar superpíxeles, se pueden mencionar las siguientes categorías.

**Basados en cuencas.** El enfoque de la cuenca hidrográfica toma la imagen como una superficie topográfica donde las altas intensidades denotan picos y colinas, mientras que las bajas intensidades denotan valles. El enfoque parte desde el mínimo local para producir cuencas hidrográficas, posteriormente, cada valle aislado (mínimo local) se llena con agua de diferentes colores (etiquetas). A medida que el agua sube, dependiendo de los picos (gradientes) cercanos, el agua de diferentes valles comienza a fusionarse. Para evitar eso, se construyen barreras en los lugares donde el agua se fusiona. El proceso de llenar el agua y construir barreras continúa hasta que todos los picos estén bajo el agua. Finalmente la segmentación se obtiene de las barreras que se obtuvieron.

Algunos algoritmos superpíxel basados en este enfoque controlan la compacidad, por ejemplo **CW** (*Compact Watershed*) (Neubert and Protzel, 2014), controla la compacidad con un sólo parámetro. Otro ejemplo es **WP** (*WaterPixels*) (Machairas et al., 2015), que es una implementación modificada que considera las restricciones de compacidad de los superpíxeles y puede controlar el número de superpíxeles directamente. La mayoría de los trabajos se basan en (Michielsen and De Raedt, 2000) para la implementación de este método.

**Basados en agrupación.** Estos algoritmos se basan en algoritmos de agrupación para generar superpíxeles. Por ejemplo, el algoritmo **SLIC** (*Simple Linear Iterative Clustering*) (Achanta et al., 2010) está inspirado en el algoritmo *k-means* para agrupar píxeles. El número de superpíxeles y la compacidad son controlables a través de los parámetros  $K$  y  $w$  respectivamente.

Para agrupar píxeles, **SLIC** utiliza una métrica de distancia que combina la información espacial y de color de cada píxel. Este algoritmo requiere de un procesamiento posterior para reforzar la conectividad entre los superpíxeles. Otros enfoques como **LSC** (*Linear Spectral Clustering*) (Li and Chen, 2015), se basan en agrupación espectral utilizando una función de costo optimizada que combina la función objetivo del algoritmo *k-means* y *Normalized Cuts* (**NC**).

**Basados optimización energética.** Estos algoritmos optimizan iterativamente una energía formulada. Por ejemplo, En el algoritmo **ETPS** (*Efficient Topology Preserving Segmentation*) (Yao et al., 2015), la imagen se divide en una cuadrícula regular como segmentación inicial de superpíxeles, y los píxeles se intercambian entre superpíxeles vecinos con respecto a la energía. El número de superpíxeles y la compacidad se pueden controlar. El algoritmo emplea una única iteración para converger en un mejor mínimo de la energía, esto permite que sea rápido.

**Basados en grafos.** Estos algoritmos tratan cada píxel como un nodo en un grafo. Las similitudes entre los píxeles vecinos se definen como ponderaciones de borde, que a menudo se calculan como diferencias de color. Los superpíxeles se obtienen al minimizar una función de costo definida en el gráfico. La complejidad del cálculo varía según el método. Algunos algoritmos populares que utilizan este enfoque son: **NC** (*Normalized Cuts*) (Shi and Malik, 2000), **FH** (*Felzenszwalb and Huttenlocher*) (Felzenszwalb and Huttenlocher, 2004) y **POISE** (*Proposals for Objects from Improved*) (Humayun et al., 2015).

**Basados en evolución del contorno.** Estos algoritmos se basan en un esquema de crecimiento de región agrupando los píxeles perceptivos y adyacentes para representar superpíxeles. Un algoritmo popular que usa este enfoque es **ERGC** (*Eikonal Region Growing Clustering*) (Buysens et al., 2014). El enfoque de este algoritmo es un esquema de crecimiento de la región, donde el mapa potencial utilizado es dinámico y evoluciona durante la difusión. **ERGC** genera superpíxeles regulares y compactos con bajo costo computacional. **ERGC** fue introducido para imágenes 2D, pero puede extenderse a volúmenes 3D para producir supervóxeles con modificaciones menores del algoritmo.

En el trabajo de (Stutz et al., 2017) se hace una clasificación de los algoritmos para extraer superpíxeles con implementaciones de código abierto, los cuales se listan en la Tabla 2.1. Los algoritmos considerados se clasificaron de acuerdo con el enfoque que utilizan para generar los superpíxeles. También se proporcionan otros detalles como los espacios de color compatibles con los algoritmos y los parámetros que son controlables.

Tabla 2.1: Resumen de los algoritmos para extraer superpíxeles (Stutz et al., 2017).

Nombre	Categorización	Escala de grises					Espacios de color					Control sobre		
		RGB	Lab	Luv	HSV	YCrCb	# Superpíxeles	# Iteraciones	Compacidad					
W	Cuena	✓	✓	✓	✓	✓	✓	-	-	-	✓	-	-	
EAMS	Densidad	-	✓	✓	-	-	-	-	-	-	-	-	-	
NC	Grafo	-	✓	-	-	-	-	-	-	-	✓	-	-	
FH	Grafo	-	✓	-	-	-	-	-	-	-	-	-	-	
RW	Grafo	-	✓	-	-	-	-	-	-	-	✓	-	-	
QS	Densidad	-	✓	-	✓	-	-	-	-	-	-	-	-	
PF	Trayectoria	-	✓	-	-	-	-	-	✓	-	-	-	-	
TP	Evolución del contorno	-	✓	-	-	-	-	-	-	-	✓	-	-	
CIS	Grafo	✓	✓	-	-	-	-	-	-	-	✓	✓	-	
SLIC	Agrupación	-	✓	-	✓	-	-	-	-	-	✓	✓	✓	
CRS	Optimización energética	✓	✓	-	-	-	-	-	✓	-	✓	✓	✓	
ERS	Grafo	-	✓	-	-	-	-	-	-	-	✓	-	-	
PB	Grafo	-	✓	-	-	-	-	-	-	-	✓	-	-	
DASP	Agrupación	-	✓	-	-	-	-	-	-	-	✓	✓	✓	
SEEDS	Optimización energética	-	✓	-	✓	-	-	-	✓	-	✓	✓	-	
TPS	Trayectoria	-	-	-	-	-	-	-	-	-	✓	-	-	
VS	Agrupación	-	✓	-	✓	-	-	-	-	-	✓	-	✓	
CCS	Optimización energética	-	✓	-	✓	-	-	-	-	-	✓	✓	✓	
VCCS	Agrupación	-	✓	-	-	-	-	-	-	-	-	-	-	
CW	Cuena	-	✓	-	-	-	-	-	-	-	✓	-	✓	
ERGC	Evolución del contorno	-	✓	-	✓	-	-	-	-	-	✓	-	✓	
MSS	Cuena	-	✓	-	-	-	-	-	-	-	✓	-	-	
preSLIC	Agrupación	-	✓	-	✓	-	-	-	-	-	✓	✓	✓	
WP	Cuena	-	✓	-	-	-	-	-	-	-	✓	-	✓	
ETPS	Optimización energética	-	✓	-	-	-	-	-	-	-	✓	✓	✓	
LSC	Agrupación	-	✓	-	✓	-	-	-	-	-	✓	✓	✓	
POISE	Grafo	-	✓	-	-	-	-	-	-	-	✓	-	-	
SEAW	Wavelet	-	✓	-	-	-	-	-	-	-	-	-	-	

## 2.3. Agrupamiento Difuso Intuitivo

Los métodos de agrupamiento difuso asumen que cada punto de datos pertenece a más de un grupo con diferentes grados de pertenencia. El principal algoritmo de agrupamiento difuso es llamado *Fuzzy C-Means* (FCM) (Caponetti and Castellano, 2017). Los métodos de agrupamiento basados en FCM dependen de la selección de parámetros, son sensibles al ruido y su convergencia depende principalmente de la inicialización de los centros de agrupación (Dubey et al., 2016).

El enfoque difuso intuitivo surgió de la generalización de la teoría de los conjuntos difusos, conocida como la teoría de Conjuntos Difusos Intuitivos (*Intuitionistic Fuzzy Sets* o IFS). En las últimas décadas, la teoría de IFS ha recibido más atención de investigadores y profesionales, y se ha aplicado a varios campos, incluidos la toma de decisiones, la programación lógica, el diagnóstico médico, el reconocimiento de patrones, los sistemas robóticos, la topología difusa, el aprendizaje automático y predicción de mercado, etc. (Atanassov, 2017).

### Conjuntos difusos intuitivos

El enfoque difuso intuitivo parte de la generalización de la teoría de conjuntos difusos introducción por Lotfi A. Zadeh. La teoría de conjuntos difusos se ha aplicado ampliamente en varios campos de la sociedad moderna. Sin embargo, la función de pertenencia es sólo una función de un valor, que no se puede utilizar para expresar las evidencias de soporte y objeción simultáneamente en muchas situaciones prácticas (Xu, 2013). El concepto del conjunto difuso de Zadeh, se define como.

**Definición 2.1.** Sea  $X$  un conjunto fijo, entonces un conjunto difuso se denomina

$$F = \{\langle x, \mu_F(x) \rangle | x \in X\} \quad (2.5)$$

donde  $\mu_F$  es la función de pertenencia de  $F$ ,  $\mu_F : X \rightarrow [0,1]$  y  $\mu_F(x)$  indica el grado de pertenencia del elemento  $x$  a  $F$ , que es un valor único que pertenece a la unidad intervalo cerrado  $[0,1]$  (Zadeh, 1965).

La definición (2.1) representa la extensión de la función característica que lleva el valor de  $\{0,1\}$  (valor binario) a la función de pertenencia que puede tomar cualquier valor del intervalo cerrado  $[0,1]$ .

## 2.3 Agrupamiento Difuso Intuitivo

Krassimir Todorov generalizó la teoría de conjuntos difusos de Zadeh con el concepto de Conjunto Difuso Intuitivo (IFS), que se caracteriza por una función de pertenencia, una función de no pertenencia y una función de indeterminación (grado de indecisión) (Chaira, 2010). Un conjunto difuso intuitivo se define de la siguiente manera.

**Definición 2.2.** Dado un conjunto difuso intuitivo  $A$  en un conjunto finito  $X$ , su representación matemáticamente se da como:

$$A = \{\langle x, \mu_A(x), \nu_A(x) \rangle | x \in X\} \quad (2.6)$$

donde las funciones  $\mu_A(x), \nu_A(x) : x \in X \rightarrow [0, 1]$  son, respectivamente, la función de pertenencia y la función de no pertenencia de un elemento  $x$  en un conjunto finito  $X$ , con la condición necesaria:

$$0 \leq \mu_A(x) + \nu_A(x) \leq 1, \forall x \in X \quad (2.7)$$

Por otra parte, es necesario tener en cuenta un tercer parámetro, conocido como el índice difuso intuitivo o grado de indecisión  $\pi_A(x)$ , que surge debido a la falta de conocimiento o "error personal" al asignar el grado de pertenencia (Atanassov, 1986). El grado de indecisión  $\pi_A(x)$  se calcula como:

$$\pi_A(x) = 1 - \mu_A(x) - \nu_A(x), \forall x \in X \quad (2.8)$$

con la condición

$$0 \leq \pi_A(x) \leq 1, \forall x \in X \quad (2.9)$$

Por lo tanto, con la introducción de un grado de indecisión, un conjunto difuso intuitivo  $A$  en  $X$  puede representarse como:

$$A = \{(x, \mu_A(x), \nu_A(x), \pi_A(x)) | x \in X\} \quad (2.10)$$

con la condición

$$\pi_A(x) + \mu_A(x) + \nu_A(x) = 1 \quad (2.11)$$

En particular, si

$$\pi_A(x) = 1 - \mu_A(x) - [1 - \mu_A(x)] = 0, x \in X \quad (2.12)$$

entonces  $A$  se reduce al conjunto difuso de Zadeh. Por lo tanto, los conjuntos difusos son los casos especiales de las IFS. Estas características permiten que los conjuntos difusos intuitivos puedan describir los caracteres difusos de las cosas de manera más detallada y exhaustiva, y por lo tanto es más útil para tratar la vaguedad y la incertidumbre que la teoría clásica de conjuntos difusos (Xu and Cai, 2012).

### Operaciones con conjuntos difusos intuitivos

A los conjuntos difusos intuitivos se les puede aplicar determinados operadores, o bien pueden realizarse operaciones entre ellos. Al aplicar un operador sobre un sólo conjunto difuso intuitivo se obtiene otro conjunto difuso intuitivo; de la misma manera al combinar dos o más conjuntos mediante alguna operación, se obtendrá otro conjunto (Kacprzyk, 2013). Las siguientes operaciones son válidas para todos los conjuntos difusos intuitivos  $A$  y  $B$  del conjunto  $X$ :

- $A \cup B = \{max(\mu_A, \mu_B), min(\nu_A, \nu_B)\}$
- $A \cap B = \{min(\mu_A, \mu_B), max(\nu_A, \nu_B)\}$
- $A \prec B = \{x, \mu_A(x) < \mu_B(x), \nu_A(x) < \nu_B(x)\}$
- $\bar{A} = \{x, \nu_A(x), \mu_A(x)\}$
- $A \leq B = \{x, \mu_A(x) \leq \mu_B(x), \nu_A(x) \geq \nu_B(x)\}$
- $A \cdot B = \{x, \mu_A(x) \cdot \mu_B(x), \nu_A(x) + \nu_B(x) - \nu_A(x) \cdot \nu_B(x)\}$

### Medidas de distancias entre conjuntos difusos intuitivos

La distancia es un concepto importante en muchos campos, como el análisis de decisiones, el reconocimiento de patrones, en la teoría de conjuntos difusos intuitivos, etc. La distancia en el contexto de la teoría difusa intuitiva se utiliza para medir la diferencia entre dos conjuntos difusos intuitivos. Popularmente, el concepto de una distancia normalizada entre dos conjuntos difusos intuitivos requiere que la medida de la distancia normalizada pertenezca a la unidad de intervalo  $[0,1]$  (Szmidt, 2014).

Matemáticamente, se define de la siguiente manera.

**Definición 2.3.** Sea  $d : F(X) \times F(X) \rightarrow [0,1]$  un mapeo, de modo que para cualquier conjunto difuso intuitivo  $A, B$  y  $C$  de  $F(X)$ , satisfaga las siguientes propiedades:

1.  $0 \leq d(A,B) \leq 1$
2.  $d(A,B) = 0$  si y sólo si  $A = B$
3.  $d(A,B) = d(B,A)$
4.  $d(A,B) \leq d(A,C) + d(C,B)$

donde  $d(A,B)$  es la distancia normalizada entre los conjuntos difusos intuitivos  $A$  y  $B$  (Li, 2014).

La distancia Minkowski normalizada entre cualquier conjunto difuso intuitivo  $A$  y  $B$  se obtiene de la siguiente manera

$$d_q(A,B) = \sqrt[q]{\frac{1}{2n} \sum_{j=1}^n [(\mu_A(x_j) - \mu_B(x_j))^q + (v_A(x_j) - v_B(x_j))^q + (\pi_A(x_j) - \pi_B(x_j))^q]} \quad (2.13)$$

donde  $q > 0$ , es un parámetro de distancia, que puede tomar un valor adecuado según la necesidad en situaciones de administración real, y  $n$  es el número de datos en el conjunto. Cuando el parámetro de distancia  $q = 1$  en la ecuación (2.13), la distancia Hamming normalizada entre cualquier conjunto difuso intuitivo  $A$  y  $B$  se obtiene de la siguiente manera

$$d_1(A,B) = \frac{1}{2n} \sum_{j=1}^n (|\mu_A(x_j) - \mu_B(x_j)| + |v_A(x_j) - v_B(x_j)| + |\pi_A(x_j) - \pi_B(x_j)|) \quad (2.14)$$

Cuando  $q = 2$ , la distancia euclidiana normalizada entre cualquier conjunto difuso intuitivo  $A$  y  $B$  se obtiene de la siguiente manera

$$d_2(A,B) = \sqrt{\frac{1}{2n} \sum_{j=1}^n [(\mu_A(x_j) - \mu_B(x_j))^2 + (v_A(x_j) - v_B(x_j))^2 + (\pi_A(x_j) - \pi_B(x_j))^2]} \quad (2.15)$$

## Algoritmo de Agrupamiento Difuso Intuitivo

Para agrupar conjuntos difusos intuitivos (Xu & Wu, 2010) desarrollaron el algoritmo de agrupamiento difuso intuitivo (*Intuitionistic Fuzzy C-Means*, IFCM), que se basa en el método de agrupación *Fuzzy C-Means* y las medidas de distancias básicas entre los conjuntos difusos intuitivos. Lo que diferencia a IFCM de otros algoritmos como *k-means* es que no decide la pertenencia absoluta de un punto de datos a un grupo dado; en su lugar, calcula la posibilidad (el grado de pertenencia) de que un punto de datos pertenezca a ese grupo.

Por lo tanto, dependiendo de la precisión del agrupamiento que se requiere en la práctica, se pueden establecer medidas de tolerancia apropiadas. Dado que no se calcula la pertenencia absoluta, IFCM puede ser rápido debido a que la cantidad de iteraciones necesarias para lograr un ejercicio de agrupamiento específico corresponde a la precisión requerida. El algoritmo IFCM encuentra una partición difusa de los datos minimizando la siguiente función objetivo:

$$\min J_m(\mathbf{U}, \mathbf{V}^{IFS} : \mathbf{X}^{IFS}) = \sum_{i=1}^c \sum_{j=1}^N (u_{ij})^m d^2(x_j^{IFS} - v_i^{IFS}) \quad (2.16)$$

Sujeto a la condición:

$$\sum_{i=1}^c u_{ij} = 1, 1 \leq j \leq N \quad (2.17)$$

donde  $X^{IFS} = (x_1^{IFS}, x_2^{IFS}, \dots, x_n^{IFS})$  son  $N$  datos difusos intuitivos y cada  $x_j^{IFS}$  está representado en términos de conjuntos difusos intuitivos como  $x_j^{IFS} = \{(\mu(x_j), \nu(x_j), \pi(x_j))\}$ ,  $m(1 < m < \infty)$  es el factor difuso,  $c(1 < c < N)$  es el número de grupos, y  $V^{IFS} = \{v_1^{IFS}, v_2^{IFS}, \dots, v_c^{IFS}\}$  son los centroides de los grupos y cada  $v_i^{IFS}$  está representado en términos de conjuntos difusos intuitivos como  $v_i^{IFS} = \{(\mu(v_i), \nu(v_i), \pi(v_i))\}$ ,  $u_{ij}(0 \leq u_{ij} \leq 1)$  es el grado de pertenencia y  $U = (u_{ij})_{c \times N}$  es la matriz de pertenencia. La matriz de pertenencia  $U$  también cumple la condición  $0 < \sum_{i=1}^c u_{ij} < N, \forall i$  (Verma et al., 2016).

El término  $d^2(x_j^{IFS} - v_i^{IFS})$  es la distancia euclidiana difusa intuitiva entre los datos  $x_j^{IFS}$  y el centroide  $v_i^{IFS}$ , que se define como:

$$d^2(x_j^{IFS} - v_i^{IFS}) = ((\mu(x_j) - \mu(v_i))^2 + (\nu(x_j) - \nu(v_i))^2 + (\pi(x_j) - \pi(v_i))^2) \quad (2.18)$$

La pertenencia y los centroides  $v_i^{IFS} = \mu(v_i, \nu(v_i, \pi(v_i)))$  se calculan con las expresiones:

$$u_{ij} = \frac{1}{\sum_{r=1}^c \left( \frac{d^2(x_j^{IFS}, v_r^{IFS})}{d^2(x_j^{IFS}, v_i^{IFS})} \right)^{\frac{1}{m-1}}}, 1 \leq i \leq c, 1 \leq j \leq N \quad (2.19)$$

$$\mu(v_i) = \frac{\sum_{j=1}^N u_{ij}^m \mu(x_j)}{\sum_{j=1}^N u_{ij}^m}, 1 \leq i \leq c \quad (2.20)$$

$$\nu(v_i) = \frac{\sum_{j=1}^N u_{ij}^m \nu(x_j)}{\sum_{j=1}^N u_{ij}^m}, 1 \leq i \leq c \quad (2.21)$$

$$\pi(v_i) = \frac{\sum_{j=1}^N u_{ij}^m \pi(x_j)}{\sum_{j=1}^N u_{ij}^m}, 1 \leq i \leq c \quad (2.22)$$

Los pasos para el agrupamiento difuso intuitivo se resumen en el Algoritmo 2.1. Este algoritmo requiere de datos difusos intuicionistas, por lo cual, para transformar los datos de entrada  $X$  (en el caso de una imagen se consideran la intensidad de los píxeles) al dominio intuicionista, es necesario inicializar los parámetros propios del algoritmo.

Los parámetros del algoritmo son: el número de centroides  $c$ , que para este ejemplo se puede ver como el número de regiones deseadas; el fuzificador determina que tan superpuestos estarán los grupos y convencionalmente se fija  $m = 2$ ; la variable de precisión  $\varepsilon$ ; y el factor para la transformación al dominio intuitivo  $\lambda$ , los cuales se inicializan en el paso 1.

En el paso 2 se inicializa el vector de centroides  $V^c$  y la matriz de partición difusa  $U^{N \times c}$  aleatoriamente. En el paso 3, se inicia la etapa de agrupamiento a través de un bucle que tiene la condición de paro  $\|U_{ij}^{N+1} - U_{ij}^N\| < \varepsilon$ . La condición de paro utiliza una medida de precisión  $\varepsilon$ , para evaluar si la agrupación se considera satisfactoria. Esta medida de precisión se calcula utilizando el grado de pertenencia de una iteración a la siguiente, tomando el mayor de estos valores en todos los puntos de datos considerando todos los grupos. El valor de  $\varepsilon$  depende de la precisión requerida, por lo cual influye directamente en el número de iteraciones que deben ejecutarse antes de que finalice el algoritmo.

---

**Algoritmo 2.1:** Agrupamiento difuso intuitivo
 

---

**input:** Imagen  $X$   
**output:** Imagen segmentada  $Y$

- 1 Fijar  $c, m, \varepsilon, \lambda$  ;
- 2 Inicializar  $V^c, U^{N \times c}$  aleatoriamente ;
- 3  $u_{ij}^{(0)} \leftarrow \left[ \sum_{r=1}^c \left( \frac{d^2(x_i^{IFS}, v_j^{IFS(0)})}{d^2(x_i^{IFS}, v_r^{IFS(0)})} \right)^{\frac{1}{m-1}} \right]^{-1}$  ;
- 4  $l \leftarrow 1$ ;
- 5 **while**  $\|U_{ij}^l - U_{ij}^{l-1}\| < \varepsilon$  **do**
  - 6  $\mu(v_j^{(l)}) \leftarrow \sum_{i=1}^N (u_{ij}^m)^{(l-1)} \mu(x_i) / \sum_{i=1}^N (u_{ij}^m)^{(l-1)}$ ;
  - 7  $v(v_j^{(l)}) \leftarrow \sum_{i=1}^N (u_{ij}^m)^{(l-1)} v(x_i) / \sum_{i=1}^N (u_{ij}^m)^{(l-1)}$ ;
  - 8  $\pi(v_j^{(l)}) \leftarrow \sum_{i=1}^N (u_{ij}^m)^{(l-1)} \pi(x_i) / \sum_{i=1}^N (u_{ij}^m)^{(l-1)}$ ;
  - 9 actualizar  $U$ ;
  - 10 **if**  $\forall j, r, d^2(x_i^{IFS}, v_j^{IFS(l)}) > 0$  **then**
    - 11  $u_{ij}^{(l)} \leftarrow \left[ \sum_{r=1}^c \left( \frac{d^2(x_i^{IFS}, v_j^{IFS(l)})}{d^2(x_i^{IFS}, v_r^{IFS(l)})} \right)^{\frac{1}{m-1}} \right]^{-1}$  ;
  - 12 **end**
  - 13  $l \leftarrow l + 1$ ;
- 14 **end**
- 15 **return**  $Y$

---

En los pasos 4 a 7, se actualiza el vector de centroides  $V$ . Cada elemento del vector de centroides  $v_j$  esta en términos de conjuntos difusos intuitivos. Para calcularse se utilizan las expresiones (2.20), (2.21) y (2.22), donde se considera cada dato de entrada  $x_i$  y su grado de pertenencia  $u_{ij}$  al centroide  $v_j$ .

En los pasos 8 a 12, se actualiza la matriz  $U$ , donde se requiere de la información actualizada del vector de centroides  $V$  para calcular la similaridad con cada dato de entrada  $x_i$ . Estos pasos se repiten hasta que se cumpla la condición de paro. El resultado se guarda en la matriz  $U$ , que contiene el grado de pertenencia de cada dato  $x_i$  al centroide  $v_j$ .

### 2.4. Imagen de Resonancia Magnética Cerebral

Las imágenes médicas representan una herramienta importante en las ciencias de la salud ya que proveen información sobre estructuras fisiológicas específicas (órganos o tejidos). Estas imágenes, mediante métodos apropiados de instrumentación y recolección de datos, pueden reconstruirse principalmente en dos o tres dimensiones y luego mostrarse como conjuntos de datos multidimensionales (Dougherty, 2009).

Las técnicas más utilizadas son la tomografía computarizada (TC), la resonancia magnética (RM), la tomografía por emisión de positrones (PET), la tomografía computarizada por fotón único (SPECT), la ecografía y muchas otras (Chaira, 2015). Anteriormente, las imágenes se pensaban como una herramienta de diagnóstico para enfermedades, pero ahora también se usan para tratar, controlar y predecir enfermedades (Laal, 2013).

En comparación con las técnicas de diagnóstico antes mencionadas, la resonancia magnética tiene una ventaja sustancial sobre estas, ya que la RM no es invasiva y puede proporcionar información tomográfica con una alta resolución espacial y contraste de tejidos blandos (Cheng et al., 2013).

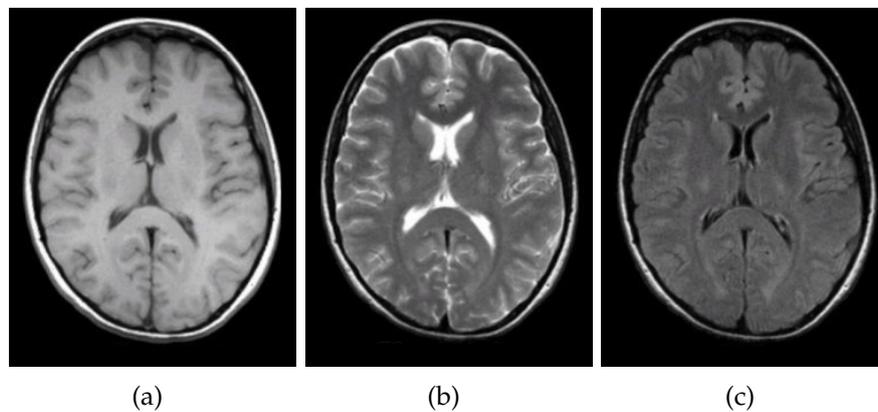
Una Imagen de Resonancia Magnética (IRM) puede definirse como una función  $I(i, j)$  en el espacio 2D o  $I(i, j, k)$  en el espacio 3D, donde  $i = 0, \dots, M - 1$ ,  $j = 1, \dots, N - 1$  y  $k = 0, \dots, D - 1$  denota coordenadas espaciales. Los valores (o amplitudes) de las funciones  $I(i, j)$  y  $I(i, j, k)$  son valores de intensidad y están representados típicamente por un valor de gris  $\{0, \dots, 255\}$ . Cada imagen consiste en un conjunto finito de elementos de imagen llamados píxeles en espacio 2D o vóxeles en espacio 3D.

Cada elemento de imagen se especifica de forma única por su valor de intensidad y sus coordenadas  $(i, j)$  para píxeles y  $(i, j, k)$  para vóxeles, donde  $i$  es el número de la fila de la imagen,  $j$  es el número de la columna de la imagen y  $k$  es el número de corte en una pila volumétrica (Despotović et al., 2015). La IRM cerebral permite explorar, medir y visualizar las estructuras anatómicas del cerebro, para analizar cambios, delinear regiones patológicas, permitiendo un progreso significativo en el estudio del cerebro.

## 2.4 Imagen de Resonancia Magnética Cerebral

La IRM cerebral es uno de los estándares de la nueva atención médica para enfermedades tales como: cáncer, enfermedades cardiovasculares, trauma y condiciones neurológicas (Laal, 2013).

Existen muchos tipos de técnicas de RM cerebral, entre las más comunes se pueden mencionar: las exploraciones ponderadas en T1, ponderadas en T2 y la secuencia Flair (*Fluid Attenuated Inversion Recovery*). En la Figura 2.3 se muestra la comparación de estas técnicas. Las imágenes ponderadas en T1 se producen utilizando tiempos TE y TR cortos. El contraste y el brillo de la imagen están determinados predominantemente por las propiedades T1 del tejido. Por el contrario, las imágenes ponderadas en T2 se producen utilizando tiempos TE y TR más largos.



**Figura 2.3:** Comparación de T1-Ponderado, T2-Ponderado y Flair en una RM cerebral.; (a) T1-Ponderado, (b) T2-Ponderado, (c) Flair (Preston, 2006).

La secuencia Flair es similar a una imagen ponderada en T2, excepto que los tiempos TE y TR son muy largos. Al hacerlo, las anomalías permanecen brillantes, pero el fluido normal del Líquido Ceforraquídeo (LCR) se atenúa y oscurece. Esta secuencia es muy sensible a la patología, permitiendo diferenciar entre LCR y una anomalía con mucho más facilidad (Dougherty, 2009) (Preston, 2006). En la Tabla 2.2 se muestran los tiempos en TR y TE más comunes en las secuencias de RM.

**Tabla 2.2:** Las secuencias de IRM más comunes y sus tiempos aproximados de TR y TE.

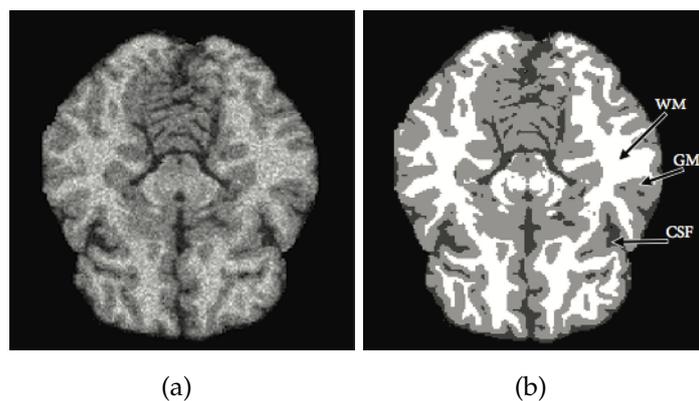
	TR(msec)	TE(msec)
T1-Ponderado (TR y TE corto)	500	14
T2-Ponderado (TR y TE largo)	4000	90
Flair (TR y TE muy largo)	9000	114

## 2.4 Imagen de Resonancia Magnética Cerebral

El análisis de imágenes de resonancia magnética cerebral abarca desde la percepción de la imagen hasta la interpretación de la imagen. Por lo cual se pueden mencionar los siguientes niveles o etapas de procesamiento. El nivel bajo (preprocesamiento) se ocupa del mejoramiento de la imagen a través de procesos como: modificación del contraste, reducción de ruido, realce de bordes, etc. En este nivel se utiliza únicamente la información de los píxeles. El siguiente nivel se refiere a las regiones de la imagen, donde se extraen las características de diferentes regiones u objetos.

El siguiente proceso es el procesamiento de alto nivel donde se considera la totalidad, partes de la imagen o serie de imágenes. Las imágenes están segmentadas para su clasificación, reconocimiento e interpretación. En el alto nivel, el conocimiento de los expertos está involucrado para tomar decisiones.

Los componentes fundamentales del análisis estructural de la RM cerebral incluyen la clasificación de los datos de la RM en tipos de tejido específicos y la identificación y descripción de estructuras anatómicas específicas. La clasificación significa asignar a cada elemento de la imagen una clase de tejido, donde las clases se definen de antemano. En el caso de la RM cerebral, los elementos de la imagen se clasifican generalmente en tres tipos de tejidos principales: materia blanca (WM), materia gris (GM) y líquido cefalorraquídeo (CSF) (Tamajka and Benesova, 2016). En la Figura 2.4, se muestra un ejemplo de la segmentación de una imagen de resonancia magnética cerebral, en los tres tejidos principales.



**Figura 2.4:** Tejidos principales en una RM cerebral; (a) imagen original, (b) imagen segmentada con los tres tejidos principales en una resonancia magnética cerebral: materia blanca (WM), materia gris (GM) y líquido cefalorraquídeo (CSF) (Despotović et al., 2015).

Uno de los principales desafíos del procesamiento de imágenes de resonancia magnética cerebral, es el desarrollo de algoritmos de segmentación de imágenes en 3D. Aunque en las últimas décadas se propusieron varios métodos sofisticados para el análisis automático de imágenes volumétricas, los rendimientos de casi todas las técnicas sufrieron limitaciones, como el aumento del tiempo y la carga de memoria necesaria para realizar la segmentación. Esto se debe principalmente por el aumento de la resolución espacial de los cortes individuales, así como por el número de cortes incluidos en una exploración de resonancia magnética típica (Fabijańska and Gołowski, 2016) (Verma et al., 2013).

## 2.5. Discusión

Una de las principales dificultades del procesamiento de imágenes médicas es el desarrollo de algoritmos de segmentación de imágenes en 3D. Los avances en la adquisición de imágenes son un obstáculo debido al aumento constante en la resolución de las imágenes volumétricas adquiridas por los escáneres de tomografía computarizada (TC) y resonancia magnética (IRM); por lo cual se necesitan métodos sofisticados para mejorar el rendimiento de las aplicaciones del procesamiento de imágenes (Fabijańska and Gołowski, 2016), (Despres and Jia, 2017).

La sobresegmentación con algoritmos para generar superpíxeles son una alternativa eficiente para reducir la complejidad de una imagen, debido a esto, en los últimos años se publicaron varios enfoques que buscan mejorar la sobresegmentación con superpíxeles. Una de las ventajas de estos algoritmos es que pueden considerar los superpíxeles como las unidades de procesamiento básicas para construir un gráfico 3D, mejorando los tiempos de cálculo y el requisito de memoria (Tian et al., 2016).

Los algoritmos para generar superpíxeles que se estudiaron en la Sección 2.2.2, utilizan diferentes enfoques como agrupamiento, grafos, optimización energética, etc. En estos métodos, la asignación de los píxeles a un determinado superpíxel se convierte en una asignación totalmente binaria, lo cual puede causar una mala adaptación de los superpíxeles a los bordes de la imagen. El algoritmo de Agrupamiento Difuso Intuitivo (IFCM), posee características que le permiten manejar eficientemente la incertidumbre. Por lo cual este algoritmo, podría mejorar la extracción de superpíxeles, con mayor adherencia a los bordes naturales de la imagen.

# Capítulo 3

## Estado del Arte

En este Capítulo se presenta una breve descripción de los trabajos más relevantes relacionados con la segmentación de imágenes utilizando métodos de extracción de superpíxeles, así como de algunas técnicas de agrupación.

### 3.1. Antecedentes

Se realizó una revisión de los proyectos de investigación desarrollados en CENIDET. Aunque no se encontraron investigaciones de sobresegmentación utilizando superpíxeles, existen trabajos en el área de agrupamiento y segmentación de imágenes. Como principal aporte de estos trabajos, se considera el análisis de las metodologías empleadas para la segmentación de imágenes utilizando diferentes enfoques.

#### **Metodología para la estructuración y uso de conocimiento en segmentación de imágenes digitales (Cervantes, 2006)**

En este trabajo se propuso una metodología para la segmentación de imágenes en donde el conocimiento relacionado se encuentra separado de los algoritmos necesarios para el procesamiento de la imagen. Esta metodología basada en diferentes técnicas se representaron con *frames* y reglas de producción. También se diseñó e implementó un lenguaje (LBFR) para representación de conocimiento el cual se basa en una combinación de *frames* y reglas. Se desarrolló un prototipo para un sistema de segmentación de imágenes basado en la arquitectura propuesta. También se desarrollaron las bases de conocimiento para segmentar puntos, líneas, círculos y elipses.

#### **Caracterización de texturas naturales (Avellaneda, 2009)**

En este trabajo se propuso una metodología para realizar el proceso de segmentación y reconocimiento. La metodología utilizó las reglas generadas por la herramienta *See5* en la etapa de entrenamiento, el algoritmo *Fuzzy C-Means* y una caracterización de 5 clases de textura (árboles, arena, cielo, montañas y nubes). Para la caracterización se utilizaron los siguientes descriptores: estadísticos de primer orden en los espacios de color HSI y RGB, estadísticos de segundo orden utilizando la matriz de Co-ocurrencia, momentos invariantes y descriptores obtenidos en el dominio de la frecuencia.

La principal aportación de esta investigación es la metodología que se propuso para realizar al mismo tiempo la segmentación y el reconocimiento de las regiones de textura presentes en una imagen de escenas naturales. Los resultados obtenidos mostraron que dicha metodología permitió que la segmentación y el reconocimiento de las texturas fueran invariante a traslación y ante algunos cambios de rotación y escala.

#### **Sintonización de una Red Totalmente Conectada para Segmentación de Dos Clases de Objetos en Imágenes (Suárez, 2018)**

En este trabajo se realizó la sintonización de una Red Neuronal Convolutiva (Convolutional Neural Network, CNN) para segmentar dos clases de objetos del repositorio BSDS500 (Berkeley Segmentation Data Set). Se experimentó con dos modelos de Redes Neuronales Convolutivas, FCN-Alexnet y FCN-8s. En la etapa de experimentación se configuraron diferentes hiperparámetros (valor que no es aprendido en el entrenamiento de la red), de los cuales, los más importantes son el número de capas convolutivas y la tasa de aprendizaje.

Otros hiperparámetros como el número de iteraciones de prueba, número máximo de iteraciones no influyen significativamente. También fueron probados los diferentes métodos de optimización existentes en el entorno de trabajo llamado Cafe, siendo el Gradiente Descendente Estocástico (SGD) el método con mejores resultados. El modelo de CNN más eficiente fue FCN-8s, ya que obtuvo los mejores resultados con respecto a las métricas utilizadas: Índice Probabilístico Rand, Variación de la Información y Error de Consistencia Global.

Para agilizar el procesamiento de las imágenes, los modelos de CCN antes mencionados, fueron implementados en una Unidad de Procesamiento Gráfico (Graphics Processing Unit, GPU) NVIDIA.

#### **Segmentación no Paramétrica de Tejidos Cerebrales Mediante una Arquitectura Paralela de Redes Neuronales Convolucionales (Morales, 2018)**

En este trabajo se desarrolló e implementó un método de segmentación no paramétrica de tejidos cerebrales, basada en una arquitectura paralela de Redes Neuronales Convolucionales (CNN). El modelo U-Net fué utilizado para la adaptación de una arquitectura paralela. La arquitectura implementada se compone de cuatro Redes Neuronales Convolucionales, a las cuales les fue realizado un ajuste paramétrico, para su entrenamiento individual. Cada uno de los modelos que componen a la arquitectura paralela, lograron realizar la segmentación binaria de un tejido cerebral; la segmentación completa de la imagen evaluada se obtuvo mediante la unificación de los tejidos.

Los repositorios de imágenes utilizadas fueron BrainWeb para entrenamiento y BraTS 2017 para validación y pruebas. A los estudios de resonancia magnética utilizados para entrenamiento se les aplicó un pre-procesamiento para la supresión de tejidos no blandos, seguido de una transformación a formato TIF (Tagged Image File Format) para su uso en los framework Keras y TensorFlow; finalmente, se realizó un aumento de información para el entrenamiento de los modelos de CNN utilizados.

Se utilizó una muestra de estudios e imágenes del repositorio BraTS 2017 para pruebas en las cuales se logró la detección de las regiones que componen a la imagen de resonancia magnética cerebral. Ésto sin realizar una inicialización de parámetros o indicación del número de regiones esperadas, aún con la variación de regiones en las imágenes cerebrales. La metodología tiene un buen rendimiento de segmentación con respecto a las métricas: Jaccard, Coeficiente de similitud Dice (Dice Similarity Coefficient) y Área bajo la curva (AUC). En promedio los resultados obtenidos con el repositorio BrainWeb fueron de 0.9653 y en cuanto al repositorio BraTS 2017 se obtuvo un 0.9692.

## 3.2. Trabajos relacionados

La investigación relacionada con los algoritmos superpíxel ha tenido un crecimiento sustancial durante los últimos años. Por lo cual, la literatura sobre este tema es extensa. En esta sección se revisaron los trabajos más relevantes encontrados en la literatura relacionados con la sobsegmentación de superpíxeles, así como algunas de sus aplicaciones en el procesamiento de imágenes. También se presenta una breve descripción de los algoritmos que se utilizan para la evaluación comparativa del algoritmo propuesto.

### **TurboPixels: Superpíxeles rápidos utilizando flujos geométricos (Levinshtein et al., 2009)**

El algoritmo TurboPixels (TP), dilata progresivamente un conjunto de ubicaciones de semillas utilizando un flujo geométrico basado en conjuntos de niveles. El flujo geométrico se basó en los gradientes de imagen locales, con el objetivo de distribuir regularmente superpíxeles en el plano de la imagen. De esta manera el flujo geométrico evolucionó para obtener límites de superpíxeles. Esta evolución se obtiene de acuerdo a la función de inserción que se denota:

$$\Psi_t = -S \|\nabla \Psi\| \quad (3.1)$$

donde,  $\Psi$  es la distancia euclidiana sobre el plano de la imagen. La distancia de un píxel es positiva si está en la región asignada y negativa si no lo está. Para inicializar el método se tomó  $K$  semillas circulares aproximadamente iguales a  $\sqrt{N/K}$ , donde  $N$  es el número total de píxeles en la imagen y  $K$  es el número de superpíxeles requeridos. El radio inicial es de un píxel, y evoluciona de acuerdo a la discretización de primer orden en el tiempo de (3.1), el cual se denota como:

$$\Psi^{n+1} = \Psi^n - S_I S_B \|\nabla \Psi^n\| \Delta t \quad (3.2)$$

donde,  $\Delta t$  corresponde a un “paso de tiempo”  $t$  en la evolución del límite,  $S_I$  denota la velocidad de la imagen y  $S_B$  denota la velocidad del límite. La evolución del límite se controló mediante el producto de estas dos velocidades.

El término  $S_I$  depende de la estructura de la imagen local y la geometría de los superpíxeles en cada punto del límite,  $S_I$  se calcula utilizando la siguiente ecuación:

$$S_I(x,y) = [1 - \alpha k(x,y)]\phi(x,y) - \beta[N(x,y) \cdot \nabla\phi(x,y)] \quad (3.3)$$

donde,  $\phi(x,y)$  es la función de afinidad local, calculada para cada píxel en el plano de la imagen;  $k(x,y)$  denota una función de curvatura, que expresa la curvatura del límite en el punto  $(x,y)$  y suaviza el límite evolutivo;  $\alpha$  y  $\beta$  son parámetros de balanceo que pesan la contribución del término de curvatura;  $N(x,y)$  es una función que calcula la normal exterior del conjunto de nivel cero en un punto  $(x,y)$  y  $\phi(x,y)$ , es la función de afinidad local. El término  $S_B$ , depende de la proximidad del punto del límite a otros súper píxeles.

$S_B(x,y) = 0$  si y solo si  $(x,y)$  está en el esqueleto homotópico 2D de la región. Esta formulación permitió que el límite de cada superpíxel se guíe completamente por la imagen subyacente hasta que se acerque mucho a otro límite de superpíxel. La ecuación (3.2) se aplicó hasta que cualquier punto en el límite evolutivo alcance el borde de la banda estrecha. **TP** se basa en algoritmos de complejidad variable, pero en la práctica, tiene un comportamiento lineal. El proceso se resume en el algoritmo 3.1.

---

### Algoritmo 3.1: TurboPixel

---

**input:** Imagen  $I$ , Número de superpíxeles  $K$ .  
**output:** Imagen sobresegmentada  $B$ .

- 1 colocar  $K$  semillas en una cuadrícula regular en la imagen  $I$ ;
- 2 asignar píxeles iniciales  $\sum_{x,y}[\Psi^0(x,y) \geq 0]$ ;
- 3 calcular  $\phi(x,y)$ ;
- 4  $n \leftarrow 0$ ;
- 5 **while**  $\sum_{x,y}[\Psi^0(x,y)] > \Delta_n$  **do**
- 6 calcular  $S_I$ ;
- 7 calcular  $S_B$ ;
- 8  $S \leftarrow S_I S_B$ ;
- 9 calcular  $\Psi^{n+1}$  con respecto a  $\Psi^n$ ;
- 10  $n \leftarrow n + 1$ ;
- 11 calcular  $\leftarrow \sum_{x,y}[\Psi^n(x,y) \geq 0]$ ;
- 12  $B \leftarrow \Psi^n$
- 13 **end**
- 14 **return**  $B$

---

**Superpíxeles SLIC (Achanta et al., 2010)**

El algoritmo SLIC (*Simple Linear Iterative Clustering*) es una adaptación del algoritmo *k-means* para generar superpíxeles. Aunque el algoritmo tiene varios años desde que se dio a conocer, se sigue aplicado en trabajos recientes. En la literatura se pueden encontrar mejoras al algoritmo, así como metodologías basadas en SLIC para diferentes aplicaciones. Esto se debe a que el algoritmo es fácil de implementar y ofrece ventajas como: reducción el costo computacional y una alta velocidad para generar los superpíxeles. Los pasos para la implementación de este método se resumen en el Algoritmo 3.2.

**Algoritmo 3.2: SLIC**


---

```

input: imagen  $I$ , número de superpíxeles  $K$ 
output: segmentación de superpíxel  $R$ 
1 inicializar  $C_k \leftarrow [l_k, a_k, b_k, x_k, y_k]^T$  en cuadrículas regulares de tamaño  $S$ ;
2 mover  $C_k$  a la posición de gradiente más bajo en un vecindario de  $3 \times 3$ .
3 repeat
4   for  $k = 1$  to  $K$  do
5     for píxel  $n_i$  en una región  $2S \times 2S$  alrededor de  $C_k$  do
6       if  $d(n_i, R_k) < d(n_i, R_{R(n_i)})$  then
7          $R_k \leftarrow R_k \cup \{n_i\}$ ;
8          $R_{R(n_i)} \leftarrow R_{R(n_i)} - \{n_i\}$ ;
9       end
10    end
11  end
12 until  $E \leq umbral$ ;
13 return  $R$ 

```

---

Para generar los superpíxeles, SLIC comienza transformando la imagen al espacio de color CIELAB. SLIC agrupa píxeles utilizando el vector de características  $C_k = [l_k, a_k, b_k, x_k, y_k]$ ,  $k = 1, 2, \dots, K$ , que consiste en los valores de color del espacio CIELAB y las coordenadas  $(x_k, y_k)$  de los píxeles en la imagen. La imagen se divide en cuadrículas regulares de tamaño  $S = \sqrt{N/K}$ ; donde  $N$  es el número total de píxeles en la imagen y  $K$  es el número de superpíxeles requeridos.

Para disminuir la posibilidad de que el centro del grupo sea un borde o un punto de ruido, se calcula el gradiente mínimo en un vecindario  $3 \times 3$ . El cálculo se hace de la siguiente forma:

$$G(x,y) = ||I(x+1,y) - I(x-1,y)||^2 + ||I(x,y+1) - I(x,y-1)||^2 \quad (3.4)$$

donde  $I(x,y)$  es el vector de características  $C_k$  correspondiente al píxel en la posición  $(x,y)$ , y  $||\cdot||$  es la norma  $L_2$ . Para aumentar la velocidad de cálculo, los píxeles similares se buscan en la región  $2S \times 2S$  de los puntos centrales del superpíxel en lugar de la imagen completa. Cada píxel de la imagen está asociado con el centro del grupo más cercano, cuya área de búsqueda se superpone a este píxel. La similitud entre el píxel y el centro del superpíxel se calcula combinando la distancia de color y la distancia espacial utilizando las siguientes expresiones:

$$\begin{aligned} d_c &= \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \\ d_s &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\ D_s &= \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 + w^2} \end{aligned} \quad (3.5)$$

donde  $d_c$  es la distancia de color entre píxeles,  $d_s$  la distancia espacial entre píxeles,  $S$  la distancia entre los puntos centrales,  $w$  en un parámetro de equilibrio para evaluar las proporciones de color y distancia. El parámetro  $w$  proporcionando control sobre la compacidad de los bloques superpíxel después del agrupamiento. Una vez que todos los píxeles están asociados con el centro del grupo más cercano, se calcula un nuevo centro promediando todos los píxeles que pertenecen al grupo. Luego este proceso se repite iterativamente hasta la convergencia.

#### Superpíxeles con optimización booleana (Zhang et al., 2011)

Este método se basa en un algoritmo de eliminación recursiva y la sustitución inversa para optimizar las funciones de energía pseudo-booleas. Este algoritmo toma la segmentación de superpíxel como un problema de asignación de etiquetas múltiples. Inicialmente, la imagen de entrada está cubierta por tiras horizontales que se superponen a medias. Cada píxel tiene la posibilidad de ser asignado a una de dos franjas latentes alternativas. Entonces, la tarea es decidir a qué tiras pertenecen los píxeles. Este proceso se formuló como un problema de etiquetado binario en *Markov Random Fields* (MRFs).

La pertenencia de cada píxel con respecto a otros dos conjuntos de tiras latentes  $\{H_i\}$  y  $\{V_i\}$ , se define como

- If  $p \in H_{2i}$  y  $label(p) = 0$ , then  $p \in H'_{2i}$
- If  $p \in H_{2i+1}$  y  $label(p) = 1$ , then  $p \in H'_{2i+1}$

El algoritmo PB (*Pseudo-Boolean*) logra la sobresegmentación mediante la asignación binaria de la etiqueta  $\{0,1\}$  a cada píxel de la imagen. La función objetivo se compone de dos funciones pseudo-booleanas que pueden optimizarse mediante el algoritmo de eliminación antes mencionado. PB genera superpíxeles que son regulares en tamaño y forma sin la necesidad de un parámetro adicional como en los algoritmos SLIC y LSC. La velocidad de PB es independiente de la cantidad de superpíxeles. PB se resume en el Algoritmo 3.3.

---

### Algoritmo 3.3: PB

---

**input:** imagen  $I$ , número de superpíxeles  $K$   
**output:** segmentación de superpíxel  $R$

- 1 cubrir la imagen con tiras horizontales superpuestas  $\{H_i\}$ ;
- 2 asignar etiquetas binarias a cada píxel  $i$  para minimizar el costo total, que generan  $\{H'_i\}$ ;
- 3 implementar los pasos 1 y 2 en las tiras verticales  $\{V_i\}$ , que generan  $\{V'_i\}$ ;
- 4 **for**  $i,j$  **to**  $N,M$  **do**
- 5 **if**  $H_i \cap V_j \neq \emptyset$  **then**
- 6  $R_k \leftarrow H_i \cap V_j$ ;
- 7 **end**
- 8 **end**
- 9 **return**  $R$

---

### Crecimiento de región para un agrupamiento eficiente basado en Eikonal (Buysens et al., 2014)

El algoritmo ERGC se basa en la ecuación *Eikonal* para generar superpíxeles. El enfoque de este algoritmo es un esquema de crecimiento de la región, donde el mapa potencial utilizado es dinámico y evoluciona durante la difusión. El algoritmo completo consta de 3 pasos:

1. La difusión de las semillas deseadas se da inicialmente con el 90% colocándolas en una cuadrícula.

2. Se mejora la sobresegmentación agregando el 10% restante de las semillas.
3. Se itera el procedimiento de semillas hasta que se alcance el criterio de detención.

También se propusieron pasos de refinamiento para mejorar el costo computacional del método. ERGC es fácil de usar, de forma predeterminada, el único parámetro es el número deseado de superpíxeles, también ofrece control sobre la compacidad de los superpíxeles. ERGC fue introducido para imágenes 2D, pero puede extenderse a volúmenes 3D para producir supervóxeles con modificaciones menores del algoritmo. ERGC se resume en el Algoritmo 3.4.

---

### Algoritmo 3.4: ERGC

---

**input:** imagen  $I$ , el grafo  $G$ , número de superpíxeles  $K$   
**output:** segmentación de superpíxel  $R$

- 1  $V_0 \leftarrow \arg \min_{v \in V} (\delta_v / N_v)$ ;
- 2 Resolver  $\|(\nabla_w - f)(u)\|_p = \begin{cases} P(u) \forall u \in V \\ f(u) = \phi(u) \forall u \in V_0 \end{cases}$
- 3 Guardar distancias geodésicas  $U_0 \leftarrow d(0)$ ;
- 4  $it \leftarrow 1$ ;
- 5 **while**  $it < n$  **do**
- 6  $V_0 \leftarrow V_0 \cup \arg \max_{v \in V} (U_{it}(v))$ ;
- 7 Repetir el paso 2 con  $V_0$ ;
- 8  $it \leftarrow it + 1$ ;
- 9  $U_{it} \leftarrow d(it)$ ;
- 10 **end**
- 11 **return**  $R$

---

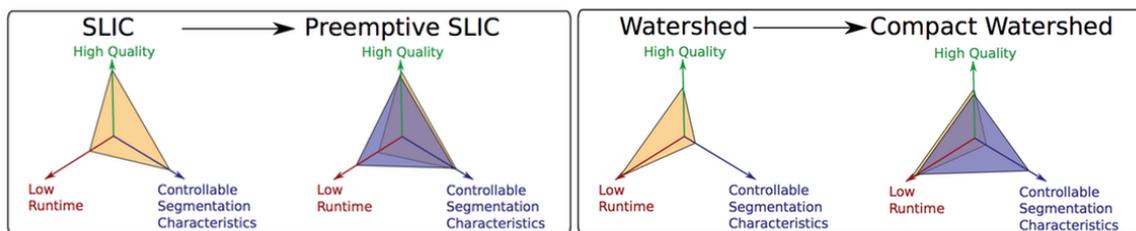
### Píxeles de Agua: Superpíxeles basados en la transformación de cuencas (Machairas et al., 2014)

En este trabajo se proporcionó un esquema para aplicar la transformación de cuencas (*Watershed transformation*) para la generación de superpíxeles. Para lograr un equilibrio entre la regularidad del superpíxel y la adherencia a los límites del objeto se utilizó una regulación espacial del gradiente a través del parámetro  $k$ . La segmentación utilizando cuencas se basa en dos principios: la definición de los marcadores, a partir de los cuales comienza la inundación (*flooding*), y la definición de un gradiente.

El cálculo de la cuenca hidrográfica produce de forma natural un gráfico de adyacencia de la región. Este gráfico se puede usar para construir superpíxeles con diferentes resoluciones o para fusionarlos y obtener una segmentación de alto nivel. La implementación de este método depende de dos parámetros principales, los pasos de la cuadrícula regular  $\sigma$  y el parámetro de regularización espacial  $k$ . Sin embargo, se deben ajustar dos parámetros "ocultos", vinculados a la selección mínima: los márgenes de las células y el parámetro del área de preprocesamiento.

#### Cuenca compacta y SLIC preferente: Sobre la mejora de las compensaciones de los algoritmos de segmentación de superpíxeles (Neubert and Protzel, 2014)

En este trabajo se presentaron dos nuevos algoritmos de segmentación de superpíxeles, basados en algoritmos existentes, logrando un mejor equilibrio entre la alta calidad, el bajo tiempo de ejecución y las características de segmentación controlables. En la Figura 3.1 se muestra la representación del equilibrio entre estas características.



**Figura 3.1:** Representación del equilibrio en los algoritmos SLIC y CW (Neubert *et al.*, 2014).

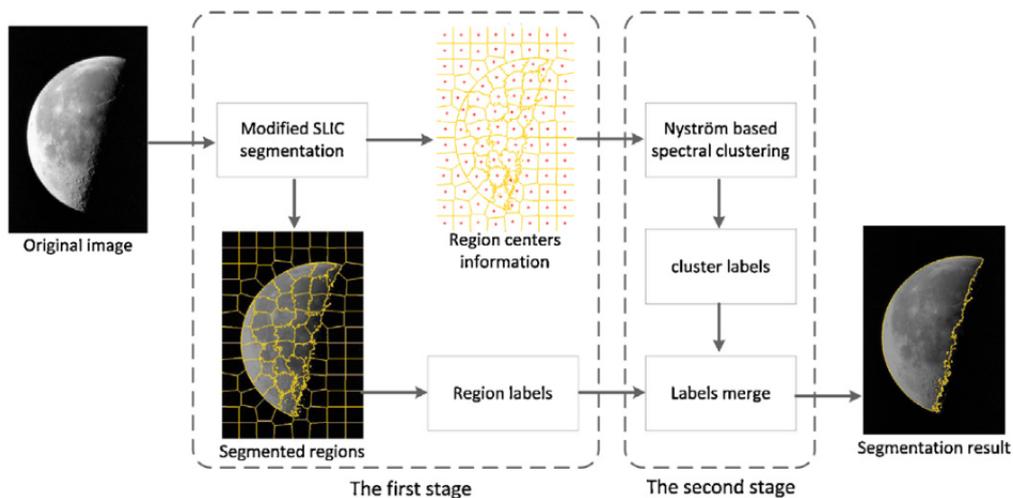
El primer algoritmo es SLIC-preferente, es una versión más rápida de SLIC. Se propuso utilizar un criterio de terminación local para cada grupo en vez de el criterio de terminación global que utiliza el algoritmo convencional. Esto con el fin de evitar volver a visitar los grupos y las áreas de la imagen sin ningún cambio importante desde la última iteración.

La optimización de código disminuyó el tiempo de ejecución aproximadamente 100ms para números de segmento de superpíxel típicos, hasta aproximadamente 50ms. En combinación con el criterio de terminación local propuesto, el algoritmo SLIC-preferente resultante mejoró la velocidad de ejecución (30 Hz para el tamaño de imagen  $481 \times 321$ ) en una computadora de escritorio estándar.

El segundo algoritmo que se propuso es cuenca compacta basado en la segmentación de *Seeded Watershed* donde se introdujo una restricción de compacidad controlable para la segmentación, basada en una medida de distancia entre el píxel y la semilla del segmento. Esto resultó en una restricción en el tamaño y el alargamiento de los segmentos y, por lo tanto, favorece la creación de segmentos compactos.

**Segmentación de imágenes utilizando agrupación espectral modificada y basada en SLIC y Nyström (Bai et al., 2014)**

En este trabajo se propuso una metodología eficiente de segmentación de imágenes, que utiliza superpíxeles y la agrupación espectral basada en Nyström. La metodología tiene dos etapas principales, como se muestra en la Figura 3.2. En la primera etapa se aplicó la segmentación de imágenes utilizando el algoritmo SLIC. Para aplicar SLIC con imágenes en escala de grises, se realizó una modificación del algoritmo convencional en la definición de distancia.



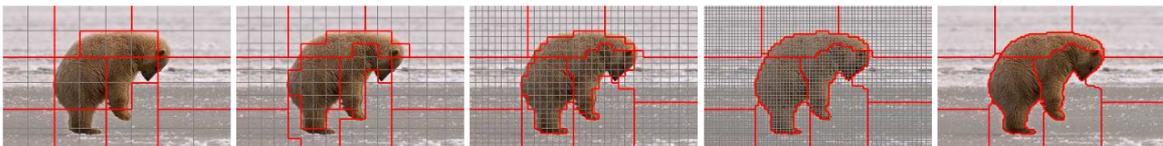
**Figura 3.2:** Segmentación de imágenes utilizando SLIC y agrupación espectral (Bai et al.,2014).

La utilización de SLIC permitió ahorrar memoria y mejorar la calidad del resultado de segmentación. El algoritmo modificado de SLIC divide el objeto y el fondo en la imagen en diferentes regiones. En la segunda etapa, las intensidades de las regiones segmentadas se tomaron como las características de las regiones correspondientes.

Posteriormente, se realizó la agrupación en regiones espectrales basados en el método Nyström. Por último, se realizó la asignación de los píxeles a las regiones correspondientes, produciendo la segmentación final. El método propuesto logró reducir significativamente el requisito de memoria haciendo que sea eficaz para una segmentación robusta.

### Segmentación de preservación topológica gruesa a fina en tiempo real (Yao et al., 2015)

En este trabajo se propuso el algoritmo para generar superpíxeles ETPS (*Efficient Topology Preserving Segmentation*), que aborda el problema de la segmentación no supervisada en forma de superpíxeles. Se formuló el problema de sobresegmentación con un enfoque de energía de campos aleatorios de *Markov* y una función objetivo similar al algoritmo *k-means*. Además, se estableció un límite de tamaño de las regiones para evitar superpíxeles pequeños. En la Figura 3.3 se ilustra el proceso de generación de superpíxeles con el método propuesto.



**Figura 3.3:** Extracción de superpíxeles con el algoritmo ETPS; Las actualizaciones de nivel de límite de grueso a fino comienzan en el nivel aproximado (izquierda) y continúan hasta el nivel más fino de forma iterativa. El resultado final se muestra a la derecha (Yao *et al.*, 2015).

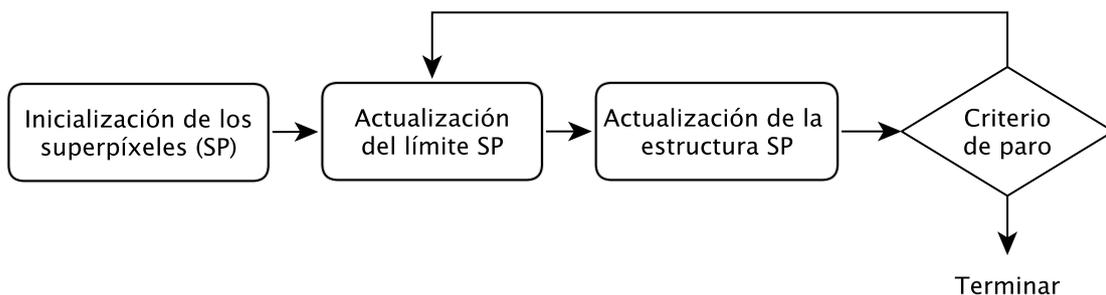
El algoritmo funciona de la siguiente manera: comienza inicializando los superpíxeles en una cuadrícula regular y calculando las estimaciones iniciales de la posición y el color para cada superpíxel, posteriormente se optimizan de manera eficiente todos los niveles (desde grueso hasta fino). Esto permite hacer movimientos rápidos (en niveles más bajos) alcanzando un óptimo local mucho mejor que el establecido en la función objetivo, lo cual hace que el algoritmo resultante sea rápido.

#### Convexidad limitada para la extracción eficiente de superpíxeles y supervóxeles (Tasli et al., 2015)

En este trabajo se presentó un método eficiente de extracción de superpíxeles y supervóxeles con contribuciones en términos de eficiencia computacional y rendimiento de segmentación. En la técnica que propuso, los superpíxeles y los supervóxeles se actualizan iterativamente a través de los píxeles de los límites en función del color y la similitud espacial. La idea de adaptación de límites y la selección de la función de energía son las dos contribuciones principales de este método, que permiten una implementación eficiente.

Se evaluaron diferentes combinaciones de funciones de energía y se estudió los efectos de utilizar diferentes espacios de color y métricas de distancia, con el fin de optimizar el rendimiento de segmentación y la carga computacional del método. El espacio de color LAB mostró un rendimiento superior en términos de adaptación de límites en comparación con RGB. Del mismo modo, la utilización de la distancia geodésica mejoró el rendimiento de la segmentación en comparación con la métrica euclidiana.

El método logró una fuerte adaptación en los límites del objeto en los superpíxeles y supervóxeles generados. El flujo algorítmico del método se puede explorar en cuatro pasos principales: (1) inicialización de los superpíxeles (SP); (2) actualización del límite SP; (3) actualización de estructura SP; (4) terminación. Estos pasos se ilustran en la Figura 3.4.



**Figura 3.4:** Flujo del algoritmo para la generación de superpíxeles (Tasli et al., 2015).

- Inicialización de superpíxel: en el primer paso, la imagen se divide en regiones de igual tamaño de acuerdo con el número de superpíxeles deseados.
- Actualización de límite: Se realiza una búsqueda codiciosa en los píxeles de límite. Durante la adaptación del límite, se minimiza la función de costo que relaciona la similitud de los píxeles con los candidatos superpíxeles correspondientes.
- Actualización de la estructura: El modelo de superpíxel se vuelve a calcular en función de los píxeles de límite eliminados o fusionados. Esta actualización proporciona grupos de píxeles para adaptar los cambios a lo largo de los límites y convergen en modelos superpíxel compactos en términos de similitud de píxeles. Los pasos de actualización de límite y estructura se repiten varias veces hasta que se cumplen los criterios de detención.

### **Segmentación de superpíxeles con agrupamiento espectral lineal (Li and Chen, 2015)**

En este trabajo se propuso el algoritmo LSC (*Linear Spectral Clustering*) para generar superpíxeles. Para obtener una sobresegmentación eficiente, se combinaron diferentes técnicas como los cortes normalizados, el algoritmo *k-means* y la agrupación espectral. Este método adopta una formulación de cortes normalizados utilizando una métrica de similitud que mide el color y la proximidad espacial entre los píxeles de la imagen.

Para mejorar el rendimiento, el algoritmo optimiza la función de costo de los cortes normalizados aplicando de forma iterativa agrupaciones simples de *k-means* en el espacio de características. Esto fue posible ya que al ponderar adecuadamente cada punto en este espacio de características, las funciones objetivos del algoritmo *k-means* y los cortes normalizados comparten el mismo punto óptimo. LSC es de complejidad computacional lineal y de alta eficiencia en memoria y es capaz de preservar las propiedades globales de las imágenes. En el Algoritmo 3.5 se resume el proceso.

**Algoritmo 3.5: LSC**


---

**input:** imagen  $I$ , número de superpíxeles  $k$   
**output:** segmentación de superpíxel  $L$

- 1  $\phi(p) \leftarrow [l_p, \alpha_p, \beta_b, x_p, y_p]$ ;
- 2 dividir la imagen en  $k$  regiones en intervalos fijos  $v_x$  y  $v_y$ ;
- 3  $L(p) \leftarrow 0$ ;
- 4  $D(p) \leftarrow \infty$ ;
- 5 **repeat**
- 6     **for**  $m_k, c_k$  **do**
- 7         **for**  $p_i$  en un vecindario  $\tau v_x \times \tau v_y$  al rededor de  $c_k$  **do**
- 8              $D(\phi(p), m_k)$
- 9             **if**  $D < d(p)$  **then**
- 10                  $d(p) \leftarrow D$ ;
- 11                  $L(p) \leftarrow k$ ;
- 12             **end**
- 13         **end**
- 14     **end**
- 15     actualizar  $m_k$  y  $c_k$  para todos grupos;
- 16 **until**  $m_k$  de todos los grupos converge;
- 17 fusionar pequeños superpíxeles con sus vecinos;
- 18 **return**  $L$

---

**gSLICr: superpíxeles SLIC a más de 250Hz (Ren et al., 2015)**

En este trabajo se desarrolló una implementación paralela del algoritmo SLIC con GPUs, utilizando el marco NVIDIA CUDA. La implementación de la GPU se realizó en 5 etapas, las cuales se describen a continuación.

- Conversión de espacio de imagen: En esta etapa, la imagen de entrada RGB se convierte a Cielab, utilizando un hilo por cada píxel.
- Inicialización del centro de agrupación: se utilizó un hilo por centro de agrupación para inicializar el mapa superpíxel.
- Encontrar las asociaciones de grupos: cada píxel de la imagen determina cuál es el grupo más cercano utilizando la distancia  $5D$ . Esto se realizó utilizando un hilo por píxel.
- Actualización del centro de grupo: se actualiza cada centro del grupo utilizando los píxeles asignados. Este proceso se realizó en dos núcleos separados.

- Aplicación de la conectividad: los píxeles perdidos se eliminan con dos llamadas de un hilo por píxel del mismo núcleo. Estos procesos solicitan el cambio de etiqueta de un píxel.

Este enfoque logró aceleraciones de hasta 83 veces más rápido en comparación a la implementación secuencial estándar. La implementación está diseñada para producir el mismo resultado que la implementación secuencial de SLIC, por lo que los resultados que se obtuvieron son idénticos en términos cualitativos y cuantitativos.

### **Algoritmo de agrupamiento rápido de particionamiento y fusión basado en la extensión (Ghanem et al., 2015)**

En este trabajo se propuso un algoritmo de agrupamiento rápido, denominado *Dimension-based Partitioning and Merging* (DPM). Para describir el proceso de agrupación por el algoritmo DPM, se pueden mencionar tres etapas principales. En la primera etapa, los datos fueron divididos en pequeños volúmenes densos durante el procesamiento sucesivo de las dimensiones del conjunto de datos. Luego, en la segunda etapa, el ruido fue filtrado utilizando densidades dimensionales de las particiones generadas. Finalmente, en la tercera etapa, el proceso de fusión se utilizó para construir los grupos basados en muestras de datos de partición.

DPM logró minimizar en gran medida la sobrecarga de las operaciones de agrupamiento, lo que mejora su velocidad en el agrupamiento. DPM tiene tres parámetros de control que son  $t$  como el número de divisiones de dimensión,  $n_{thr}$  como umbral de densidad dimensional de partición y  $p_{thr}$  como la sensibilidad de detección de picos de histograma. DPM fue capaz de detectar el número de grupos automáticamente en función de estos tres parámetros de control.

### **SLIC Superpíxel robusto al ruido para imágenes naturales (Dong and Zhou, 2016)**

En este trabajo se propusieron tres medidas para mejorar la robustez de SLIC contra el ruido : 1) Una nueva medición de distancia de intensidad de píxel, que se diseñó considerando explícitamente la varianza de ruido dentro del grupo; 2) La medición de la distancia espacial se mejoró explotando la variación de las ubicaciones de píxeles en un grupo.

3) Se propuso un estimador de ruido para actualizar los centros de grupo excluyendo los posibles valores atípicos causados por el ruido. La nueva medición de distancia de color incorpora una matriz  $3 \times 3$  donde la diagonal principal está compuesta por la desviación estándar para cada dimensión del vector de intensidad de color  $[l, a, b]^T$  que pertenece al  $i$ -ésimo grupo.

Para hacer una estimación robusta, se excluyeron los valores atípicos restringiendo el rango de píxeles aumentados. El rendimiento de segmentación del método propuesto es comparable con el algoritmo SLIC convencional para los casos limpios. Sin embargo, para imágenes ruidosas, SLIC no puede agrupar correctamente las regiones perceptivamente significativas en superpíxeles, es decir, no se adhiere a los bordes de la imagen.

### **Aceleración basada en submuestreo de SLIC para la segmentación de superpíxeles (Choi and Oh, 2016)**

En este trabajo se propuso un algoritmo de segmentación de superpíxeles acelerado basado en SLIC, donde el número de segmentos candidatos para cada píxel se reduce de manera efectiva al explotar la alta redundancia espacial dentro de la imagen. Los autores consideraron la redundancia interpíxel, debido a que las imágenes generalmente se caracterizan por grandes regiones de valores de píxeles constantes o casi constantes, lo cual permite una considerable correlación espacial entre los píxeles adyacentes.

En el contexto de la segmentación de superpíxeles, la redundancia interpíxel implica que si un píxel  $i$  está asociado con un determinado grupo, sus píxeles vecinos también tienden a pertenecer al grupo. El enfoque de este algoritmo es explotar la redundancia interpíxel a través de un método de submuestreo para mejorar el paso de asignación de grupos del algoritmo SLIC convencional. El algoritmo funciona casi cinco veces más rápido que el algoritmo SLIC convencional mientras que produce un alto rendimiento de segmentación de superpíxeles. En la Figura 3.5 se ejemplifica el proceso de asignación de grupos utilizando la redundancia interpíxel.

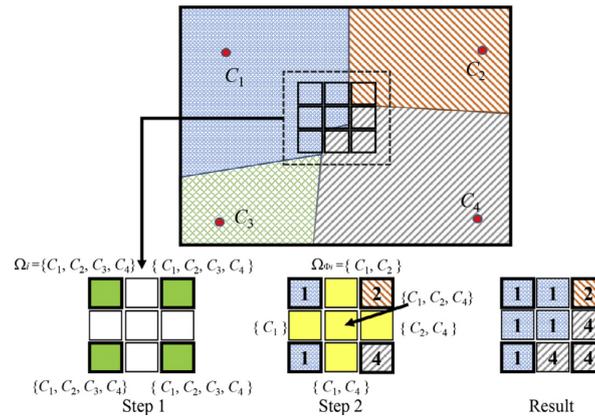


Figura 3.5: Aceleración del algoritmo SLIC a través de submuestreo (Kang-Sun & Oh, 2016).

### Un novedoso método de agrupamiento espectral con superpíxeles para la segmentación de imágenes (Yang et al., 2016)

En este trabajo se propuso un nuevo método de agrupamiento espectral con superpíxeles para la segmentación de imágenes (SCS), también se presentó una nueva medida de similitud difusa del *kernel*, que utiliza la distribución de la pertenencia en la matriz de partición obtenida por el algoritmo de agrupación *kernel fuzzy C-means* (KFCM). La nueva medida de similitud redujo efectivamente la sensibilidad del parámetro de escalamiento de la agrupación espectral, y se aplicó como un preprocesamiento basada en superpíxel para reducir el costo computacional. Al comparar con otros algoritmos clásicos de agrupación en los experimentos realizados, los resultados mostraron que el enfoque propuesto (SCS) logró una alta precisión comparable a los algoritmos clásicos actuales.

### Una aplicación de *Fuzzy C-Means* basada en supervóxel con un soporte de GPU para la segmentación de imágenes volumétricas del cerebro (Fabijańska and Goławski, 2016)

En este trabajo se consideró el problema de la segmentación de imágenes médicas volumétricas. La segmentación rápida y efectiva se obtuvo aplicando el enfoque propuesto que combina la idea de supervóxels y el algoritmo *Fuzzy C-Means* (FCM). El método FCM se adaptó específicamente a la segmentación de las imágenes volumétricas del cerebro (estudios de Tomografía Computarizada TC), que se componen de decenas a cientos de millones de vóxeles.

La reducción propuesta del tamaño de datos de entrada basado en supervóxels varía de 85% a 95% en los ejemplos probados a expensas de varios segundos para crear regiones de supervóxel. La aceleración adicional del método se logró con el soporte del procesador gráfico (GPU). La compresión de datos anterior y la computación paralela de GPU limitan el tiempo de procesamiento a segundos únicos, prácticamente sin ningún costo adicional para los equipos de hardware.

### **Segmentación basada en superpíxel para Imágenes de RM de la próstata en 3D (Tian et al., 2016)**

En este trabajo se propuso un algoritmo de corte de gráficos 3D basado en superpíxeles para obtener la superficie de la imagen de RM de la próstata. Se utilizó un método híbrido que combina cortes de gráficos y un modelo de contorno activo para segmentar las imágenes RM. En lugar de píxeles, los superpíxeles fueron considerados como las unidades de procesamiento básicas para construir un gráfico basado en superpíxel.

Los superpíxeles se etiquetaron como el fondo mediante la minimización de una función de energía usando un corte de gráfico en 3D. El método propuesto se benefició de los superpíxeles en dos aspectos: 1) Hace robustas la forma y las características grises, lo que reduce el riesgo de etiquetar los superpíxeles con etiquetas incorrectas. 2) El método propuesto redujo los costos computacionales y de memoria, lo que hace que la segmentación del volumen 3D sea manejable. Los resultados experimentales muestran que el método propuesto supera a los métodos más modernos en términos de precisión y robustez.

### **maskSLIC: generación de superpíxeles regionales con aplicación a la caracterización de patologías locales en imágenes médicas (Irving, 2016)**

En este trabajo se presentó maskSLIC, una extensión de SLIC para crear superpíxeles dentro de regiones de interés. Las pruebas se realizaron utilizando el conjunto de datos BRATS 2013. El enfoque propuesto fue capaz de representar mejor las subregiones tumorales subyacentes y logra resultados significativamente mejores que SLIC. Las modificaciones al método SLIC estándar mejoró el análisis dentro de regiones definidas, como un tumor u órgano.

### **Agrupación espectral de *kernel* rápido(KSC) (Langone and Suykens, 2017)**

En este trabajo se presentó un algoritmo de agrupamiento espectral rápido capaz de manejar de forma efectiva grandes cantidades de datos resolviendo el problema de escalabilidad en la agrupación espectral, tanto en el uso de la memoria como en el tiempo de cálculo con instancias grandes de datos. El diseño de este método se basó en una configuración del kernel, se utilizó la aproximación de Nyström del mapa de características para resolver el problema de optimización primaria, que caracteriza un modelo de agrupamiento espectral del kernel.

Esto condujo a una reducción de la complejidad del tiempo en comparación con el agrupamiento espectral estándar debido a que se evitan la construcción de la matriz laplaciana y su autodescomposición. Este enfoque ofreció dos ventajas principales; 1) un procedimiento de ajuste riguroso para la selección de un número adecuado de grupos  $y$ ; 2) la predicción de las pertenencias de los grupos para puntos no vistos utilizando la extensión fuera de la muestra.

### **Segmentación de Superpíxel: Un referente (Wang et al., 2017a)**

Este trabajo evaluó 15 algoritmos, de entre los cuales se pueden destacar los algoritmos *Watershed*, *N-cut*, *MeanShift*, *FH*, entre otros. El repositorio utilizado para la evaluación, cuyo código está disponible para el público, proporciona métricas y conjuntos de datos para una comparación cuantitativa de algoritmos para generar superpíxeles. Los resultados de la evaluación mostraron que algunos algoritmos para generar superpíxeles funcionan mejor que otros.

Los algoritmos basados en agrupación fueron más eficiente que los basados en grafos. También se observó que los algoritmos que añaden una restricción compacta a la función de objetivo, producen superpíxeles más compactos, coherentes y regulares. Todos los algoritmos evaluados en este trabajo son difíciles de aplicar a la escena en tiempo real. El tiempo de ejecución de varios algoritmos dependió principalmente del tiempo de iteración durante la optimización, como LRW, SEEDS, *Lattice Cut*. La mayoría de estos algoritmos generaron superpíxeles lentamente cuando el número de superpíxeles es mayor que 1000.

### **Segmentación de imagen sin parámetros con SLIC (Boemer et al., 2018)**

En este trabajo se desarrolló un marco de segmentación de imágenes utilizando SLIC sin parámetros, este enfoque llamado PF-SLIC (*Parameter-free SLIC*) utilizó Máquinas de Aprendizaje Extremo (*Extreme Learning Machines ELM*) para predecir el número de superpíxeles óptimo. ELM son redes neuronales de capa oculta y de propagación hacia delante, con un tiempo de entrenamiento extremadamente rápido y un rendimiento comparable al de otros modelos de aprendizaje automático.

Lo anterior garantizó que la segmentación de superpíxeles producida fuera rápida. El ajuste de parámetros de SLIC se abordó como un problema de predicción multiclase, donde las clases son un conjunto de valores para el parámetro que se desea ajustar. Para generar un conjunto de entrenamiento, se segmentó una colección de imágenes de muestra. El parámetro  $k_{opt}^s$  óptimo produjo una segmentación de superpíxel que mantiene una buena adherencia de límite y detalle, con el menor número de superpíxeles posible. PF-SLIC logró un rendimiento con tiempos de ejecución bajos. Esta implementación es adecuada para procesar imágenes y vídeos de ultra alta definición (4K), cuyo gran tamaño hace que los algoritmos lentos no sean viables.

### **Detección de regiones de lesión de materia blanca en IRM usando SLIC0 y red neuronal convolucional (Diniz et al., 2018)**

Este trabajo presentó una metodología computacional capaz de detectar regiones de lesiones de materia blanca del cerebro de IRM en la modalidad FLAIR. La metodología propuesta utilizó el algoritmo SLIC0 (la versión de SLIC sin parámetros) como parte de un preprocesamiento. Posteriormente, se utilizó una red neuronal convolucional para la clasificación de los superpíxeles obtenidos en el paso anterior.

La metodología se aplicó a 91 imágenes de resonancia magnética proporcionadas por la compañía de diagnóstico médico DASA, y alcanzó una precisión de 98.73%, especificidad de 98.77% y sensibilidad de 78.79% con 0.005 de falsos positivos, sin ninguna técnica de reducción de falsos positivos, en la detección de regiones de lesión de materia blanca. Esta metodología es un referente de las aplicaciones de los algoritmos superpíxel para lograr el éxito en la detección de regiones de interés en imágenes de resonancia magnética.

### 3.3. Discusión

Los trabajos antes mencionados presentan información relevante para esta tesis. En la Tabla 3.1 se presenta un resumen de estos trabajos considerando el objetivo, la metodología utilizada y los resultados. Los algoritmos para generar superpíxeles analizados en este capítulo, parten de una inicialización común, al dividir la imagen en rejillas regulares proporcionales al tamaño del superpíxel. La reducción del espacio de búsqueda, beneficia principalmente en el rendimiento del algoritmo. Otra estrategia recurrente en estos algoritmos, es utilizar diferentes espacios de color, lo cual ayuda a disminuir las variaciones de intensidad en las regiones iniciales, permitiendo mejorar la etapa de agrupamiento y obtener superpíxeles con mayor homogeneidad. Los espacios de color más utilizados son: CIELAB, CIELUV, YCBCR y HSV.

Los algoritmos para generar superpíxeles, generalmente utilizan el conjunto de datos BSDS (*Berkeley Segmentation Dataset*) para su evaluación, debido a que estos algoritmos están diseñados para trabajar con imágenes a color. Este conjunto de datos proporciona puntos de referencia que implementan medidas comunes para evaluar los algoritmos de segmentación y detectores de contornos; además de proporcionar la segmentación *ground truth* por imagen. En el trabajo de (Wang et al., 2017a), se utilizaron 13 métricas para evaluar la eficiencia de los algoritmos para generar superpíxeles, entre las cuales se pueden destacar: Rec (*Boundary Recall*), UE (*Undersegmentation Error*) y EV (*Explained Variation*), los cuales son mencionados frecuentemente en los trabajos presentados en este capítulo. Estas métricas son fundamentales para la evaluación del rendimiento, por lo cual se analizan con mayor detalle en el Capítulo 5.

Entre los trabajos revisados se incluyen algunos enfocados en el procesamiento y análisis de imágenes médicas, en los cuales, se utilizan algoritmos para generar superpíxeles como un paso de preprocesamiento. En (Diniz et al., 2018), se propuso un método para la detección de regiones con lesiones en imágenes RM cerebral, (Tian et al., 2016) propuso un método de segmentación de imágenes volumétricas utilizando superpíxeles, y finalmente otro trabajo importante es (Fabijańska and Goławski, 2016), donde se propuso un método para segmentar imágenes TC (*Computed Tomography*), mediante una adaptación del algoritmo *Fuzzy C-Means* para generar superpíxeles.

Tabla 3.1: Resumen del Estado del Arte.

Artículo	Objetivo	Método/Algoritmo	Resultado	Utilidad
<b>TurboPixels: Superpíxeles rápidos utilizando flujos geométricos (Levinshtein et al., 2009)</b>	Generar superpíxeles utilizando un esquema de crecimiento de región mediante flujos geométricos.	<ul style="list-style-type: none"> <li>■ <i>Gradientes</i>.</li> </ul>	Generó superpíxeles que se adaptan a los bordes de la imagen. El método propuesto se basa en algoritmos de complejidad variable, pero en la práctica, tiene un comportamiento lineal.	Información sobre los métodos de extracción de superpíxeles.
<b>Superpíxeles SLIC (Achanta et al., 2010)</b>	Generar superpíxeles mediante un enfoque basado en <i>k-means</i> .	<ul style="list-style-type: none"> <li>■ <i>k-means</i>.</li> <li>■ <i>Gradientes</i>.</li> </ul>	El algoritmo generó superpíxeles regulares y compactos con alta adherencia a los bordes de la imagen, además, al limitar el espacio de búsqueda, se logró reducir el costo computacional obteniendo una alta velocidad para generar los superpíxeles.	Información sobre los métodos de extracción de superpíxeles.
<b>Superpíxeles con optimización booleana (Zhang et al., 2011)</b>	Generar superpíxeles mediante la optimización de funciones pseudo-booleanas.	<ul style="list-style-type: none"> <li>■ <i>Markov Random Fields (MRF)</i>.</li> </ul>	El algoritmo PB (Pseudo-Boolean) logró generar superpíxeles regulares en tamaño y forma sin la necesidad de un parámetro adicional. El enfoque pseudo-booleano permitió que algoritmo PB obtuviera una alta velocidad para generar los superpíxeles.	Información sobre los métodos de extracción de superpíxeles.

Tabla 3.1: Resumen del Estado del Arte (continuación).

Artículo	Objetivo	Método/Algoritmo	Resultado	Utilidad
Crecimiento de región basado en <i>Eikonal</i> para una agrupación eficiente (Buyssens et al., 2014)	Generar superpíxeles utilizando crecimiento de región basado en la ecuación <i>Eikonal</i> .	<ul style="list-style-type: none"> <li>▪ Ecuación <i>Eikonal</i>.</li> </ul>	El enfoque propuesto generó superpíxeles regulares y compactos. El método puede extenderse para generar supervóxeles.	Información sobre métodos de extracción de superpíxeles.
Píxeles de agua: Superpíxeles basado en la transformación de cuencas (Machairas et al., 2014)	Generar superpíxeles utilizando transformación de cuencas hidrográficas.	<ul style="list-style-type: none"> <li>▪ <i>Watershed transformation</i>.</li> </ul>	Los superpíxeles resultantes mostraron alta adherencia a los bordes de la imagen. La regularidad de los superpíxeles se controlaron a través del parámetro $k$ .	Información sobre los métodos de extracción de superpíxeles.
Cuenca-compacta y SLIC-preferente: Sobre la mejora de las compensaciones de los algoritmos de segmentación de superpíxeles (Neubert and Protzel, 2014)	Mejorar los algoritmos de cuenca hidrográfica y SLIC para extraer superpíxeles proporcionando un equilibrio entre la alta calidad, el bajo tiempo de ejecución y las características de segmentación.	<ul style="list-style-type: none"> <li>▪ <i>Watershed</i>.</li> <li>▪ SLIC.</li> </ul>	Cuenca-compacta: los superpíxeles resultantes presentaron mayor regularidad en comparación con el algoritmo convencional. SLIC-preferente: la optimización del código en combinación con el criterio de terminación local propuesto, permitieron que el algoritmo resultante se ejecute a velocidad de cuadro (30 Hz para el tamaño de imagen $481 \times 321$ ) en una computadora de escritorio estándar.	Información sobre la mejora en la velocidad del algoritmo SLIC.

Tabla 3.1: Resumen del Estado del Arte (continuación).

Artículo	Objetivo	Método/Algoritmo	Resultado	Utilidad
Segmentación de imágenes utilizando agrupación espectral modificado basado en SLIC y Nyström (Bai et al., 2014)	Generar una segmentación de alto nivel utilizando superpíxeles y una modificación del algoritmo de agrupamiento Nyström.	<ul style="list-style-type: none"> <li>▪ Agrupamiento Nyström.</li> <li>▪ SLIC.</li> </ul>	En este trabajo se realizó una modificación del algoritmo SLIC. Los cambios se realizaron en el cálculo de la distancia en color, lo cual permitió generar superpíxeles en imágenes en escala de grises. Las regiones resultantes fueron fusionadas posteriormente mediante el algoritmo de agrupamiento Nyström. El enfoque propuesto logró generar una segmentación de alto nivel y redujo significativamente el requisito de memoria.	Información sobre la adaptación de SLIC para extraer superpíxeles en imágenes en escala de grises.
Segmentación de preservación topológica gruesa a fina en tiempo real (Yao et al., 2015)	Generar superpíxeles utilizando un enfoque de optimización energética.	<ul style="list-style-type: none"> <li>▪ Markov random field.</li> <li>▪ k-means.</li> </ul>	El método propuesto generó superpíxeles poco regulares pero con una alta adherencia a los bordes de la imagen. El algoritmo empleó una única iteración para converger en un mínimo de la energía.	Información sobre los métodos de extracción de superpíxeles.
Convexidad limitada para la extracción eficiente de superpíxeles y supervóxelos (Tasli et al., 2015)	Implementar un método eficiente de generación de superpíxeles utilizando una técnica de adaptación de límites.	<ul style="list-style-type: none"> <li>▪ CCS (Convexity constrained Superpixels).</li> </ul>	El enfoque propuesto mejoró el rendimiento en la extracción de superpíxeles y supervóxelos tanto en la precisión como en la velocidad.	Información sobre el espacio de color LAB y diferentes métricas de distancia para mejorar la extracción de superpíxeles.

Tabla 3.1: Resumen del Estado del Arte (continuación).

Artículo	Objetivo	Método/Algoritmo	Resultado	Utilidad
Segmentación de superpíxeles con agrupamiento espectral lineal (Li and Chen, 2015)	Generar superpíxeles utilizando agrupación espectral.	<ul style="list-style-type: none"> <li>▪ Agrupación espectral.</li> <li>▪ <i>k-means</i>.</li> </ul>	El enfoque propuesto generó superpíxeles regulares con alta eficiencia de memoria y fue capaz de preservar las propiedades globales de las imágenes.	Información sobre métodos de extracción de superpíxeles utilizando algoritmos de agrupamiento.
gSLICr: Superpíxeles SLIC a más de 250Hz (Ren et al., 2015)	Desarrollar una versión de SLIC paralelizada utilizando GPUs, para aplicaciones en tiempo real.	Se implementaron GPUs en el algoritmo SLIC utilizando un marco NVIDIA CUDA.	El enfoque de paralelización, propuesto en este trabajo, logró un rendimiento superior con respecto al enfoque convencional. El algoritmo resultante fue 83x más rápido.	Información sobre la versión paralelizada de SLIC.
Algoritmo de agrupamiento rápido de particionamiento y fusión basado en la extensión (Ghanem et al., 2015)	Agrupar conjuntos de datos numéricos multidimensionales.	Heurístico; los datos de entrada fueron particionados en pequeños bloques, posteriormente se eliminó el ruido, finalmente las particiones densas restantes fueron fusionadas para obtener los grupos finales.	El enfoque propuesto minimizó en gran medida la sobrecarga de las operaciones de agrupamiento y mejoró el rendimiento en las etapas posteriores.	Información sobre métodos de agrupación.

Tabla 3.1: Resumen del Estado del Arte (continuación).

Artículo	Objetivo	Método/Algoritmo	Resultado	Utilidad
SLIC Superpíxel robusto al ruido para imágenes naturales (Dong and Zhou, 2016)	Mejorar SLIC para su aplicación en imágenes con ruido.	<ul style="list-style-type: none"> <li>SLIC.</li> </ul>	La mejora en el algoritmo SLIC se basó en una nueva medida de similitud que utiliza la información estadística de los píxeles. En las pruebas con imágenes con ruido, el enfoque propuesto, conservó la adherencia a los bordes de la imagen, a diferencia del algoritmo convencional.	Información sobre el rendimiento de SLIC con imágenes con ruido.
Aceleración basada en submuestreo de SLIC para la segmentación de superpíxeles (Choi and Oh, 2016)	Mejorar la velocidad y rendimiento del algoritmo SLIC utilizando una técnica de submuestreo.	<ul style="list-style-type: none"> <li>SLIC.</li> <li><i>Subsampling</i>.</li> </ul>	El método propuesto aprovechó la redundancia interpíxel y consiguió mejorar el rendimiento del algoritmo con respecto al enfoque convencional. El algoritmo modificado funcionó casi 5x más rápido, sin afectar las características de los superpíxeles obtenidos.	Información sobre la mejora del rendimiento de SLIC aprovechando la redundancia espacial entre los píxeles.
Un novedoso método de agrupamiento espectral para la segmentación de imágenes con superpíxeles (Yang et al., 2016)	Generar superpíxeles utilizando agrupamiento espectral.	<ul style="list-style-type: none"> <li>Agrupamiento espectral.</li> <li><i>Kernel Fuzzy C-means</i> (KFCM).</li> </ul>	El enfoque propuesto utilizó una medida de similitud difusa que se basó en la distribución de la matriz pertenencia del algoritmo KFCM. El método resultante generó superpíxeles regulares y compactos con alta adherencia a los bordes sin aumentar el requisito de memoria.	Información sobre métodos de extracción de superpíxeles utilizando algoritmos de agrupamiento.

Tabla 3.1: Resumen del Estado del Arte (continuación).

Artículo	Objetivo	Método/Algoritmo	Resultado	Utilidad
Una aplicación de <i>Fuzzy C-Means</i> basados en supervóxeles con un soporte de GPU para la segmentación de imágenes volumétricas del cerebro (Fabijańska and Gołowski, 2016)	Segmentar imágenes volumétricas del cerebro utilizando un enfoque basado supervóxeles.	<ul style="list-style-type: none"> <li>▪ <i>Fuzzy C-Means</i>.</li> <li>▪ Unidad de procesamiento gráfico (GPU).</li> </ul>	El enfoque propuesto generó supervóxeles mediante la adaptación del algoritmo <i>Fuzzy C-Means</i> . La segmentación de la imagen volumétrica (estudio de Tomografía Computarizada) se obtuvo agrupando supervóxeles con características similares. La aceleración adicional del método se logró con el soporte del procesador gráfico (GPU).	Información sobre la adaptación del algoritmo <i>Fuzzy C-Means</i> para sobresegmentar una imagen volumétrica.
Segmentación basada en supervóxeles para Imágenes de RM de la próstata en 3D (Tian et al., 2016)	Segmentar imágenes de resonancia magnética de la próstata en 3D utilizando supervóxeles.	<ul style="list-style-type: none"> <li>▪ <i>Graph Cut</i>.</li> </ul>	El enfoque propuesto parte de una sobresegmentación del estudio completo, utilizando el algoritmo <i>Graph Cut</i> . Posteriormente, los supervóxeles se agrupan en base a una función de energía, obteniendo una segmentación. El método propuesto redujo el costo computacional y de memoria.	Información sobre extracción de supervóxeles en imágenes médicas.
<i>maskSLIC</i> : generación de supervóxeles regionales con aplicación a la caracterización de patologías locales en imágenes médicas (Irving, 2016)	Adaptar el algoritmo SLIC para generar supervóxeles dentro de regiones de interés.	<ul style="list-style-type: none"> <li>▪ <i>maskSLIC</i>.</li> </ul>	El enfoque propuesto fue capaz de generar supervóxeles en regiones de interés y logró representar mejor las subregiones tumorales subyacentes, obteniendo resultados significativamente mejores que SLIC.	Información sobre extracción de supervóxeles en imágenes médicas.

Tabla 3.1: Resumen del Estado del Arte (continuación).

Artículo	Objetivo	Método/Algoritmo	Resultado	Utilidad
Agrupación espectral rápida de kernel (KSC) (Langone and Suykens, 2017)	Resolver el problema de escalabilidad en la agrupación espectral.	<ul style="list-style-type: none"> <li>Agrupamiento Nyström.</li> </ul>	El enfoque propuesto redujo la complejidad del tiempo en comparación con el agrupamiento espectral estándar.	Información sobre métodos de agrupación.
Segmentación de Superpíxel: Un referente (Wang et al., 2017a)	Proporcionar un esquema de evaluación estandarizado de algoritmos para generar superpíxeles, incluyendo conjuntos de datos e implementación de código abierto en el lenguaje Matlab.	15 algoritmos para generar superpíxeles.	Se construyó un marco de referencia para la evaluación de algoritmos para generar superpíxeles. El punto de referencia evaluó 15 métodos de segmentación de superpíxeles con 13 métricas, que posteriormente se integró a una biblioteca.	Información de bases de datos y métricas para la evaluación de algoritmos que generan superpíxeles.
Segmentación de imagen sin parámetros con SLIC (Boemer et al., 2018)	Crear una versión del algoritmo SLIC libre de parámetros.	<ul style="list-style-type: none"> <li>PF-SLIC.</li> <li><i>Extreme Learning Machines</i> (ELM).</li> </ul>	El enfoque propuesto produjo una segmentación de superpíxel que mantiene una buena adherencia a los bordes de la imagen, con un número de superpíxeles relativamente bajo. PF-SLIC logró un rendimiento con tiempos de ejecución mínimos.	Información sobre los métodos de extracción de superpíxeles.
Detección de regiones de lesión de materia blanca en IRM usando SLIC0 y red neuronal convolucional (Diniz et al., 2018)	Crear una metodología computacional para la detección de regiones lesionadas en imágenes de resonancia magnética del cerebro.	<ul style="list-style-type: none"> <li>SLIC0.</li> </ul>	La metodología se aplicó a 91 imágenes de resonancia magnética proporcionadas por la compañía de diagnóstico médico DASA, y alcanzó una precisión de 98.73%, especificidad de 98.77% y sensibilidad de 78.79% con 0.005 de falsos positivos.	Información sobre extracción de superpíxeles en imágenes médicas.

# Capítulo 4

## Metodología

En este capítulo se aborda el tema principal de esta investigación y se presenta el enfoque propuesto para la extracción de superpíxeles utilizando el algoritmo de Agrupamiento Difuso Intuitivo. El objetivo principal es aprovechar las características de la teoría difusa intuitiva para lograr la extracción eficiente de superpíxeles, tomando en cuenta las propiedades básicas de los algoritmos superpíxel. El método propuesto se plantea para la sobresegmentación de imágenes de resonancia magnética cerebral, debido a su importancia actual en el campo de investigación médica. Adicionalmente, se hace una extensión del método propuesto para extraer superpíxeles en imágenes a color.

### 4.1. IFCM Superpíxeles

Esta implementación es una adaptación del algoritmo de Agrupamiento Difuso Intuitivo para generar superpíxeles, considerando las propiedades de los superpíxeles, tales como: la homogeneidad de los segmentos, así como la adaptación a los límites naturales de la imagen. Los algoritmos tradicionales para generar superpíxeles, estudiados en la Sección 2.2, utilizan diferentes enfoques como agrupamiento, grafos, optimización energética, etc. En estos métodos, la asignación de los píxeles a un determinado superpíxel se convierte en una asignación totalmente binaria.

Para mejorar este proceso, el enfoque propuesto considera los píxeles como conjuntos difusos intuitivos. Con esto, cada píxel pertenece a todos los superpíxeles iniciales con diferentes grados de pertenencia.

La homogeneidad de los superpíxeles se va refinando en la etapa de agrupamiento, donde la pertenencia de cada píxel a un determinado superpíxel se calcula iterativamente. La etapa de etiquetado se hace en base a la matriz de pertenencia  $U$ , que se obtiene al finalizar la etapa de agrupamiento. En esta implementación también se considera un paso posprocesamiento para reforzar la conectividad entre los superpíxeles.

## 4.2. Descripción del método

El algoritmo propuesto agrupa los píxeles representados como Conjuntos Difusos Intuitivos (IFS). Esta representación se hace en base a (Dubey et al., 2017), y se denota de la siguiente manera. Para una imagen en escala de grises, con ancho  $W$ , altura  $H$  y  $N = W \times H$  píxeles, con niveles de intensidad entre 0 y  $L - 1$ , la representación de la imagen con conjuntos difusos intuitivos se da como:

$$I = \{x_i, \mu(x_i), \nu(x_i), \pi(x_i)\}, 1 < i < N \quad (4.1)$$

donde  $\mu(x_i)$ ,  $\nu(x_i)$  y  $\pi(x_i)$  son los grados de pertenencia, no pertenencia e indecisión, del píxel  $i$  en la imagen respectivamente, y  $N$  es el número total de píxeles en la imagen. La transformación al dominio intuitivo se realiza con los métodos utilizados en la literatura. Con estos métodos, es posible construir IFS, conservando las propiedades definidas por (Atanassov, 1987). Para la imagen  $I$ , representado como IFS (como se muestra en expresión (4.1)), el grado de pertenencia  $\mu_I : X \rightarrow [0, 1]$  de cada píxel  $x_i$ , se obtiene normalizando los niveles de intensidad utilizando la siguiente expresión:

$$\mu(x_i) = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \quad i = 1, \dots, N; \quad x_i \in 0, \dots, L - 1 \quad (4.2)$$

donde  $\max(x_i)$  y  $\min(x_i)$  denotan el máximo y el mínimo valor de los niveles de gris de la imagen respectivamente. El grado de no pertenencia se obtiene considerando las complementaciones de (Sugeno, 1993) o (Yager, 1979). En la literatura se utiliza convencionalmente la función de negación de Sugeno para su generación  $\nu_I : X \rightarrow [0, 1]$ , que se define como:

$$\nu(x_i) = \frac{1 - \mu(x_i)}{1 + \lambda\mu(x_i)} \quad \lambda > 0 \quad (4.3)$$

donde  $\lambda$  es un variable que toma valores  $0 < \lambda < \infty$ , y se deriva de la solución original de la función de negación de Sugeno.

El grado de indecisión  $\pi(x_i)$  (también llamado índice intuicionista), se obtiene al restar el grado de pertenencia y no pertenencia de uno.

$$\pi(x_i) = 1 - \mu(x_i) - \nu(x_i) \quad (4.4)$$

con las condiciones  $\pi(x_i) \leq 1$  y  $\mu(x_i) + \nu(x_i) + \pi(x_i) = 1$ . Con el proceso descrito anteriormente, la imagen de entrada  $I$ , se transforma al dominio difuso intuitivo, al representar la intensidad de los píxeles como IFS. En el siguiente paso, los IFS obtenidos se agrupan considerando sus propiedades (el grado de pertenencia, no pertenencia e indecisión). Esta implementación se basa en el algoritmo de Agrupamiento Difuso Intuitivo (*Intuitionistic Fuzzy C-Means IFCM*) propuesto por (Xu&Wu, 2010), analizado en la Sección 2.3. El algoritmo IFCM, particiona los datos utilizando una función objetivo, en la implementación de (Xu&Wu, 2010), se denota como :

$$\min J_m(U, Z, V) = \sum_{j=1}^p \sum_{i=1}^c u_{ij}^m d^2(Z_j, V_i) \quad (4.5)$$

donde  $Z = \{Z_1, Z_2, \dots, Z_p\}$  son  $p$  IFS,  $c$  es el número de grupos, y  $V = \{V_1, V_2, \dots, V_c\}$  son los centroides iniciales. El parámetro  $m$  es el coeficiente difuso ( $m > 1$ ), también llamado *fuzzifier*,  $u_{ij}$  es el grado de pertenencia del  $j$ -ésimo dato al  $i$ -ésimo grupo,  $U = (u_{ij})_{c \times p}$  es la matriz de pertenencia y  $d^2(Z_j^{IFS}, V_i^{IFS})$  es la distancia euclidiana difusa intuitiva al cuadrado que se calcula de siguiente forma:

$$d^2(Z_j^{IFS}, V_i^{IFS}) = ((\mu(Z_j) - \mu(V_i))^2 + (\nu(Z_j) - \nu(V_i))^2 + (\pi(Z_j) - \pi(V_i))^2) \quad (4.6)$$

El algoritmo IFCM, es esencialmente un método de optimización, que minimiza la función objetivo. Para generar superpíxeles utilizando este enfoque, se plantea una adaptación del algoritmo IFCM, tomando en cuenta las propiedades básicas de los superpíxeles, sin afectar el rendimiento del algoritmo de agrupamiento. Considerando que cada superpíxel inicial tiene un dato representativo, la función objetivo se denota de la siguiente manera:

$$\min J_m(U, V^{IFS} : X^{IFS}) = \sum_{j=1}^N \sum_{i=1}^K u_{ij}^m d^2(x_j, v_i) \quad (4.7)$$

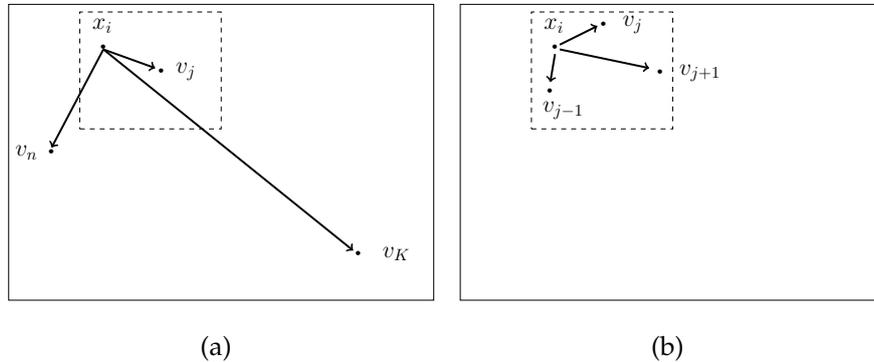
donde  $X^{IFS}$  son  $N$  datos,  $V^{IFS}$  son los centroides de los  $K$  superpíxeles iniciales, ambos representados como conjuntos difusos intuitivos.

El vector de centroides  $V^{FIS} = \{V_1^{IFS}, V_2^{IFS}, \dots, V_K^{IFS}\}$  se compone inicialmente de los valores de intensidad de los píxeles en la imagen. Estos valores se obtienen al dividir la imagen en rejillas regulares de tamaño  $R = \sqrt{N/K}$ . Para obtener una representación más precisa de los centroides  $V^{IFS}$ , la inicialización se refuerza utilizando un método basado en la distribución normal. Este método considera la información local de los superpíxeles iniciales, obteniendo una representación más confiable. Esta inicialización, permite que el método propuesto aumente su rendimiento y converja en pocas iteraciones.

En esta implementación el espacio de búsqueda se limita en una región de tamaño  $2R \times 2R$ , esto con el fin de aumentar la velocidad del algoritmo y conservar la homogeneidad de los segmentos. El algoritmo convencional, calcula la pertenencia de un determinado dato  $x_j$ , considerando todos los centroides  $V^{IFS} = \{V_1^{IFS}, V_2^{IFS}, \dots, V_K^{IFS}\}$ . Esto puede causar que los segmentos no sean homogéneos, debido a que se consideran píxeles que no están dentro del espacio de búsqueda. Para abordar este inconveniente se consideran los tres centroides más próximos, al rededor del espacio de búsqueda. Con estos ajustes, la expresión para calcular se obtiene de la siguiente forma:

$$u_{ij} = \frac{1}{\sum_{r=i}^{i \in K_p} \left( \frac{d^2(x_j^{IFS}, v_i^{IFS})}{d^2(x_j^{IFS}, v_r^{IFS})} \right)^{\frac{1}{m-1}}} \quad (4.8)$$

donde  $u_{ij}$  es la pertenencia,  $K_p$  es un vector que contiene los 3 centroides más próximos ( $K_p \subseteq K$ ),  $d^2(x_j^{IFS}, v_i^{IFS})$  es la distancia euclidiana, descrita anteriormente. En la Figura 4.1 se muestra una representación de estas modificaciones.



**Figura 4.1:** Reducción de espacio de búsqueda; (a) proceso de búsqueda del algoritmo IFCM, (b) búsqueda del algoritmo modificado, en una región limitada.

La actualización del vector de centroides  $V^{IFS} = \{v_1^{IFS}, \dots, v_K^{IFS}\}$  se realiza de forma convencional, utilizando las siguientes expresiones:

$$\begin{aligned}\mu(v_i) &= \frac{\sum_{j=1}^N u_{ij}^m \mu(x_j)}{\sum_{j=1}^N u_{ij}^m} \\ \nu(v_i) &= \frac{\sum_{j=1}^N u_{ij}^m \nu(x_j)}{\sum_{j=1}^N u_{ij}^m} \\ \pi(v_i) &= \frac{\sum_{j=1}^N u_{ij}^m \pi(x_j)}{\sum_{j=1}^N u_{ij}^m}\end{aligned}\tag{4.9}$$

donde  $v_i^{IFS} = \{\mu(v_i), \nu(v_i), \pi(v_i)\}$  es el centroide,  $x_j$  es dato,  $u_{ij}$  es el grado de pertenencia del  $j$ -ésimo dato al  $i$ -ésimo grupo y  $m$  es el coeficiente difuso. Este proceso se repite hasta que el error sea menor al establecido o se llegue al número máximo de iteraciones. La precisión del algoritmo puede controlarse con la variable  $\varepsilon$ . El número máximo de iteraciones se establece en 10, aunque en las pruebas preliminares del algoritmo se observa que converge entre 2 y 5 iteraciones.

El resultado de los algoritmos para extraer superpíxeles usualmente se guardan en un vector con la misma dimensión que la imagen de entrada, donde cada elemento del vector corresponde al grado de pertenencia del píxel en la misma posición  $i$  al superpíxel  $S_n$ .

Desde la perspectiva del agrupamiento difuso, cada píxel pertenece a todos los superpíxeles con diferentes grados de pertenencia. Partiendo de este punto, la asignación de los píxeles a un determinado superpíxel con el enfoque propuesto, se realiza a través de un mapeo a la matriz de pertenencia  $U = (u_{ij})_{N \times K}$ . En este proceso se obtiene el mayor grado de pertenencia  $u_{ij}$  de cada píxel  $j$  y se asigna al vector de etiquetas  $L^N$ , utilizando la expresión (18). Esto considerando que la matriz de pertenencia  $U$  cumple con la condición  $\sum_{j=1}^K u_{ij} = 1$ .

$$\begin{cases} L_j = \arg \max P(u_{ij}) \\ \{u_{ij} | u_{ij} \in U \wedge 1 \leq i \leq K, 1 \leq j \leq N, \forall ij\} \end{cases}\tag{4.10}$$

donde  $L_j$  corresponde a la posición del grado de pertenencia del píxel  $j$  en la imagen original,  $N$  es el número de datos,  $K$  es el número de superpíxeles y  $u_{ij}$  es el grado de pertenencia.

Posteriormente la sobsegmentación se refuerza utilizando un paso posprocesamiento que une superpíxeles pequeños con sus vecinos. La unión de los superpíxeles puede controlarse a través de un parámetro  $\alpha$ , que se compara con una medida de disimilaridad entre los superpíxeles. Para unir los segmentos se obtiene la intensidad media del superpíxel y posteriormente se calcula la distancia euclidiana  $d(s_k, s_{k+1})$ . Los segmentos se unen si  $d(s_k, s_{k+1}) < \alpha$ . En las Figuras 4.2a y 4.2b, se muestra el flujo del método propuesto para extraer superpíxeles, contemplado sus diferentes etapas. En el Algoritmo 4.1 se resumen los pasos para su implementación.

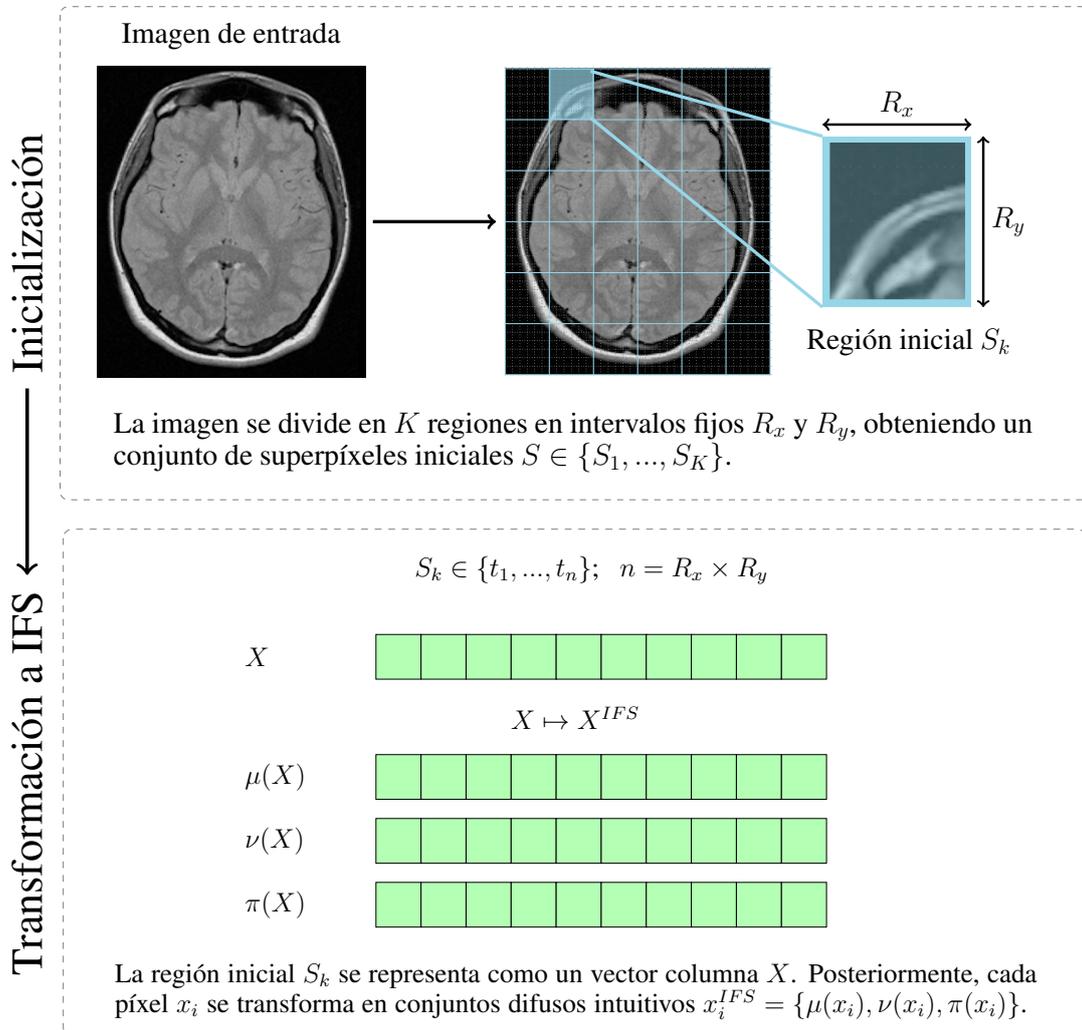
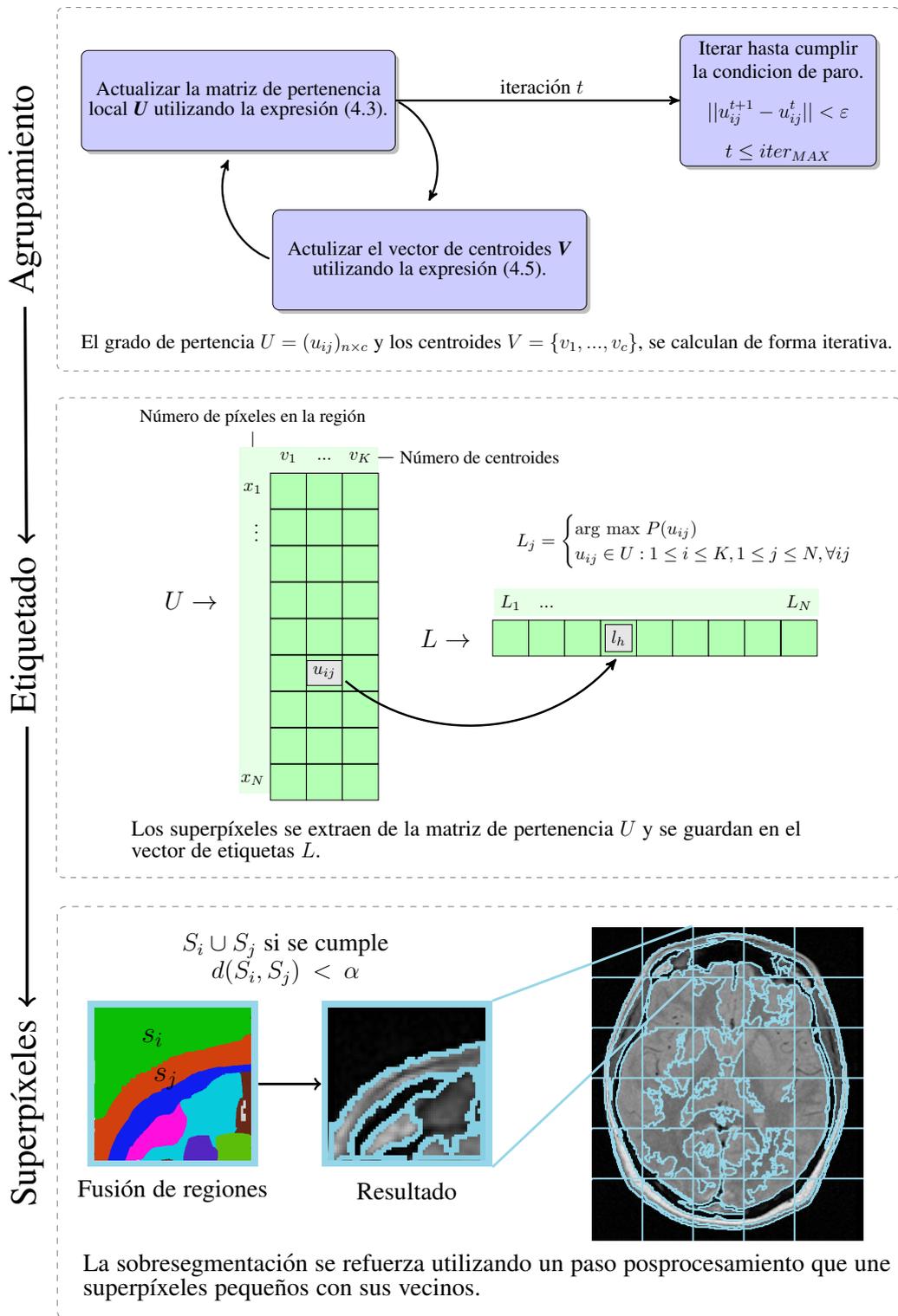


Figura 4.2a: Flujo de procesos para la extracción de superpíxeles con el algoritmo IFCM, en las etapas de inicialización, transformación de los datos y agrupamiento.



**Figura 4.2b:** Flujo de procesos para la extracción de superpíxeles con el algoritmo IFCM, en las etapas de etiquetado y extracción de superpíxeles.

**Algoritmo 4.1:** Superpíxeles basados en IFCM.

---

**input:** Imagen es escala de grises  $X$ ,  $K$ .  
**output:** Imagen sobresegmentada  $Y$ .

- 1  $X \mapsto \tilde{X}$ ;
- 2  $\tilde{X} \mapsto X^{IFS}$ ;
- 3 inicializar  $V_i^{IFS} = [\mu(v_i), \nu(v_i), \pi(v_i)]$ ;  $i = 1, \dots, K$ ; muestreando píxeles en la rejilla regular;
- 4 calcular  $u_{ij}^{(0)} \leftarrow \frac{1}{\sum_{r=1}^K \left( \frac{d^2(x_j^{IFS}, v_r^{IFS})}{d^2(x_j^{IFS}, v_i^{IFS})} \right)^{\frac{1}{m-1}}}$ ;
- 5  $l \leftarrow 0$ ;
- 6 **while**  $\max \|U^{(l+1)} - U^l\| < \varepsilon$  **do**
  - 7  $v_i^{IFS} \leftarrow \left\{ \frac{\sum_{j=1}^N u_{ij}^m \mu(x_j)}{\sum_{j=1}^N u_{ij}^m}, \frac{\sum_{j=1}^N u_{ij}^m \nu(x_j)}{\sum_{j=1}^N u_{ij}^m}, \frac{\sum_{j=1}^N u_{ij}^m \pi(x_j)}{\sum_{j=1}^N u_{ij}^m} \right\}$
  - 8 **for**  $i \leftarrow 1$  **to**  $K$  **do**
    - 9 **for each**  $x_j^{IFS} \in 2R \times 2R$  **do**
    - 10 **if**  $\forall i, r, d^2(x_j^{IFS} - v_r^{IFS}) > 0$  **then**
    - 11  $u_{ij}^{(l)} \leftarrow \frac{1}{\sum_{r=i}^{i \in K_p} \left( \frac{d^2(x_j^{IFS}, v_r^{IFS})}{d^2(x_j^{IFS}, v_i^{IFS})} \right)^{\frac{1}{m-1}}}$ ;
    - 12 **end**
    - 13 **end**
    - 14 **end**
    - 15  $l \leftarrow l + 1$ ;
    - 16 **end**
    - 17 **for**  $i \leftarrow 1$  **to**  $N$  **do**
      - 18 **for**  $j \leftarrow 1$  **to**  $K$  **do**
      - 19  $Y_i \leftarrow \operatorname{argmax}_j p(u_{ij} \in U : u_{ij} > Y_i)$ ;
      - 20 **end**
      - 21 **end**
      - 22 **return**  $Y$

---

El algoritmo 4.1 resume los pasos del método propuesto. La entrada es una imagen en escala de grises representada como un vector columna de tamaño  $W \times H$ ,

donde  $W$  y  $H$  representan las dimensiones de la imagen en lo ancho y alto respectivamente. En los pasos 1 y 2, se realiza la conversión al espacio difuso intuitivo. Para esto la imagen de entrada  $X$  se convierte al espacio difuso, a través de una normalización de los niveles de intensidad de los píxeles, utilizando la expresión (4.2). Con esto, los niveles de intensidad toman valores en el rango  $[0,1]$ . Este paso es importante ya que a partir de la pertenencia  $\mu(x_j)$ , se generan la no pertenencia  $\nu(x_j)$  y el índice intuitivo  $\pi(x_j)$ .

Posteriormente se inicializa el vector de centroides  $V^{IFS}$  con el muestreo de píxeles al dividir la imagen en rejillas regulares de tamaño  $R = \sqrt{N/K}$ . Para inicializar  $V^{IFS}$  se utiliza un método que se basa en la distribución normal tomando en cuenta la media de la región. Esta inicialización permite representar mejor la región y debido a que se considera la información local, ayuda a que el algoritmo converja en pocas iteraciones.

Antes de entrar al proceso iterativo, el cálculo de la pertenencia  $u_{ij}^{(0)}$  se realiza con las expresiones convencionales, considerando todas las regiones. En el paso 6 se establece la condición de paro. En el paso 7 se actualiza los centroides utilizando las expresiones (4.9). En los pasos 8,14 se calcula la pertenencia a través del recorrido de las regiones. En los pasos 17-19 se extraen los superpíxeles de la matriz de pertenencia  $U$ . El resultado se denota como un vector de etiquetas  $Y^N$ , que representa la imagen sobresegmentada.

### Inicialización

El algoritmo IFCM puede tener mejor rendimiento con una buena inicialización de los centroides. El algoritmo convencional utiliza una inicialización aleatoria. En esta implementación se utiliza una inicialización con base en una distribución normal, considerando la media aritmética ( $\bar{x}_\mu$ ,  $\bar{x}_\nu$  y  $\bar{x}_\pi$ ) de los IFS en cada región inicial, representados como conjuntos difusos intuitivos.

**Algoritmo 4.2:** Inicialización.

---

```

input:  $X^{IFS}$ ,  $K$ .
output:  $V^{IFS}$ .
1 Fijar  $\phi$ ;
2 for  $i \leftarrow 1$  to  $K$  do
3   for  $x_j^{IFS} \in 2R \times 2R$  do
4      $\bar{x}_\mu \leftarrow \frac{1}{N_R} \sum_{j=1}^{N_R} \mu(x_j)$ ;
5      $\bar{x}_\nu \leftarrow \frac{1}{N_R} \sum_{j=1}^{N_R} \nu(x_j)$ ;
6      $\bar{x}_\pi \leftarrow \frac{1}{N_R} \sum_{j=1}^{N_R} \pi(x_j)$ ;
7   end
8   for  $x_j^{IFS} \in 2R \times 2R$  do
9      $a_\mu \leftarrow \max\{a_\mu, \mu(x_j) \in X^{FIS}\}$ ;
10     $a_\nu \leftarrow \max\{a_\nu, \nu(x_j) \in X^{FIS}\}$ ;
11     $a_\pi \leftarrow \max\{a_\pi, \pi(x_j) \in X^{FIS}\}$ ;
12  end
13   $\mu(v_i) \leftarrow [(2a_\mu)\sigma - \phi] + \bar{x}_\mu$ ;
14   $\nu(v_i) \leftarrow [(2a_\nu)\sigma - \phi] + \bar{x}_\nu$ ;
15   $\pi(v_i) \leftarrow [(2a_\pi)\sigma - \phi] + \bar{x}_\pi$ ;
16   $v_i^{IFS} \leftarrow \{\mu(v_i), \nu(v_i), \pi(v_i)\}$ ;
17 end
18 return  $V^{FIS}$ 

```

---

El algoritmo toma como entrada la imagen representada como conjunto difuso intuitivo  $X^{IFS}$ , el número de superpíxeles  $K$ . El algoritmo obtiene la media aritmética de cada índice difuso ( $\bar{x}_\mu$ ,  $\bar{x}_\nu$  y  $\bar{x}_\pi$ ) de los píxeles en la región delimitada  $2R \times 2R$ .  $\phi$  es una variable que toma valores  $0 < \phi < 1$ ,  $\sigma$  es un número aleatorio generado con base a una distribución gaussiana.

### Extensión a imágenes a color

El método propuesto también puede extenderse para manejar imágenes a color, con pocas modificaciones al algoritmo, que incluye los tres canales, representada como  $I: W \times H \rightarrow \mathbb{R}^3$ . Cada canal se puede tratar como una característica independiente, por lo cual es necesario realizar la conversión al dominio difuso intuitivo separando los canales. En esta implementación se utiliza el espacio de color IJK, debido a que es de fácil conversión, se calcula aplicando una transformación simple al espacio de color RGB.

En comparación a otros espacios de color como: CIE L\*a\*b\* o YCbCr, el espacio de color IJK separa la luminosidad en el canal  $I$  y la información de textura en los canales  $J$  y  $K$ .

### Sistema de color ortonormal IJK

El espacio de color IJK se obtiene al aplicar una rotación simple al espacio de color RGB. El eje  $I$  se extiende a lo largo del vector  $y$ , por lo tanto, representaba la luminancia del espacio de color RGB. El eje  $J$  se encuentra en el plano ortogonal al eje  $I$ , y está en la dirección de rojo en el espacio de color RGB. El eje  $K$  es ortogonal a los ejes  $I$  y  $J$ , y por lo tanto se encuentra en el plano ortogonal al eje  $I$  en la dirección de azul-verde (Bromiley et al., 2001). La conversión del espacio de color RGB a IJK se realiza con las siguientes ecuaciones particulares (Pătrașcu, 2007).

$$I = \frac{R + G + B}{3} \quad (4.11)$$

$$J = \frac{2R - G - B}{2\sqrt{3}} \quad (4.12)$$

$$K = \frac{G - B}{\sqrt{6}} \quad (4.13)$$

Algunos trabajos en la literatura, como (Cyganek, 2009) y (Mújica-Vargas et al., 2013), utilizan este espacio de color debido a que la información de luminancia se separa en un solo componente ( $I$ ), además, los tres ejes son ortogonales y la conversión es particularmente simple. El método propuesto para imágenes de color se resumen en el Algoritmo 4.3.

El algoritmo no tiene un cambio significativo con respecto a la versión para imágenes en escala de grises, ya que para procesar los tres canales, se pueden interpretar como vectores independientes. En el paso 4, la expresión se modifica, añadiendo  $x_j^{IFS}$ , que se utiliza para denotar los canales, correspondientes al espacio de color IJK, como conjuntos difusos intuitivos. Estas modificaciones también se pueden observar en la actualización del vector de centroides (paso 7) y en la actualización de la matriz de pertenencia (paso 11).

**Algoritmo 4.3:** Superpíxeles basados en IFCM para imágenes a color.

---

**input:** Imagen es escala de grises  $X$ ,  $K$ .  
**output:** Imagen sobresegmentada  $Y$ .

- 1  $X \mapsto \tilde{X}$ ;
- 2  $\tilde{X} \mapsto X^{IFS}$ ;
- 3 inicializar  $V_i^{IFS} = [\mu(v_i), \nu(v_i), \pi(v_i)]$ ;  $i = 1, \dots, K$ ; muestreando píxeles en la rejilla regular;
- 4 calcular  $u_{ij}^{(0)} \leftarrow \frac{1}{\sum_{r=1}^K \left( \frac{d^2(x_j^{IFS \times 3}, v_r^{IFS})}{d^2(x_j^{IFS \times 3}, v_r^{IFS})} \right)^{\frac{1}{m-1}}}$ ;
- 5  $l \leftarrow 0$ ;
- 6 **while**  $\max ||U^{(l+1)} - U^l|| < \varepsilon$  **do**
  - 7  $v_i^{IFS} \leftarrow \left\{ \begin{array}{l} \frac{\sum_{j=1}^N u_{ij}^m \mu(x_j^3)}{\sum_{j=1}^N u_{ij}^m}, \frac{\sum_{j=1}^N u_{ij}^m \nu(x_j^3)}{\sum_{j=1}^N u_{ij}^m}, \frac{\sum_{j=1}^N u_{ij}^m \pi(x_j^3)}{\sum_{j=1}^N u_{ij}^m} \end{array} \right\}$
  - 8 **for**  $i \leftarrow 1$  **to**  $K$  **do**
    - 9 **for each**  $x_j^{IFS \times 3} \in 2R \times 2R$  **do**
      - 10 **if**  $\forall i, r, d^2(x_j^{IFS \times 3} - v_r^{IFS}) > 0$  **then**
        - 11  $u_{ij}^{(l)} \leftarrow \frac{1}{\sum_{r=i}^{i \in K_p} \left( \frac{d^2(x_j^{IFS \times 3}, v_r^{IFS})}{d^2(x_j^{IFS \times 3}, v_r^{IFS})} \right)^{\frac{1}{m-1}}}$
        - 12 **end**
      - 13 **end**
    - 14 **end**
    - 15  $l \leftarrow l + 1$ ;
  - 16 **end**
  - 17 **for**  $i \leftarrow 1$  **to**  $N$  **do**
    - 18 **for**  $j \leftarrow 1$  **to**  $K$  **do**
      - 19  $Y_i \leftarrow \operatorname{argmax}_j p(u_{ij} \in U : u_{ij} > Y_i)$ ;
      - 20 **end**
    - 21 **end**
    - 22 **return**  $Y$

---

### 4.3. Discusión

El método propuesto adapta al algoritmo IFCM para extraer superpíxeles. Esta modificación se realiza principalmente para realizar la agrupación es regiones limitadas. Esta estrategia es adoptada por la mayoría de los algoritmos para generación de superpíxeles, teniendo un gran beneficio principalmente en la reducción del tiempo de ejecución.

La inicialización del vector de centroides  $V$ , se forma inicialmente, mediante el muestreo de píxeles en una region  $2R \times 2R$ . Para reforzar la inicialización, se utilizó un método con base a una distribución gaussiana, el cual es capaz de representar los datos de manera eficiente al utilizar información de las regiones.

El método converge en pocas iteraciones, adicionalmente se establece la variable  $iter_{max}$  que controla el número máximo de iteración. El algoritmo se extendió para imágenes a color con pocas modificaciones a la versión para imágenes en escala de grises. En esta implementación se optó por utilizar el espacio de color IJK, debido a que es de fácil conversión, y es capaz de representar eficientemente la información de luminosidad y textura en imágenes a color.

# Capítulo 5

## Experimentación y resultados

En este capítulo se describe las diferentes pruebas que se realizaron para evaluar la eficiencia del algoritmo propuesto. Se experimentó con imágenes en escala de grises y a color. La experimentación se dividió en dos partes:

1) Se realizó una serie de pruebas con imágenes en escala de grises y a color; el propósito de esta prueba es el rendimiento del algoritmo con diferentes imágenes, así como la optimización de los parámetros.

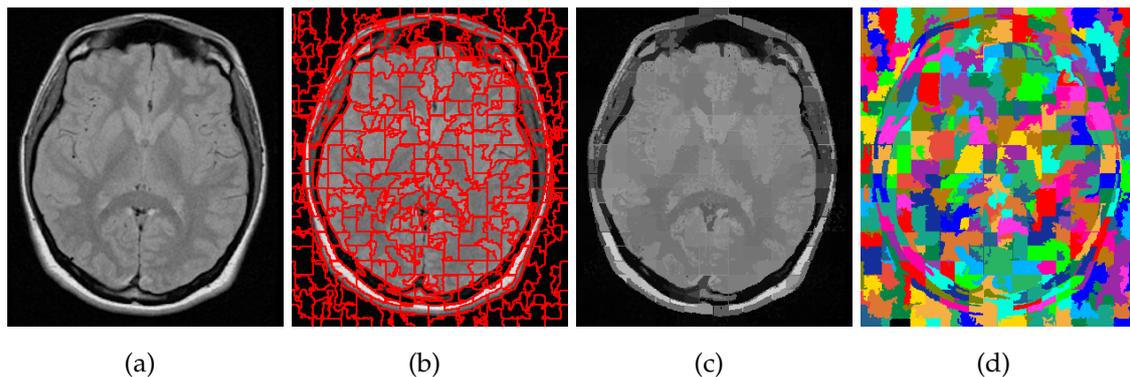
2) La segunda prueba es una comparación con técnicas para extracción de superpíxeles existente en la literatura. Para esta prueba se contemplaron diferentes conjuntos de datos, así como métricas para evaluar su eficiencia cuantitativamente.

### 5.1. Implementaciones

La implementación del algoritmo **IFCM** para extraer superpíxeles, descrito en la Sección 4.2, se desarrolló en el Lenguaje C siguiendo el paradigma MVC (*Model View Controller*) con el fin de tener mayor estabilidad y mantenibilidad durante su codificación. Siguiendo este paradigma, los módulos implementados tiene una mejor fluidez y tiene la ventaja de poder reutilizar el código como un conjunto de funciones externas.

Esta implementación cuenta con diferentes utilidades para procesar imágenes en formato PPM (*Portable Pixmap Format*) y PGM (*Portable Graymap Format*) desde la entrada estándar, leer y escribir archivos CSV (*comma-separated values*) y generar las

diferentes representaciones de los superpíxeles (contornos, media de las regiones y colores aleatorios) para visualizar los resultados como se muestra en la Figura 5.1. También se escribieron scripts en bash y en Matlab para facilitar el procesamiento de la base de datos. Estas implementaciones se hicieron con el fin de tener mayor compatibilidad con el repositorio utilizado.



**Figura 5.1:** Imagen de resonancia magnética cerebral con las diferentes representaciones de superpíxeles; (a) original, (b) contornos, (c) intensidad media de los segmentos y (d) colores aleatorios.

La representación con contornos permite dar una estimización cualitativa de la adaptación a los bordes de la imagen. Por otra parte la representación con la intensidad media de los superpíxeles, permite visualizar que tan homogéneos son los segmentos obtenidos. Con esta representación, la imagen se aprecia con menos detalles aunque se conservan los más esenciales. La representación con los colores aleatorios usualmente se utilizan para dar una estimación visual de la compacidad y de la interconectividad entre los superpíxeles resultantes.

### 5.1.1. Repositorio

Desde que el término superpíxel fue propuesto por (Ren and Malik,2010), se han desarrollado varios algoritmos para segmentación de superpíxeles. En la mayoría de los trabajos existentes como: (Moore et al., 2008), (Achanta et al., 2012), (Fu et al., 2014) y (Wang et al., 2017a), se han propuesto diferentes métricas en forma independiente, debido a esto, se genera cierta incertidumbre en la forma de como las implementaron y de como manejan el *ground truth*.

## 5.1 Implementaciones

Derivado del trabajo de (Stluz,2017), se ofrece un repositorio<sup>1</sup> como punto de referencia para la comparación cualitativa y cuantitativa de los algoritmos para generación de superpíxeles. Contiene implementaciones de 13 métricas para trazar resultados y ofrece compatibilidad con bases de datos como BSDS500, NYUV2, SBD, SUNRGBD y Fash e incluye diferentes herramientas de preprocesamiento, formato y conversión. En la Tabla 5.1 se muestra un resumen de las bases de datos compatibles con este repositorio.

**Tabla 5.1:** Resumen de los conjuntos de datos compatibles con en el repositorio.

	BSDS500	SBD	NYUV2	SUNRGBD	Fash
<i>Train</i>	200	238	199	200	222
<i>Test</i>	200	477	399	400	463
<i>Train size</i>	481 × 321	316 × 240	608 × 448	658 × 486	400 × 600
<i>Test size</i>	481 × 321	314 × 242	608 × 448	660 × 488	400 × 600

En la Tabla 5.1 se especifica la cantidad de imágenes por subconjunto en cada repositorio. BSDS500, Fash y SBD se componen de imágenes en formato `.jpg`, por otra parte, NYU y SUNRGBD se componen de imágenes RGBD obtenida a través de secuencias de vídeo en formato `.mat`.

Este repositorio requiere de un formato específico para procesar los segmentos *ground truth*. Por lo cual se le aplicó un preprocesamiento para convertir estos segmentos a los formatos requeridos en los dos conjuntos de datos considerados.

---

<sup>1</sup> Este repositorio es una extensión de una investigación que presenta una evaluación completa de 28 algoritmos de última generación en 5 conjuntos de datos diferentes. Esta página proporciona un nuevo punto de referencia que permite centrarse en una evaluación justa, proporcionando información novedosa. El punto de referencia que incluye todos los algoritmos evaluados está disponible: <https://davidstutz.de/projects/superpixel-benchmark/>

### 5.1.2. Algoritmos

Los cinco algoritmos considerados para la evaluación comparativa se describieron en el Capítulo 3, los cuales son **W** (*Watershed*), **SLIC** (*Simple Linear Iterative Clustering*), **LSC** (*Linear Spectral Clustering*), **TP** (*TurboPixel*) y **PB** (*Pseudo Boolean Superpixel*). Los algoritmos considerados pueden trabajar con imágenes a color y en escala de grises. En la Tabla 5.2 se presenta un resumen de estos algoritmos.

**Tabla 5.2:** Resumen de los algoritmos considerados para la evaluación.

Nombre	Enfoque	Año	Ref.	Parámetros	Control sobre		
					Superpíxeles	Compacidad	Iteraciones
<b>W</b>	Cuenca	2015	(Hu et al., 2015)	1	✓	-	-
<b>LSC</b>	Agrupación	2015	(Li and Chen, 2015)	3	✓	✓	✓
<b>SLIC</b>	Agrupación	2010	(Achanta et al., 2010)	2	✓	✓	✓
<b>TP</b>	Evolución del contorno	2009	(Levinshtein et al; 2009)	3	✓	✓	-
<b>PB</b>	Grafos	2011	(Zhang et al., 2011)	2	✓	-	-

### 5.1.3. Conjunto de datos

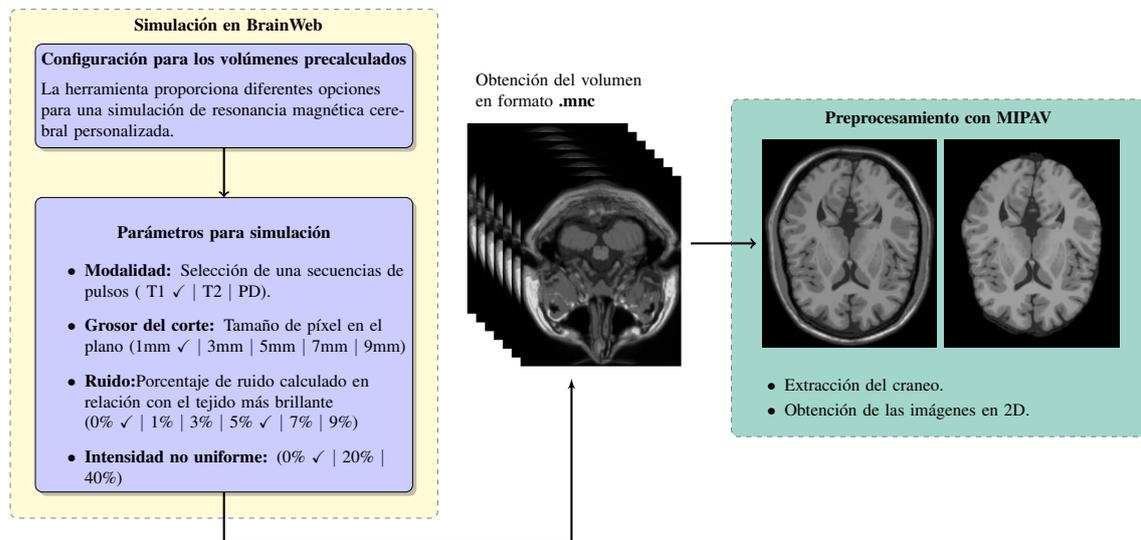
Para una evaluación completa, se contemplaron los conjuntos de datos BrainWeb (Cocosco et al., 1997) y BSDS500 (Martin et al., 2001). Las pruebas con el conjunto de datos de BrainWeb, se consideran con el objetivo de evaluar el rendimiento del algoritmo **IFCM** para extraer superpíxeles en imágenes complejas como lo es la resonancia magnética cerebral. Adicionalmente, se contempló BSDS500 para la evaluación con imágenes a color. La adquisición y las características de los conjunto de datos se describen a continuación.

#### BrainWeb

Para la evaluación con imágenes cerebrales, se utilizó un conjunto de datos que corresponde a un estudio de RM del cerebro sin anomalías proporcionado por simulador BrainWeb. El conjunto de datos de Brainweb, aunque consiste en un único espectro digital, viene con diferentes opciones de simulación relacionadas con el grosor del corte, la cantidad de ruido y la intensidad no uniforme en la imagen simulada.

Para esta evaluación se simularon 2 estudios. El primer estudio se generó en modalidad T1, libre de ruido, con una resolución de  $181 \times 217 \times 181$ , con un tamaño de píxel en el plano de  $1mm$  y un grosor del corte de  $1mm$ .

El segundo estudio presenta las mismas características que el anterior, agregando únicamente 5% de ruido gaussiano. Utilizando el visualizador de imágenes médicas MIPAV, se aplicó un preprocesamiento para extraer el cráneo del estudio como se observa en la Figura 5.2.



**Figura 5.2:** Adquisición de las imágenes de RM cerebral utilizando el simulador BrainWeb.

Este preprocesamiento se realiza debido a que los tejidos no cerebrovasculares (cráneo y grasa) presentan intensidades que se superponen con las intensidades de los tejidos cerebrales de interés como la materia blanca, la materia gris y el líquido cefalorraquídeo (Despotović et al., 2015), obstaculizando los métodos de segmentación. MIPAV proporciona varios métodos para extraer superficies de los volúmenes como el algoritmo BET. Este algoritmo extrae la superficie del cerebro de una MRI, eliminando las áreas fuera del cerebro.

Finalmente, se obtuvieron 181 imágenes por estudio en formato PGM, con un rango dinámico de 8 bits y una resolución de  $181 \times 217$  píxeles. La ventaja de utilizar el conjunto de datos de Brainweb en la evaluación es la disponibilidad del *ground truth* que se obtienen a través de volúmenes que conforman el modelo anatómico del cerebro.

## BSDS500

El conjunto de datos de segmentación de Berkeley es uno de los más utilizados para la evaluación de los algoritmos para generación de superpíxeles. Consta de 500 imágenes, cada una con 5 segmentaciones *ground truth* diferentes de alta calidad como se aprecia en la Figura 5.3. BSDS500 se divide en un subconjunto de entrenamiento de 200 imágenes, un subconjunto de validación de 100 imágenes y un subconjunto de prueba de 200 imágenes.

Las imágenes representan simples escenas al aire libre, que muestran paisajes, edificios, animales y humanos, donde el primer plano y el fondo suelen identificarse fácilmente. Sin embargo, las escenas naturales donde los límites del segmento no son claramente identificables, contribuyen a la dificultad del conjunto de datos.



**Figura 5.3:** Imagen tomada del conjunto de datos BSDS500-*test*-2018, la imagen original y sus 5 segmentaciones *ground truth*.

### 5.1.4. Métricas

Para evaluar la calidad de la sobresegmentación, se utilizan tres métricas principales: Recuerdo de límites (*Rec*), el error de subsegmentación (*UE*) y la variación explicada (*EV*), que cuantifican el nivel de adherencia a los bordes de la imagen.

#### Recuerdo de límites (*Rec*)

Esta métrica es la más utilizada para evaluar la adhesión a los límites de la imagen dado el *ground truth*. Sea  $FN(G, S)$  y  $TP(G, S)$  el número de píxeles falsos negativos y de verdaderos positivos en una segmentación de superpíxeles  $S$  con respecto a el *ground truth*  $G$ . Entonces, la métrica *Rec* se define como

$$Rec(G, S) = \frac{TP(G, S)}{TP(G, S) + FN(G, S)} \quad (5.1)$$

En general, *Rec* aproximado a uno representa una mejor adherencia de límites con respecto a los límites del *ground truth*. En la práctica, un píxel límite en *S* se corresponde con un píxel límite arbitrario en *G* dentro de un vecindario local. La recuperación de límites mide qué fracción de los bordes del *ground truth* se encuentran dentro de al menos dos píxeles de un límite de superpíxeles (Martin et al., 2004). El resultado de esta métrica son valores en un rango de [0,1], donde un valor aproximado a la unidad, indica un rendimiento alto.

### El Error de Subsegmentación (*UE*)

Esta métrica mide la desviación de los superpíxeles con respecto a un segmento *ground truth* específico. Dada una región de  $g_i$  del *ground truth* y el conjunto de superpíxeles necesarios para cubrirla,  $s_j | s_j \cap g_i$  mide cuántos píxeles de  $s_j$  se desvían a través del límite de  $g_i$ ,  $M$  representa el número de segmentos de *ground truth*,  $N$  es el número total de píxeles en la imagen y  $B$  es un número mínimo de píxeles en  $s_i$  que se superponen  $g_i$ . El error de subsegmentación se expresa como:

$$UE = \frac{1}{N} \left[ \sum_{I=1}^M \left( \sum_{s_j | s_j \cap g_i > B} |s_j| \right) - N \right] \quad (5.2)$$

El error de subsegmentación es una medida implícita de la adherencia a los bordes de la imagen (Achanta et al., 2012). El resultado de esta métrica son valores en un rango de [0,1]. A diferencia de la métrica *Rec*, un valor aproximado a cero, indica un rendimiento alto.

### Variación Explicada (*EV*)

Esta métrica mide qué tan bien, los datos en los píxeles originales están representados por los superpíxeles. La variación explicada se de nota como

$$EV = \frac{\sum_i (\mu_i - \mu)^2}{\sum_i (x_i - \mu)^2} \quad (5.3)$$

donde  $i$  es el píxel,  $x_i$  es el valor real de píxeles,  $\mu$  es la media global de píxeles y  $\mu_i$  es el valor medio de los píxeles asignados al superpíxel que contiene  $x_i$ . La variación explicada *EV* toma el máximo valor posible de 1 a medida que aumenta el número de superpíxeles. Por el contrario, toma el valor mínimo posible de 0 cuando solo hay un superpíxel (el valor de la imagen).

Al igual que la métrica  $Rec$ , un valor aproximado a la unidad, indica un rendimiento alto. Esta métrica proporciona una medida del rendimiento de los algoritmos superpíxel sin utilizar el *ground truth* (Moore et al., 2008).

## 5.2. Pruebas con el algoritmo

Antes de pasar a la comparación con los algoritmos considerados, se realiza una experimentación con el algoritmo propuesto, donde se evalúa su rendimiento con diferentes valores de  $\alpha$ . Este parámetro se utiliza en el posprocesamiento y funciona como un umbral para fusionar superpíxeles pequeños. El nivel de adherencia se cuantifica con las métricas descritas en la Sección 5.1.4.

### 5.2.1. Pruebas con el parámetro $\alpha$

La evaluación del rendimiento del algoritmo propuesto se realizó con diferentes valores de  $\alpha \in \{10, 15, 20\}$ , con el fin de obtener el valor con mejor adaptación a los bordes de la imagen. Para esta prueba se consideraron imágenes en escala de grises con las bases de datos BrainWeb. En general, la tendencia en la adherencia a los bordes de la imagen con respecto a las métricas  $Rec$ ,  $UE$  y  $EV$ , no cambia significativamente, con excepción en el tiempo de ejecución. En la Figura 5.4, se observa los resultados obtenidos. La adherencia a los bordes con respecto a la métrica  $Rec$  (ver Figura 5.4–(a)), se observa que para los valores  $\alpha = 10$  y  $\alpha = 20$  siguen aproximadamente la misma tendencia, teniendo un valor de adherencia de 0.778 para  $\alpha = 10$  y 0.776 para  $\alpha = 15$  con  $k = 200$ . En la Tabla 5.3 puede observar con más detalle estos resultados. Los valores en negritas en las Tablas 5.3, 5.4 y 5.5 indican los mejores resultados.

**Tabla 5.3:** Resultados de la evaluación con la métrica  $Rec$ .

$K$	$Rec$											
	200	300	400	500	550	700	800	850	900	1000	1200	1450
$\alpha = 10$	0.951	0.962	0.967	0.972	0.972	0.977	0.977	0.980	0.980	0.982	0.982	<b>0.987</b>
$\alpha = 15$	0.949	0.961	0.965	0.971	0.971	0.975	0.975	0.977	0.977	0.979	0.979	0.983
$\alpha = 20$	0.976	0.978	0.980	0.979	0.979	0.979	0.979	0.981	0.981	0.980	0.980	0.982

La adherencia a los bordes mejora conforme el número de superpíxeles se incrementa, obteniendo un valor de adherencia de 0.998 para  $\alpha = 10$  y 0.997 para  $\alpha = 15$  con  $k = 1450$ .

## 5.2 Pruebas con el algoritmo

Por otra parte los resultados con  $\alpha = 20$  muestran un buen rendimiento con números de superpíxeles pequeños, teniendo un nivel de adherencia de 0.90 para  $k = 200$ . Para  $k = 1450$  se observa rendimientos similares para  $\alpha = 10$  y  $\alpha = 20$  alcanzando un nivel de adherencia de 0.998.

Con respecto a la métrica *UE* (Figura 5.4–(b)), se observa que el error de subsegmentación sigue la misma tendencia para  $\alpha = 10$  y  $\alpha = 15$  para valores pequeños de superpíxeles, teniendo ambos un error de 0.099  $k = 200$ . Para  $\alpha = 20$  el error es de 0.068 considerando el mismo valor de  $K$ . Conforme se aumenta el número de superpíxeles el error disminuye, proporcionando una mejor adaptación a los bordes de la imagen. El error más pequeño se consigue con  $\alpha = 10$  con 0.046 seguidos de  $\alpha = 15$  y  $\alpha = 20$  con un error de 0.050 y 0.052 respectivamente. En la Tabla 5.4 se muestran a detalle los resultados.

**Tabla 5.4:** Resultados de la evaluación con la métrica *UE*.

K	UE											
	200	300	400	500	550	700	800	850	900	1000	1200	1450
$\alpha = 10$	0.099	0.087	0.077	0.068	0.068	0.062	0.062	0.058	0.058	0.054	0.054	<b>0.046</b>
$\alpha = 15$	0.099	0.085	0.077	0.068	0.068	0.063	0.063	0.060	0.060	0.057	0.057	0.050
$\alpha = 20$	0.068	0.062	0.056	0.056	0.056	0.055	0.055	0.053	0.053	0.055	0.055	0.052

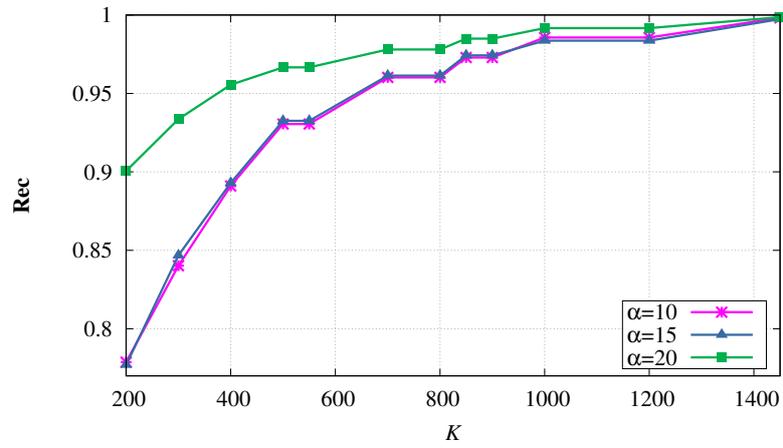
Los resultados obtenidos con la métrica *EV* se muestran en la Tabla 5.5 y en la Figura 5.4–(c). Con respecto a esta métrica, se observa la misma tendencia que en la métrica *Rec*, al mejorar la adherencia a los bordes con forme se incrementa el valor de los superpíxeles. Para  $k = 200$  se alcanza un nivel de adherencia de 0.951 0.949 y 0.976 para  $\alpha = 10$ ,  $\alpha = 15$  y  $\alpha = 20$ . El mayor nivel de adherencia se obtiene con  $\alpha = 10$  con 0.987, seguido de  $\alpha = 15$  con 0.983 y  $\alpha = 20$  0.982.

**Tabla 5.5:** Resultados de la evaluación con la métrica *EV*.

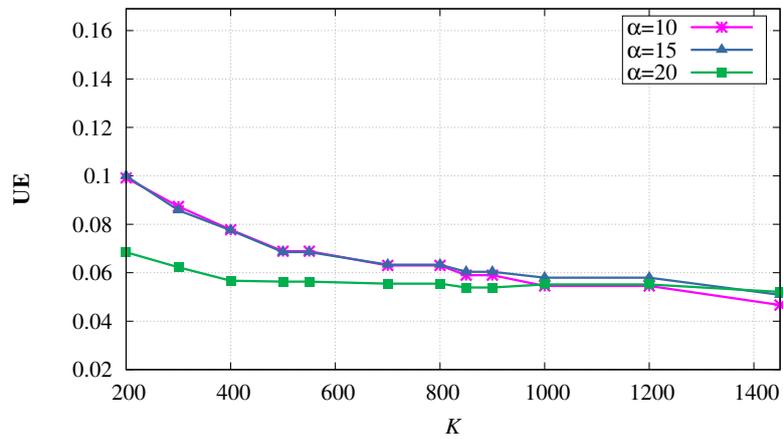
K	EV											
	200	300	400	500	550	700	800	850	900	1000	1200	1450
$\alpha = 10$	0.951	0.962	0.967	0.972	0.972	0.977	0.977	0.98	0.98	0.982	0.982	<b>0.987</b>
$\alpha = 15$	0.949	0.961	0.965	0.971	0.971	0.975	0.975	0.977	0.977	0.979	0.979	0.983
$\alpha = 20$	0.976	0.978	0.98	0.979	0.979	0.979	0.979	0.981	0.981	0.98	0.98	0.982

Los resultados anteriores muestran que  $\alpha = 10$ , tiene los mejores valores en 2 de las métricas (*UE* y *EV*), y aunque se tiene un rendimiento medio con números de superpíxeles pequeños en la métrica *Rec*, logra tener el mismo nivel de adherencia que  $\alpha = 20$  conforme se incrementar el número de superpíxeles.

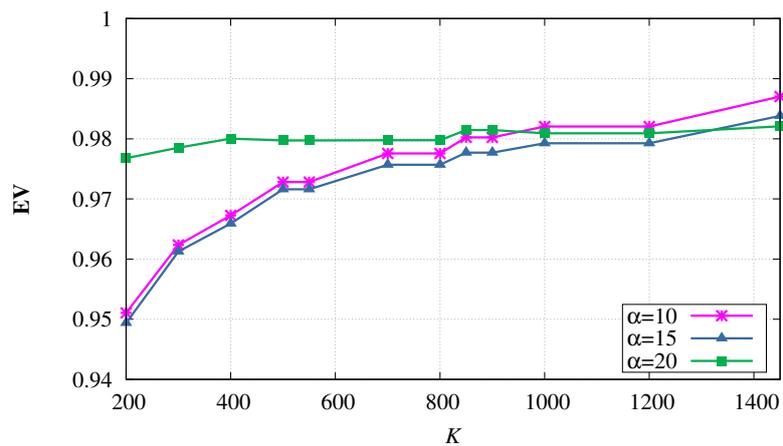
## 5.2 Pruebas con el algoritmo



(a)



(b)



(c)

**Figura 5.4:** Resultados de la experimentación con el parámetro  $\alpha$ ; (a) Recuerdo de límites (*Rec*), (b) Error de subsegmentación (*UE*) y (c) Variación Explicada (*EV*).

### 5.2.2. Optimización de parámetros

Los algoritmos para extraer superpíxeles considerados, tienen la opción de controlar el número de superpíxeles generados, pero en algunos algoritmos esta característica depende en gran medida de otros parámetros y procesos, por ejemplo, los algoritmos **SLIC**, **IFCM** y **LSC**, cuentan con un paso posprocesamiento para fusionar superpíxeles pequeños. Aunado a esto, algunos superpíxeles iniciales suelen desaparecer en la etapa de agrupamiento, lo cual hace que el número resultante no sea exacto, sino una aproximación del número  $K$  requerido.

Los algoritmos como **TP**, **PB**, **W** y **IFCM**, requieren de otro ajuste de parámetros adicionales que son específicos del enfoque que utilizan para agrupar píxeles. Para resolver este inconveniente y tener una evaluación más precisa se evaluaron los algoritmos utilizando diferentes combinaciones de parámetros y se eligieron los que dieron como resultado el número aproximado de superpíxeles deseados. Para el algoritmo **IFCM** se experimentó con diferentes valores de  $\lambda$ , factor para la transformación al dominio difuso intuitivo, que tiene una influencia significativa en la etapa de inicialización de los datos.

Se experimentó con  $0,1 > \lambda < 2,8$ , siendo  $\lambda = 0,89$  el valor que muestra mejores resultados. En el caso del parámetro  $m$  (*fuzzifier*), donde  $1 < m < \infty$ , este valor determina cuánto pueden superponerse los grupos entre si. Si  $m = 1$ , el algoritmo se reduce al algoritmo clásico de *k-means*. Cuanto mayor sea  $m$ , mayor será la superposición entre los grupos, en el estado del arte se utiliza convencionalmente  $m = 2$  para **FCM** y **IFCM**, por lo cual se conserva este valor para las pruebas posteriores. Para  $\varepsilon$ , que se utiliza para medir la precisión del algoritmo, se observa que  $\varepsilon = 1e - 6$  tiene buenos resultados con una rápida convergencia de entre 2 a 5 iteraciones.

### 5.3. Evaluación y Comparación

El algoritmo propuesto, basado en IFCM, se comparó con 5 algoritmos existentes en el estado del arte: **W**, **LSC**, **SLIC**, **TP** y **PB** ( ver Tabla 5.2). Se consideraron algoritmos con diferentes enfoques para generar superpíxeles los cuales se describieron en la Sección 3.2. Se utilizó la base de datos BrainWeb: *Simulated Data Base*. Esta base de datos permite simular estudios de resonancia magnética cerebral tanto sanas como con lesiones de esclerosis múltiple en las modalidades T1 y T2.

Adicionalmente se consideró la base de datos BSDS500, para la evaluación con imágenes a color. Las métricas que se consideraron, permiten cuantificar el nivel de adherencia a los bordes de la imagen. Los algoritmos y las métricas antes mencionados, están disponibles en el repositorio utilizado para la evaluación. En la Figura 5.5 se muestra un esquema general para la evaluación comparativa.

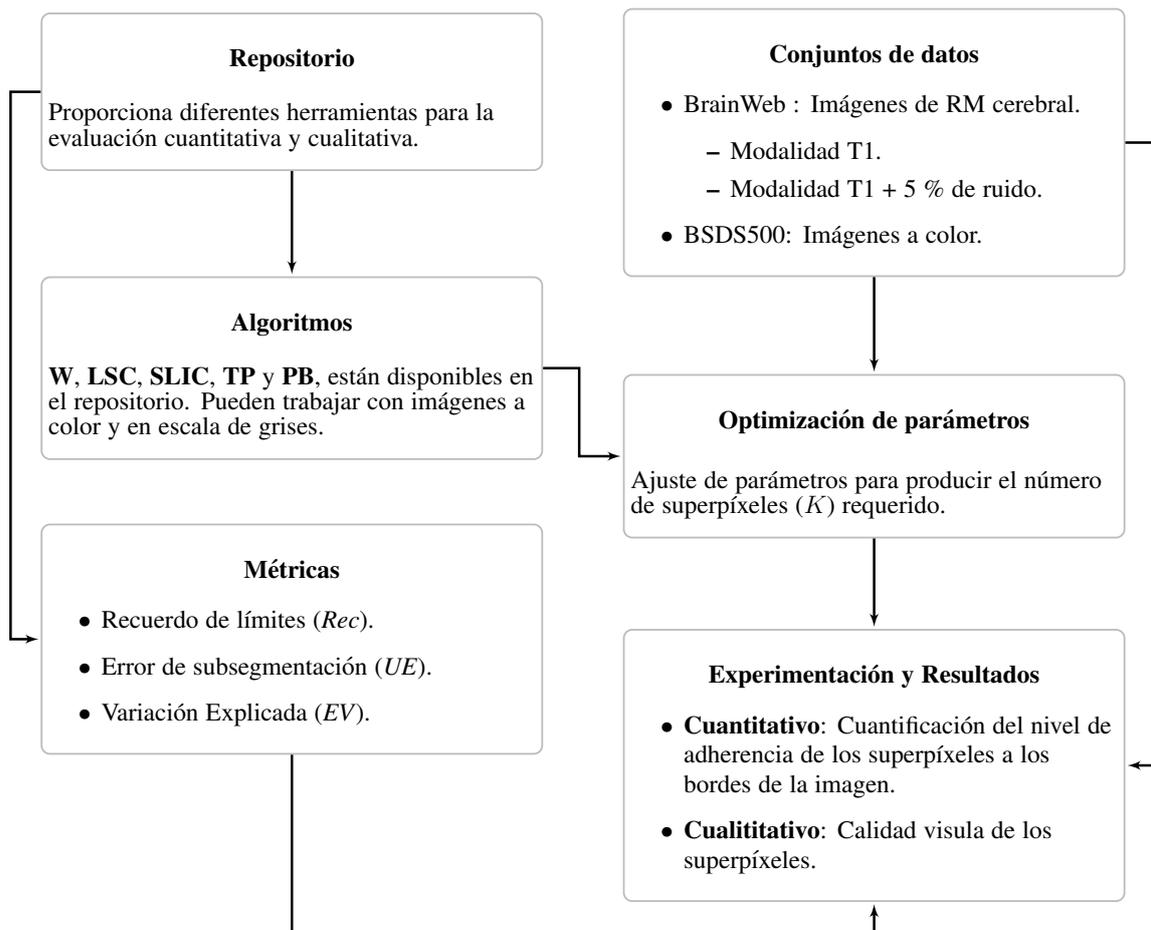


Figura 5.5: Esquema de experimentación para la evaluación comparativa.

### 5.4. Resultados con el conjuntos de datos Brain Web

En esta sección se describen los resultados obtenidos con los dos conjuntos de datos simulados con Brain Web. Como se describió en la Sección 5.1.3, el primer estudio de RM cerebral se compone de 182 imágenes, libres de ruido, con una resolución de  $181 \times 217$  píxeles. El segundo estudio se simuló agregando 5% de ruido.

Los resultados se dividen en dos partes: La parte cuantitativa analiza el rendimiento de los algoritmos con respecto a las métricas *Rec*, *UE*, *EV*, así como el tiempo de ejecución. La parte cualitativa examina la calidad visual de los superpíxeles generados considerando sus características, tales como la compacidad y la regularidad.

#### 5.4.1. RM cerebral en modo T1

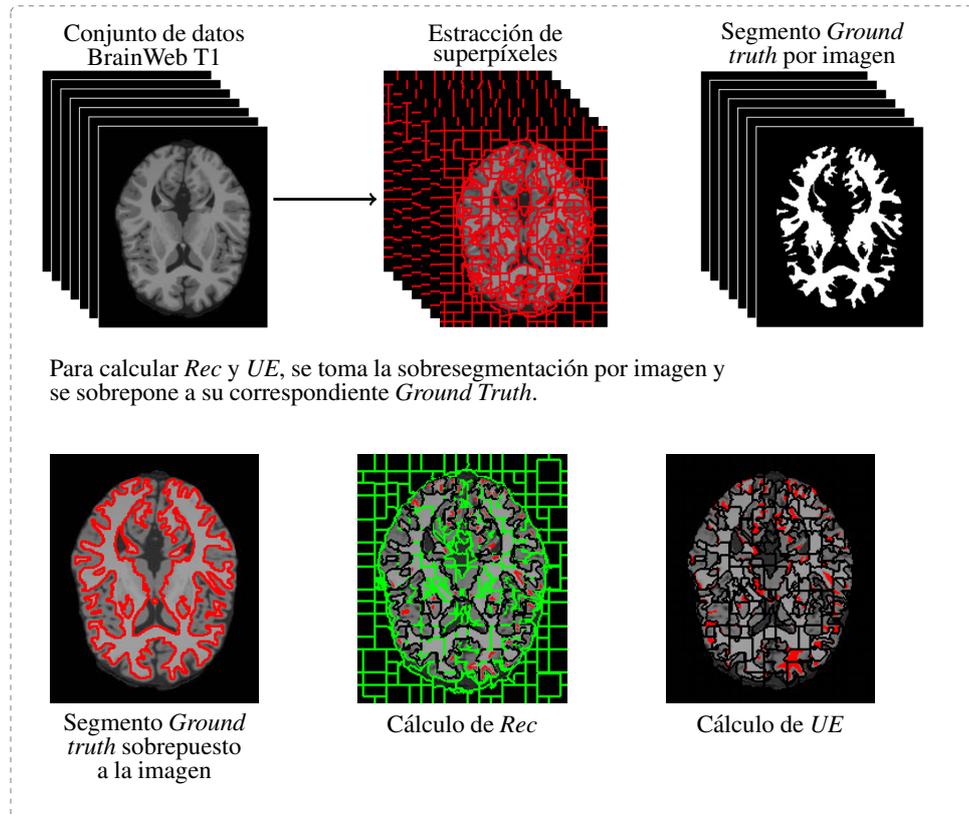
##### Cuantitativo

La cuantificación del rendimiento de los algoritmos para extracción de superpíxeles, se determina con las métricas: *Rec* (Recuerdo de límites), *UE* (Error de subsegmentación) y *EV* (Variación Explicada). Estas cuantifican el nivel de adherencia de los superpíxeles a los límites de la imagen, característica deseable en los algoritmos para extracción de superpíxeles, ya que, una mejor adaptación de los límites, proporciona una representación más precisa de la imagen original. Estas métricas están implementadas en el repositorio utilizado.

*UE* y *Rec* dependen de los segmentos *ground truth* para cuantificarse, a diferencia de *EV*, que ofrece una medida independiente al utilizar la información de los superpíxeles y de la imagen para calcularse. *UE*, *Rec* y *EV* tienen diferentes perspectivas para medir la precisión de la sobresegmentación por lo cual se obtiene una evaluación detallada. En la Figura 5.6 se muestra un ejemplo de como se cuantifica la adherencia a los bordes de la imagen, a través de las métricas *Rec* y *UE*. Ambas métricas, calculan el número de superpíxeles desviados de superpíxel  $S_j$ , con respecto al segmento *ground truth*.

Para este conjunto de datos fueron considerados diferentes valores de superpíxeles  $K \in \{200, 300, 400, 500, 550, 700, 800, 850, 900, 1000, 1200, 1450\}$ .

## 5.4 Resultados con el conjuntos de datos Brain Web



**Figura 5.6:** Cálculo de las métricas *Rec* y *UE*; la sobresegmentación se sobrepone al segmento *ground truth*, esto permite calcular los píxeles que no se ajustan a los bordes de la imagen.

El tamaño de rejilla  $R$  depende del número total de píxeles en la imagen y el número de superpíxeles  $K$  requerido. Para evitar que el tamaño del intervalo de rejillas, sea demasiado pequeño, el valor máximo de superpíxel se limita a  $K = 1450$ . Por lo cual el tamaño de intervalo mínimo que se alcanza es:  $R = \sqrt{\frac{181 \cdot 217}{1450}} = 5 \times 5$  píxeles. Con esto se asegura que la granularidad del superpíxel sea suficiente y que contenga el número de píxeles mínimos necesarios para realizar los cálculos en la etapa de agrupamiento.

Los resultados obtenidos en la evaluación, con respecto a la métrica *Rec* (Reuerdo de límites), se muestran en la Tabla 5.6. El valor en negrita representa el mejor resultado. En esta Tabla se observa que el algoritmo **LSC**, tiene el mejor rendimiento con valores de superpíxeles pequeños, logrando un nivel de adherencia de 0.946 con  $K \approx 200$ . Sin embargo, **LSC**, no logra mejorar a partir de cierto punto ( $K \approx 1000$ ).

## 5.4 Resultados con el conjuntos de datos Brain Web

Por el contrario, los algoritmos **W**, **SLIC**, **IFCM**, **TP** y **PB**, logran mejorar su rendimiento significativamente conforme se incrementa el número de superpíxeles. El algoritmo **W**, logra un buen rendimiento a partir de  $K \approx 400$  y alcanza un nivel de adherencia máximo de 0.996 con  $K \approx 1450$ .

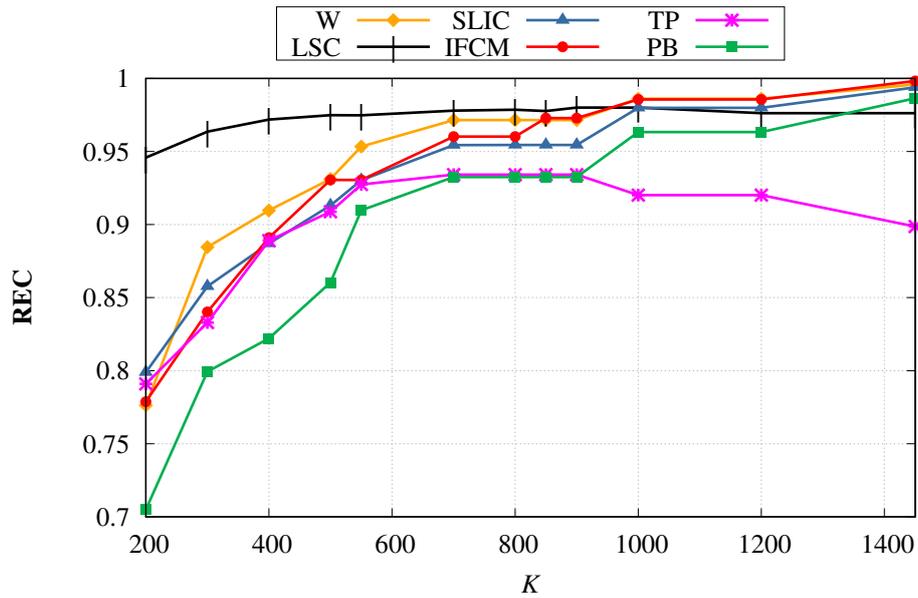
**Tabla 5.6:** *Rec*, en el conjunto de datos BrainWeb T1; el nivel de adherencia de los algoritmos se muestra para cada valor de  $K$  considerado.

$K$	<i>Rec</i>					
	<b>W</b>	<b>LSC</b>	<b>SLIC</b>	<b>IFCM</b>	<b>TP</b>	<b>PB</b>
200	0.776	0.946	0.799	0.779	0.791	0.705
300	0.885	0.964	0.858	0.840	0.833	0.799
400	0.910	0.972	0.887	0.891	0.889	0.822
500	0.931	0.975	0.913	0.930	0.909	0.860
550	0.953	0.975	0.930	0.930	0.927	0.910
700	0.972	0.978	0.954	0.960	0.934	0.933
800	0.972	0.979	0.954	0.960	0.934	0.933
850	0.972	0.978	0.954	0.973	0.934	0.933
900	0.972	0.980	0.954	0.973	0.934	0.933
1000	0.986	0.980	0.980	0.986	0.920	0.963
1200	0.986	0.976	0.980	0.987	0.920	0.963
1450	0.996	0.976	0.994	<b>0.998</b>	0.899	0.986

Los algoritmos **IFCM**, **SLIC** y **TP**, tienen un buen rendimiento a partir de  $K \approx 500$ , obteniendo un nivel de adherencia de 0.930, 0.913 y 0.909 respectivamente. **IFCM** y **SLIC**, tienen un comportamiento similar, mejorando su nivel de adherencia con  $K \geq 500$ . **IFCM** tuvo el mejor rendimiento con respecto a la métrica *Rec*, su nivel de adherencia alcanzó 0.998. Por otro lado, **SLIC** obtuvo un rendimiento similar con un nivel de adherencia de 0.994. EL algoritmo **TP**, tiene un buen rendimiento, sin embargo, al igual que el algoritmo **LSC**, no consigue mejorar a partir de cierto punto.

El algoritmo **PB**, comienza a tener un buen rendimiento a partir de  $K \approx 550$ , obteniendo un nivel de adherencia de 0.910. Aunque **PB** tiene poca adaptabilidad a los bordes con valores de  $K \leq 550$ , logra superar a **LSC**, quien tuvo los mejores resultados con números de superpíxeles pequeños. Finalmente, **PB** obtuvo un nivel de adherencia de 0.986 con  $K \approx 1450$ . En la Figura 5.7, se puede ver la tendencia de los algoritmos, con los diferentes valores de superpíxeles.

## 5.4 Resultados con el conjuntos de datos Brain Web



**Figura 5.7:** Resultados de la evaluación con *Rec*, en el conjunto de datos BrainWeb T1.

Los resultados con respecto a la métrica *UE*, se muestran en la Tabla 5.7. *UE* cuantifica los píxeles que no logran adherirse a los bordes de la imagen. Los valores pequeños *UE*, denotan una mejor adaptación a los bordes de la imagen.

**Tabla 5.7:** *UE*, en el conjunto de datos BrainWeb T1; el nivel de adherencia de los algoritmos se muestra para cada valor de *K* considerado.

<i>K</i>	<i>UE</i>					
	<b>W</b>	<b>LSC</b>	<b>SLIC</b>	<b>IFCM</b>	<b>TP</b>	<b>PB</b>
200	0.121	0.057	0.086	0.099	0.097	0.159
300	0.087	0.056	0.075	0.087	0.095	0.132
400	0.079	0.056	0.069	0.078	0.076	0.127
500	0.075	0.056	0.063	0.069	0.074	0.114
550	0.069	0.058	0.061	0.069	0.074	0.098
700	0.065	0.06	0.057	0.063	0.075	0.089
800	0.065	0.061	0.057	0.063	0.075	0.089
850	0.065	0.061	0.057	0.059	0.075	0.089
900	0.065	0.064	0.057	0.059	0.075	0.089
1000	0.061	0.064	0.051	0.055	0.076	0.076
1200	0.061	0.073	0.051	0.055	0.076	0.076
1450	0.057	0.073	0.048	<b>0.046</b>	0.096	0.063

## 5.4 Resultados con el conjuntos de datos Brain Web

**IFCM** tiene un error considerablemente bajo con números de superpíxeles pequeños, alcanzado un error de 0.099 con  $K \approx 200$ . Conforme se aumenta el número de superpíxeles, el error disminuye, obteniendo el error más bajo, en comparación con el resto de los algoritmos, alcanzando un error de 0.046 con  $K \approx 1450$ .

Los algoritmos **SLIC** y **W** tuvieron un buen desempeño con respecto a *UE*, ocupando el segundo y tercer lugar en general, obteniendo 0.048 y 0.057 respectivamente, con  $K \approx 1450$ . La tendencia de **SLIC** y **W**, es similar al algoritmo **IFCM**.

Por otra parte, los algoritmos **LSC**, **TP** tiene un rendimiento medio. Aunque **LSC** obtuvo el error más bajo con números de superpíxeles pequeños, no logra mantener su rendimiento. **TP** obtuvo un error considerablemente bajo, sin embargo, tampoco logra mejorar conforme se aumenta el número de superpíxeles. El algoritmo **PB** tuvo el error más alto con números de superpíxeles pequeños. Conforme se incrementó el valor de  $K$ , **PB** fue disminuyendo el error. En general **PB**, **LSC** ocupan el cuarto y quinto lugar con 0.063 y 0.072 respectivamente con  $K \approx 1450$ . El algoritmo **TP** obtuvo el error más alto con 0.096.

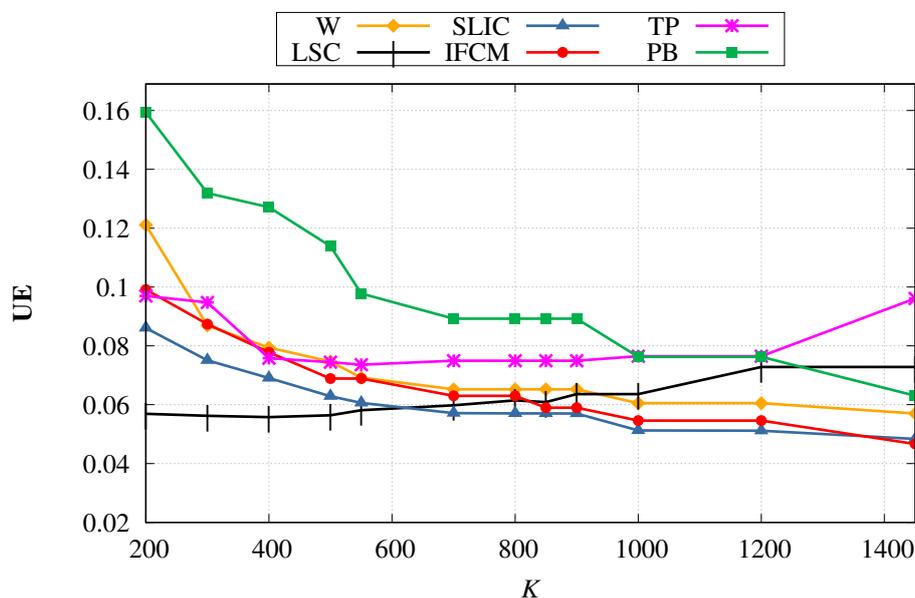


Figura 5.8: Resultados de la evaluación con *UE*, en el conjunto de datos BrainWeb T1.

## 5.4 Resultados con el conjuntos de datos Brain Web

En la Tabla 5.8 se muestran los resultados con respecto a la métrica  $EV$ . Para calcular  $EV$ , no se requieren de los segmentos *ground truth*, si no que se utiliza información local (superpíxeles) y global (imagen) para estimar la desviación de los píxeles que no se ajustan a los bordes naturales de la imagen. Por lo cual, la compacidad y regularidad de los superpíxeles resultantes influyen considerablemente en esta métrica. Todos los algoritmos tiene un buen rendimiento, para valores de superpíxeles grandes y pequeños.

El algoritmo **SLIC** obtuvo el mejor rendimiento, obteniendo un nivel de adherencia de 0.988 con  $K \approx 1450$ . Ocupando el primer lugar en general. Los algoritmos **IFCM** y **PB** ocupan el segundo y tercer lugar, con 0.987 y 0.80 respectivamente para  $K \approx 1450$ . Aunque **PB** obtuvo el menor rendimiento con 0.925 para  $K \approx 200$ , sin embargo, logra mejorar conforme se incrementa el valor de  $K$ . El algoritmo **PB** no ofrece el control de la compacidad mediante un parámetro como lo hacen los algoritmos antes mencionados, sino que el enfoque pseudo-booleano que posee le permite generar superpíxeles regulares en tamaño y forma.

**Tabla 5.8:**  $EV$ , en el conjunto de datos BrainWeb T1; el nivel de adherencia de los algoritmos se muestra para cada valor de  $K$  considerado.

K	$EV$					
	W	LSC	SLIC	IFCM	TP	PB
200	0.955	0.978	0.972	0.951	0.961	0.925
300	0.964	0.978	0.977	0.962	0.964	0.943
400	0.966	0.978	0.980	0.967	0.961	0.951
500	0.968	0.977	0.982	0.973	0.964	0.957
550	0.972	0.976	0.983	0.973	0.967	0.957
700	0.974	0.975	0.984	0.978	0.968	0.961
800	0.974	0.973	0.984	0.978	0.968	0.969
850	0.974	0.974	0.984	0.980	0.968	0.969
900	0.974	0.972	0.984	0.980	0.968	0.969
1000	0.977	0.972	0.986	0.982	0.966	0.974
1200	0.977	0.965	0.987	0.982	0.966	0.974
1450	0.979	0.965	0.988	0.987	0.953	0.980

**W** no tienen control sobre la compacidad por lo cual generan superpíxeles irregulares pero con alta adaptabilidad a los borde de la imagen. **W** ocupa el quinto lugar en general, con un nivel de adherencia de 0.979 para  $K \approx 1450$ .

## 5.4 Resultados con el conjuntos de datos Brain Web

El algoritmo **LSC** genera superpíxeles con alta regularidad pero no consigue mejorar la adaptabilidad a los bordes de la imagen con el número de superpíxeles considerados. **LSC** ocupa el quinto lugar en general, con un nivel de adherencia de 0.965 para  $K \approx 1450$ .

Finalmente, el algoritmo **TP** ocupa el ultimo lugar con respecto al resto de los algoritmos, con un nivel de adherencia de 0.953 para  $K \approx 1450$ . En la Figura 5.9, se muestra la tendencia de los algoritmos con los diferentes valores de  $K$ , para la métrica  $EV$ .

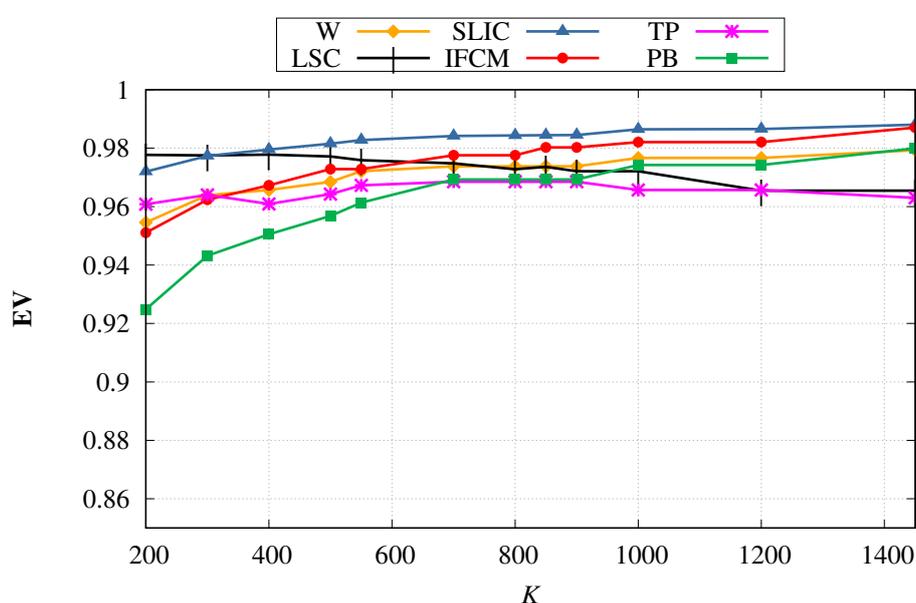


Figura 5.9: Resultados de la evaluación con  $EV$ , en el conjunto de datos BrainWeb T1..

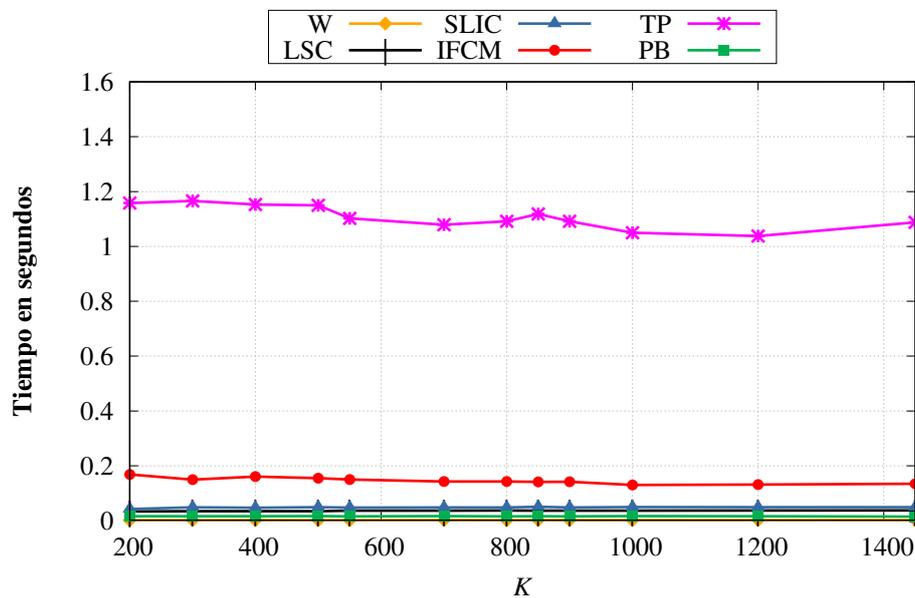
En cuanto al tiempo de ejecución, la comparación que se muestra en la Figura 5.10 y en la Tabla 5.10, se realizó con un procesador Intel Core i5 a 2.5 GHz con 4 GB de RAM. La mayoría de los algoritmos muestran un gran rendimiento al procesar imágenes con una resolución de  $181 \times 217$  píxeles, en tiempos de ejecución que no exceden al segundo. Con excepción del algoritmo **TP**, que obtuvo un tiempo de ejecución promedio de 1.107 segundos. El algoritmo con mejor rendimiento corresponde al algoritmo **W**, con tiempo de ejecución promedio de 0.002 segundos. Los algoritmo **PB**, **LSC** y **SLIC**, tiene aproximadamente el mismo rendimiento con un tiempo de ejecución promedio de 0.016, 0.036 y 0.048 segundos respectivamente. **IFCM** ocupa el quinto lugar en general, con un tiempo de ejecución promedio de 0.146 segundos.

## 5.4 Resultados con el conjuntos de datos Brain Web

En la Figura 5.10 se puede observar que la tendencia de estos algoritmos son lineales, es decir, el tiempo de ejecución no aumenta conforme se incrementa el número de superpíxeles.

**Tabla 5.9:** Tiempo de ejecución en segundos para diferentes valores superpíxeles  $K$  en el conjunto de datos de BrainWeb IRM T1.

$K$	W	LSC	SLIC	IFCM	TP	PB
200	0.002	0.033	0.042	0.169	1.158	0.016
300	0.002	0.035	0.049	0.150	1.166	0.016
400	0.002	0.035	0.047	0.161	1.153	0.016
500	0.002	0.035	0.049	0.155	1.150	0.017
550	0.002	0.037	0.048	0.150	1.102	0.016
700	0.002	0.036	0.048	0.143	1.079	0.017
800	0.002	0.037	0.048	0.143	1.092	0.016
850	0.002	0.037	0.051	0.141	1.119	0.016
900	0.002	0.037	0.048	0.142	1.092	0.016
1000	0.002	0.037	0.05	0.130	1.050	0.016
1200	0.002	0.037	0.049	0.132	1.038	0.016
1450	0.002	0.037	0.049	0.134	1.088	0.015

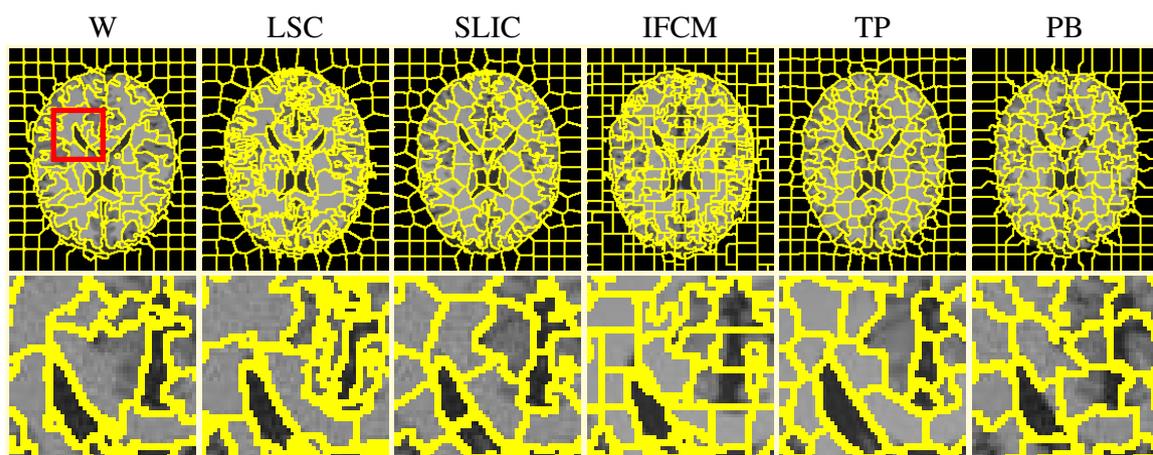


**Figura 5.10:** Resultados del tiempo de ejecución en segundos, en el conjunto de datos BrainWeb T1.

### Cualitativo

La calidad visual de los superpíxeles se determina considerando algunas características deseables en estos algoritmos, como la compacidad y la regularidad. En el trabajo de (Schick et al., 2014), la compacidad se define como el área que cubre un superpíxel, cuando los superpíxeles tienen una forma regular captan mejor la información de los límites. La regularidad se refiere al tamaño de los superpíxeles. Los superpíxeles se consideran regulares cuando tienen tamaños similares. Los algoritmos considerados para la evaluación, descritos en la Sección 3.2, utilizan diferentes enfoques para generar superpíxeles. **SLIC** y **LSC**, se basan en algoritmos de agrupamiento al igual que el algoritmo propuesto. **PB** utiliza un enfoque con grafos, **TP** se basa en la evolución de contornos y **W** utiliza el algoritmo de cuenca hidrográfica.

Los algoritmos basados en agrupamiento, muestran una representación bastante aproximada a la imagen original. Los superpíxeles se adhieren bien a los bordes de la imagen, con valores de  $K$  considerablemente pequeños ( $200 \leq K \leq 700$ ). Los algoritmos **W**, **TP** y **PB**, tienen un comportamiento similar. La mayoría de los superpíxeles logra adaptarse bien a los bordes de la imagen. Los superpíxeles que no se adhieren completamente a los bordes de la imagen, causa que se pierdan detalles, produciendo finalmente una representación regular de la imagen original. La Figura 5.11, muestra los resultados de la extracción de superpíxeles con imágenes de RM cerebral para  $K \approx 200$ .



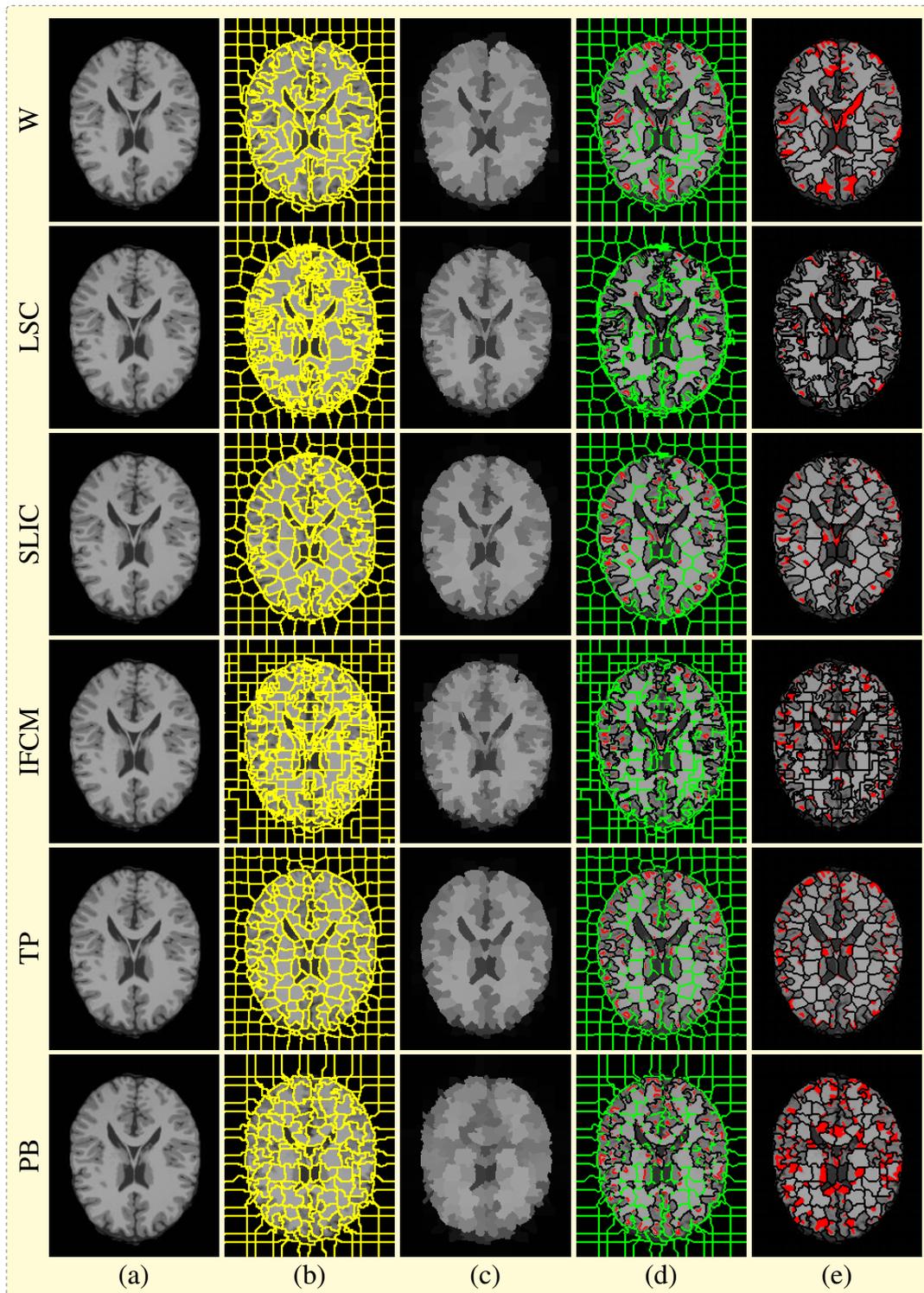
**Figura 5.11:** Extracción de superpíxeles con el conjunto de datos de BrainWeb; se generaron aproximadamente 200 superpíxeles, en cada columna se muestra el resultado del algoritmo y una ampliación a una región.

## 5.4 Resultados con el conjuntos de datos Brain Web

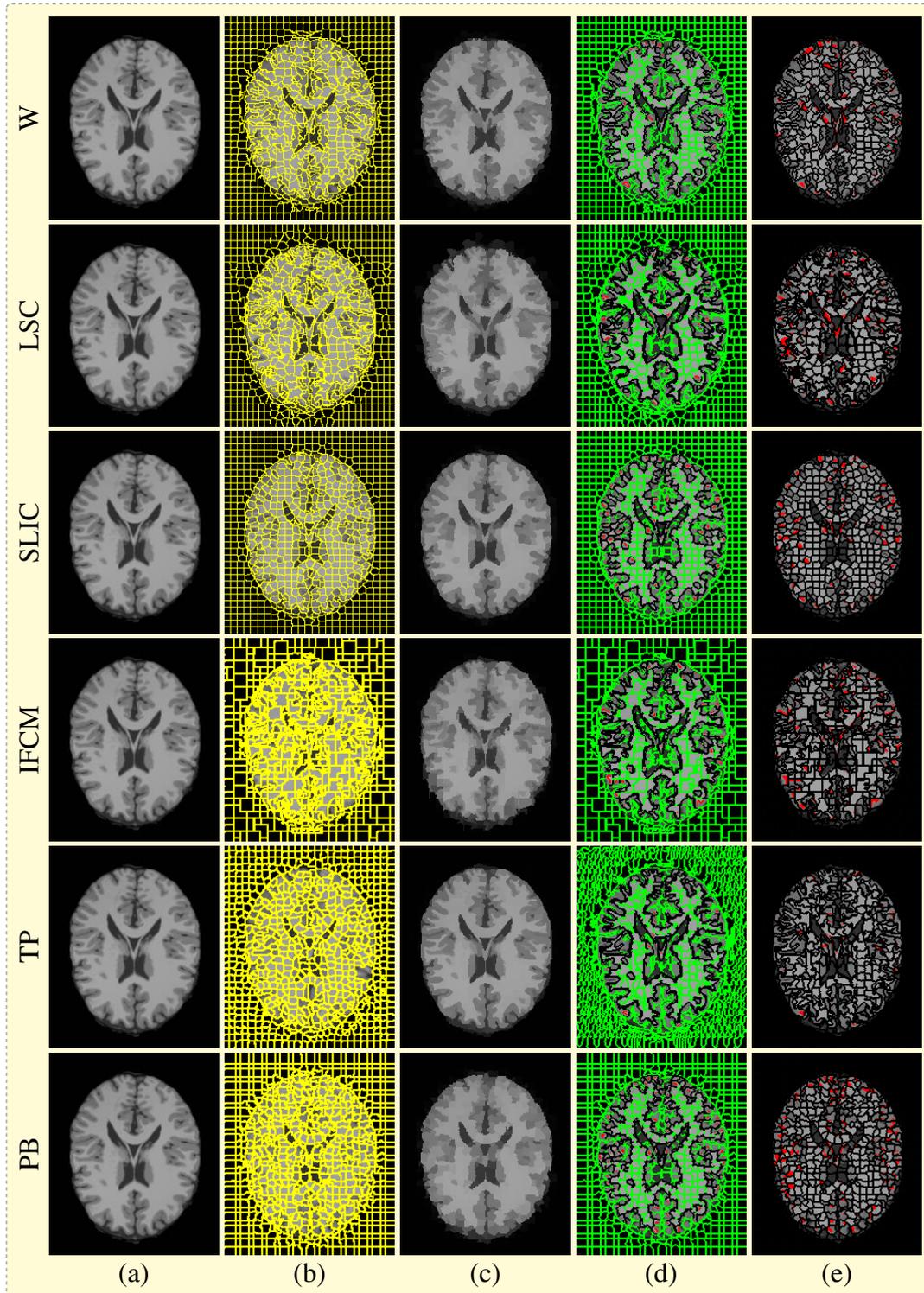
---

Los superpíxeles generados por **SLIC** son más regulares en comparación con el resto de los algoritmos, lo cual permite tener una apariencia mejor pero no logra preservar algunos detalles de la imagen original. En la Figura 5.12 se muestra los resultados de  $K \approx 200$  con más detalle al incluir las diferentes representaciones de los superpíxeles generados (contornos e intensidad media de los segmentos). También se muestra la comparación con el segmento *ground truth*, correspondiente, que se utilizó para calcular su nivel de adherencia. Por otra parte **IFCM**, **LSC** generan superpíxeles irregulares, permitiéndoles preservar más detalles, como se ve en la representación de los superpíxeles con la intensidad media (ver Figura 5.12–(c)). El algoritmo **W** genera superpíxeles considerablemente regulares, consiguiendo adaptarse bien en algunas regiones de la imagen, aunque en otras pierde detalles significativos. En cuanto a los algoritmos **PB** y **TP**, logran adaptarse al contorno de la imagen pero pierden detalles de las regiones del cerebro. **TP** fue el algoritmo que tuvo menor rendimiento ya que fue el que captó menos detalles. Conforme se va incrementado el número de superpíxeles  $K$  el rendimiento de los algoritmos mejoran considerablemente logrando preservar mejor los detalles de la imagen.

Con  $K \approx 800$  se tiene una mejor aproximación de la imagen como se ve en la Figura 5.13. A partir de este punto es difícil distinguir el algoritmo con mejor rendimiento. En la Figura 5.13 –(c), la representación con la intensidad media no es de gran apoyo ya que muestra resultados similares, en cambio la Figura 5.13 –(e) ofrece una mejor percepción de los resultados al mostrar la comparación con el segmento *ground truth* correspondiente, donde las regiones en color rojo muestran los superpíxeles que no logran adaptarse correctamente. Con esta representación se puede observar que **IFCM** tiene el mayor rendimiento con superpíxeles regulares, mostrando menos superpíxeles desviados con respecto al segmento *ground truth*. **LSC**, **SLIC** muestran rendimientos similares, teniendo pocos superpíxeles desviados. Los algoritmos **PB** y **TP**, son los que tienen el menor rendimiento ya que muestra la mayor cantidad de segmentos desviados. La compacidad y la regularidad de los superpíxeles son características deseables, aunque el algoritmo **IFCM** genera superpíxeles poco regulares logra obtener una buena aproximación de la imagen original con valores de  $K$  grandes, teniendo en cuenta que el objetivo principal de estos algoritmos es ofrecer una representación simplificada de la imagen original; se puede considerar que el algoritmo es eficiente teniendo un rendimiento similares a los algoritmos populares en el estado del arte como **SLIC**, **LSC** y **W**.



**Figura 5.12:** Resultados cualitativos de la evaluación, en el conjunto de datos de BrainWeb T1 con  $K \approx 200$ ; en la columna (a) se muestra la imagen original, (b) la representación con contornos, (c) representación con la intensidad media de los superpíxeles, (d) comparación con el segmento *ground truth* correspondiente para la métrica *Rec.* y (e) comparación con el segmento *ground truth* correspondiente para la métrica *UE*.



**Figura 5.13:** Resultados cualitativos de la evaluación, en el conjunto de datos de BrainWeb T1 con  $K \approx 800$ .; en la columna (a) se muestra la imagen original, (b) la representación con contornos, (c) representación con la intensidad media de los superpíxeles, (d) comparación con el segmento *ground truth* correspondiente para la métrica *Rec*. y (e) comparación con el segmento *ground truth* correspondiente para la métrica *UE*.

### 5.4.2. RM del cerebro en modo T1 con 5% de ruido

En esta prueba se evalúa el rendimiento de los algoritmos para extraer superpíxeles con imágenes de resonancia magnética cerebral con ruido. El estudio utilizado, se simuló en BrainWeb con las mismas características de modalidad, grosor del corte e intensidad no uniforme, del estudio utilizado en la prueba anterior, agregando únicamente 5% de ruido gaussiano.

#### Cuantitativo

Las pruebas se realizaron con la misma configuración de superpíxeles  $K$  como la prueba anterior. Los resultados obtenidos con respecto a la métrica  $Rec$ , se muestran en la Tabla 5.10, el valor en negritas muestra el mejor resultado. El algoritmo **IFCM** obtuvo un nivel de adherencia de 0.998, manteniendo su rendimiento con respecto a la prueba anterior. **IFCM** ocupa el primer lugar en general en esta métrica.

**Tabla 5.10:**  $Rec$ , en el conjunto de datos BrainWeb T1 + 5% ruido; el nivel de adherencia de los algoritmos se muestra para cada valor de  $K$  considerado.

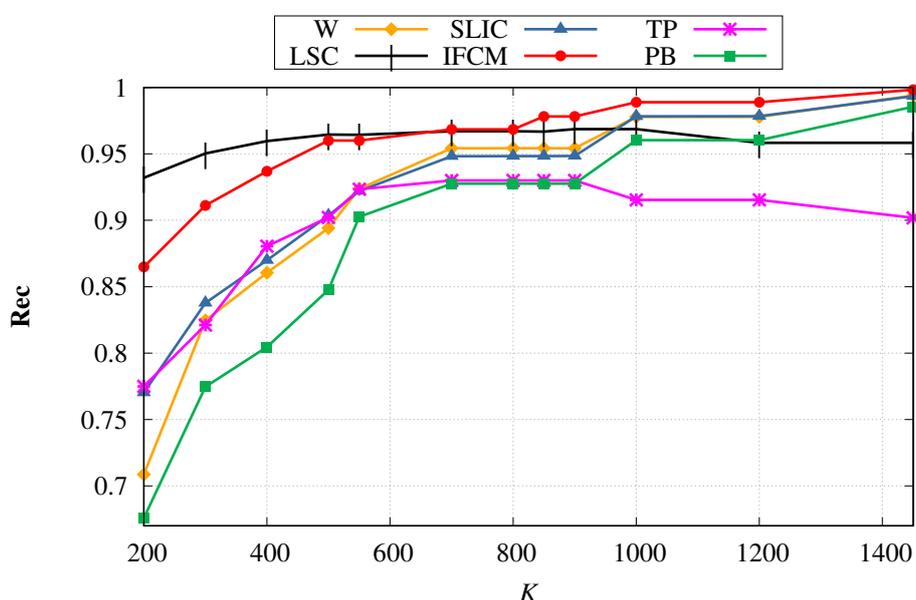
$K$	$Rec$					
	<b>W</b>	<b>LSC</b>	<b>SLIC</b>	<b>IFCM</b>	<b>TP</b>	<b>PB</b>
200	0.709	0.932	0.770	0.779	0.775	0.676
300	0.825	0.950	0.838	0.838	0.821	0.775
400	0.861	0.960	0.870	0.879	0.881	0.805
500	0.894	0.964	0.904	0.904	0.902	0.848
550	0.924	0.964	0.922	0.917	0.923	0.903
700	0.954	0.967	0.948	0.936	0.930	0.928
800	0.954	0.967	0.948	0.968	0.930	0.928
850	0.954	0.967	0.948	0.978	0.930	0.928
900	0.954	0.969	0.948	0.978	0.930	0.928
1000	0.978	0.969	0.978	0.989	0.915	0.960
1200	0.978	0.958	0.978	0.989	0.915	0.960
1450	0.994	0.958	0.993	<b>0.998</b>	0.902	0.986

**W** y **SLIC** muestran un rendimiento similar a pesar de que utilizan diferentes enfoques para generar los superpíxeles. En general ocupan el segundo y tercer lugar con un nivel de adherencia de 0.994 y 0.993 respectivamente. **PB** y **LSC** ocupan el cuarto y quinto lugar con un nivel de adherencia de 0.986 y 0.958 respectivamente.

## 5.4 Resultados con el conjuntos de datos Brain Web

Aunque **LSC** produce buenos resultados con números de superpíxeles pequeños, no logra mejorar significativamente conforme aumenta el valor de  $K$ , siendo superado por **PB**. **TP** ocupa el ultimo lugar con un nivel de adherencia de 0.902. **TP** tiene un rendimiento similar a los algoritmos **W**, **SLIC** y **IFCM** para números de superpíxeles pequeños. A partir de cierto punto ( $K \approx 700$ ) no logra mejorar su rendimiento.

El rendimiento de los algoritmos disminuye en comparación con el experimento anterior, esto se puede apreciar mejor en la Figura 5.14, donde se observa la tendencia de los algoritmos en los diferentes valores de  $K$  para esta métrica.



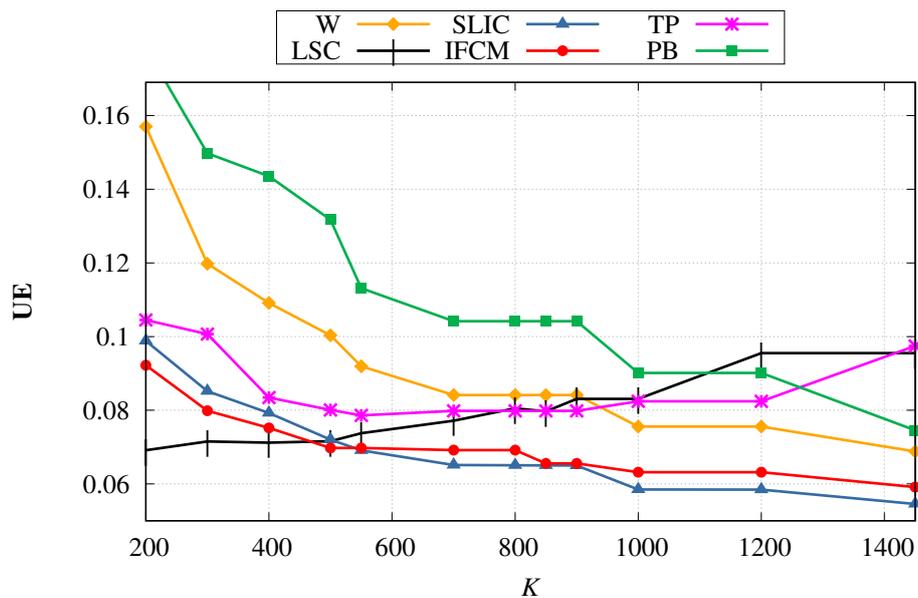
**Figura 5.14:** Resultados de la evaluación con  $Rec$ , en el conjunto de datos BrainWeb T1 + 5% de ruido.

Los resultados con respecto a la métrica  $UE$  se muestran en la Tabla 5.11. Los algoritmos **SLIC**, **IFCM** y **W**, al igual que en los resultados de las pruebas anteriores, muestran un rendimiento similar. **SLIC** produjo el menor error, obteniendo 0.055, por lo cual ocupa el primer lugar en general con respecto a la métrica  $UE$ . **IFCM** y **W** ocupan el segundo y tercer lugar con un error de 0.059 y 0.069 respectivamente. Los algoritmos **PB**, **LSC** y **TP** ocupan el cuarto, quinto y sexto lugar con un error de 0.075, 0.096 y 0.097 respectivamente. En Figura 5.15 se puede observar la tendencia de los algoritmos.

## 5.4 Resultados con el conjuntos de datos Brain Web

**Tabla 5.11:**  $UE$ , en el conjunto de datos BrainWeb T1 + 5% ruido; el nivel de adherencia de los algoritmos se muestra para cada valor de  $K$  considerado.

$K$	$UE$					
	W	LSC	SLIC	IFCM	TP	PB
200	0.157	0.069	0.099	0.092	0.104	0.178
300	0.12	0.072	0.085	0.08	0.101	0.15
400	0.109	0.071	0.079	0.075	0.083	0.144
500	0.101	0.072	0.072	0.070	0.080	0.132
550	0.092	0.074	0.069	0.070	0.079	0.113
700	0.084	0.077	0.065	0.069	0.080	0.104
800	0.084	0.081	0.065	0.069	0.080	0.104
850	0.084	0.08	0.065	0.066	0.080	0.104
900	0.084	0.083	0.065	0.066	0.080	0.104
1000	0.076	0.083	0.058	0.063	0.082	0.090
1200	0.076	0.096	0.058	0.063	0.082	0.090
1450	0.069	0.096	<b>0.055</b>	0.059	0.097	0.075



**Figura 5.15:** Resultados de la evaluación con  $UE$ , en el conjunto de datos BrainWeb T1 + 5% de ruido.

## 5.4 Resultados con el conjuntos de datos Brain Web

Los resultados para la métrica *EV*, se muestran en la Tabla 5.12, el valor en negrita indica el mejor resultado. Los algoritmos no muestran un cambio significativo con respecto a los resultados de la prueba anterior. Se observa que a pesar de utilizar imágenes con ruido, los algoritmos pueden alcanzar un nivel de adherencia mayor a 0.9. Esto puede deberse a que *EV* no utiliza el segmento *ground truth* para calcularse.

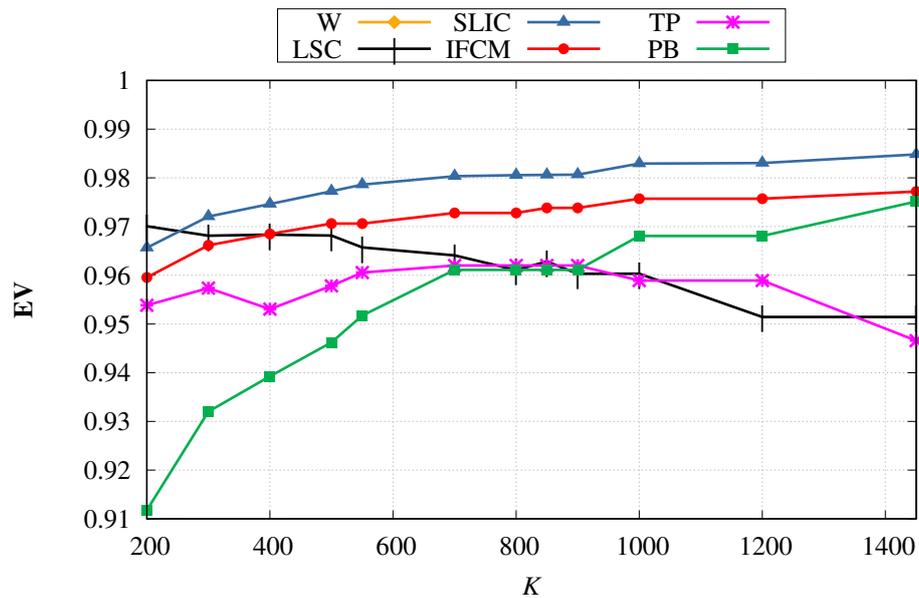
El primer lugar lo ocupa el algoritmo **SLIC** con un nivel de adherencia de 0.985. El segundo lugar lo ocupa **IFCM** con 0.977, seguido de **PB** con 0.975. El cuarto, quinto y sexto lugar lo ocupan **W**, **LSC** y **TP** con 0.973, 0.951 y 0.977 respectivamente. En la Figura 5.16 se puede observar el rendimiento.

**Tabla 5.12:** *EV*, en el conjunto de datos BrainWeb T1 + 5% ruido; el nivel de adherencia de los algoritmos se muestra para cada valor de *K* considerado.

<i>EV</i>						
<b>K</b>	<b>W</b>	<b>LSC</b>	<b>SLIC</b>	<b>IFCM</b>	<b>TP</b>	<b>PB</b>
200	0.939	0.97	0.966	0.96	0.954	0.912
300	0.951	0.968	0.972	0.966	0.957	0.932
400	0.954	0.968	0.975	0.968	0.953	0.939
500	0.958	0.968	0.977	0.971	0.958	0.946
550	0.963	0.966	0.979	0.971	0.961	0.952
700	0.965	0.964	0.98	0.973	0.962	0.961
800	0.965	0.961	0.981	0.973	0.962	0.961
850	0.965	0.963	0.981	0.974	0.962	0.961
900	0.965	0.960	0.981	0.974	0.962	0.961
1000	0.969	0.960	0.983	0.976	0.959	0.968
1200	0.969	0.951	0.983	0.976	0.959	0.968
1450	0.973	0.951	<b>0.985</b>	0.977	0.947	0.975

En cuanto al tiempo de ejecución, no hay un cambio significativo, teniendo un rendimiento similar a la prueba anterior. En la Tabla 5.13 se muestran los tiempos de ejecución para cada algoritmo. En la Figura 5.17 se muestra gráficamente.

## 5.4 Resultados con el conjuntos de datos Brain Web

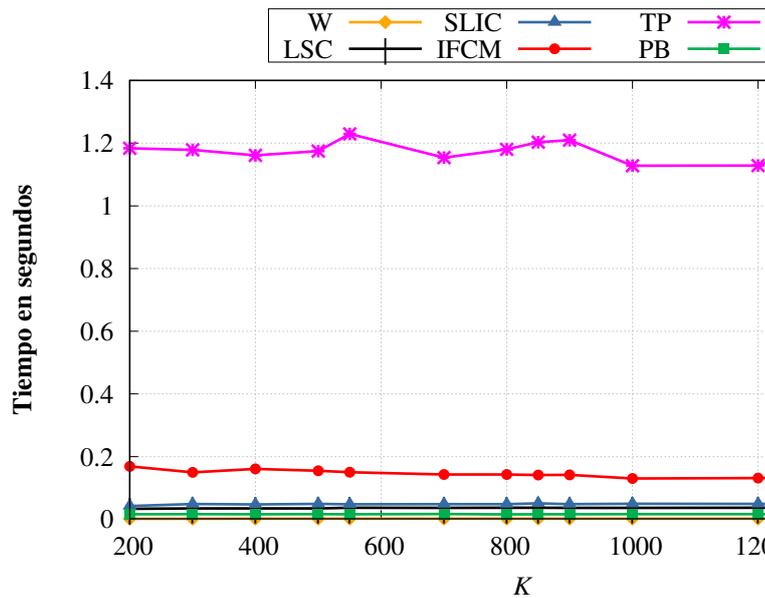


**Figura 5.16:** Resultados de la evaluación con  $EV$ , en el conjunto de datos BrainWeb T1 + 5% de ruido.

**Tabla 5.13:** Tiempo de ejecución en segundos para diferentes valores superpíxeles  $K$  en el conjunto de datos de BrainWeb IRM T1 + 5% de ruido.

$K$	W	LSC	SLIC	IFCM	TP	PB
200	0.002	0.033	0.046	0.169	1.184	0.014
300	0.002	0.036	0.049	0.150	1.179	0.015
400	0.002	0.036	0.047	0.161	1.261	0.015
500	0.002	0.036	0.049	0.155	1.374	0.014
550	0.002	0.037	0.048	0.150	1.229	0.015
700	0.002	0.036	0.048	0.143	1.154	0.017
800	0.002	0.037	0.045	0.143	1.180	0.015
850	0.002	0.038	0.051	0.141	1.203	0.014
900	0.002	0.038	0.048	0.142	1.210	0.015
1000	0.002	0.039	0.050	0.130	1.128	0.015
1200	0.002	0.038	0.049	0.132	1.128	0.015
1450	0.002	0.038	0.049	0.134	1.306	0.014

## 5.4 Resultados con el conjuntos de datos Brain Web

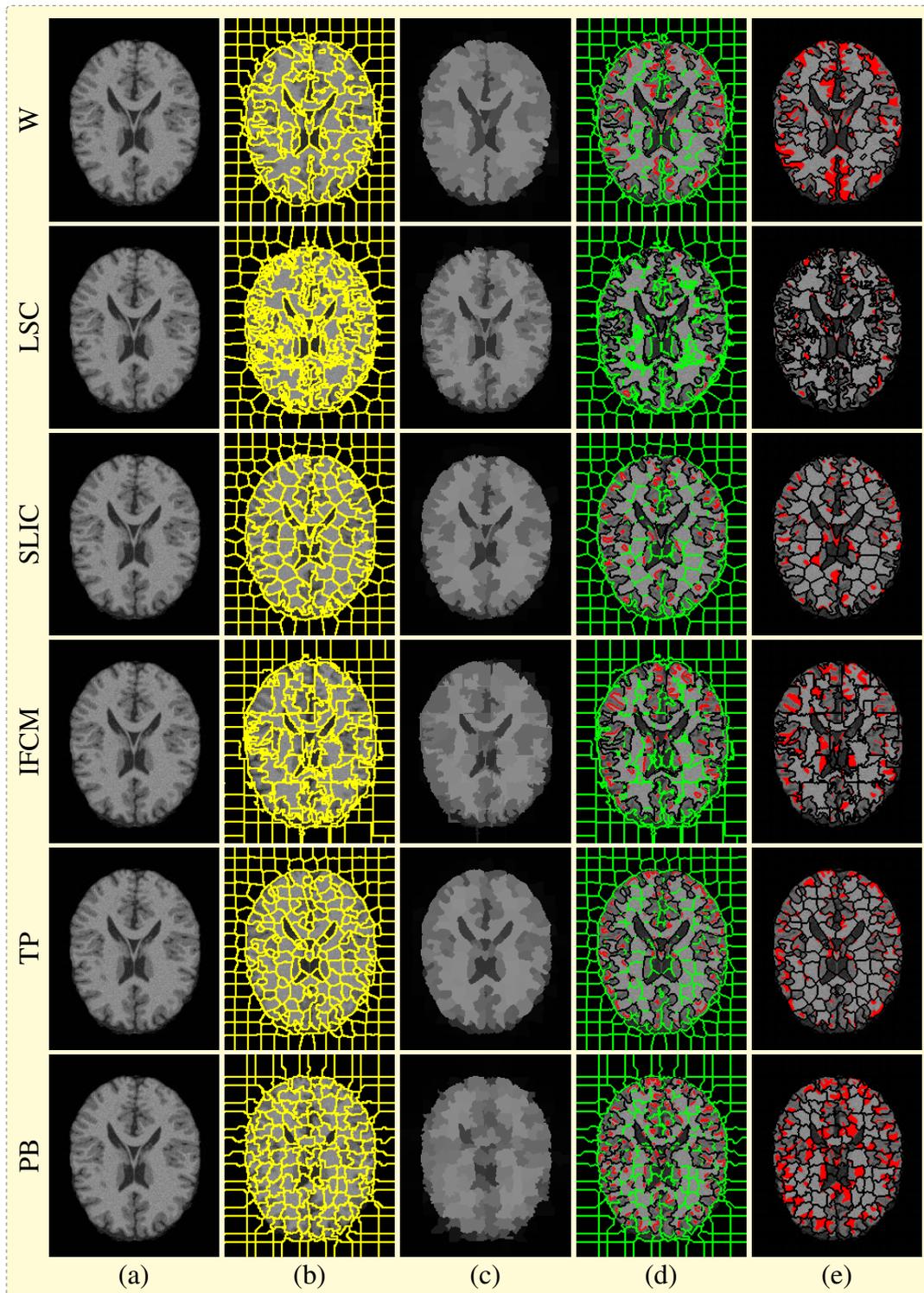


**Figura 5.17:** Resultados del tiempo de ejecución en segundos, en el conjunto de datos BrainWeb T1 + 5% de ruido.

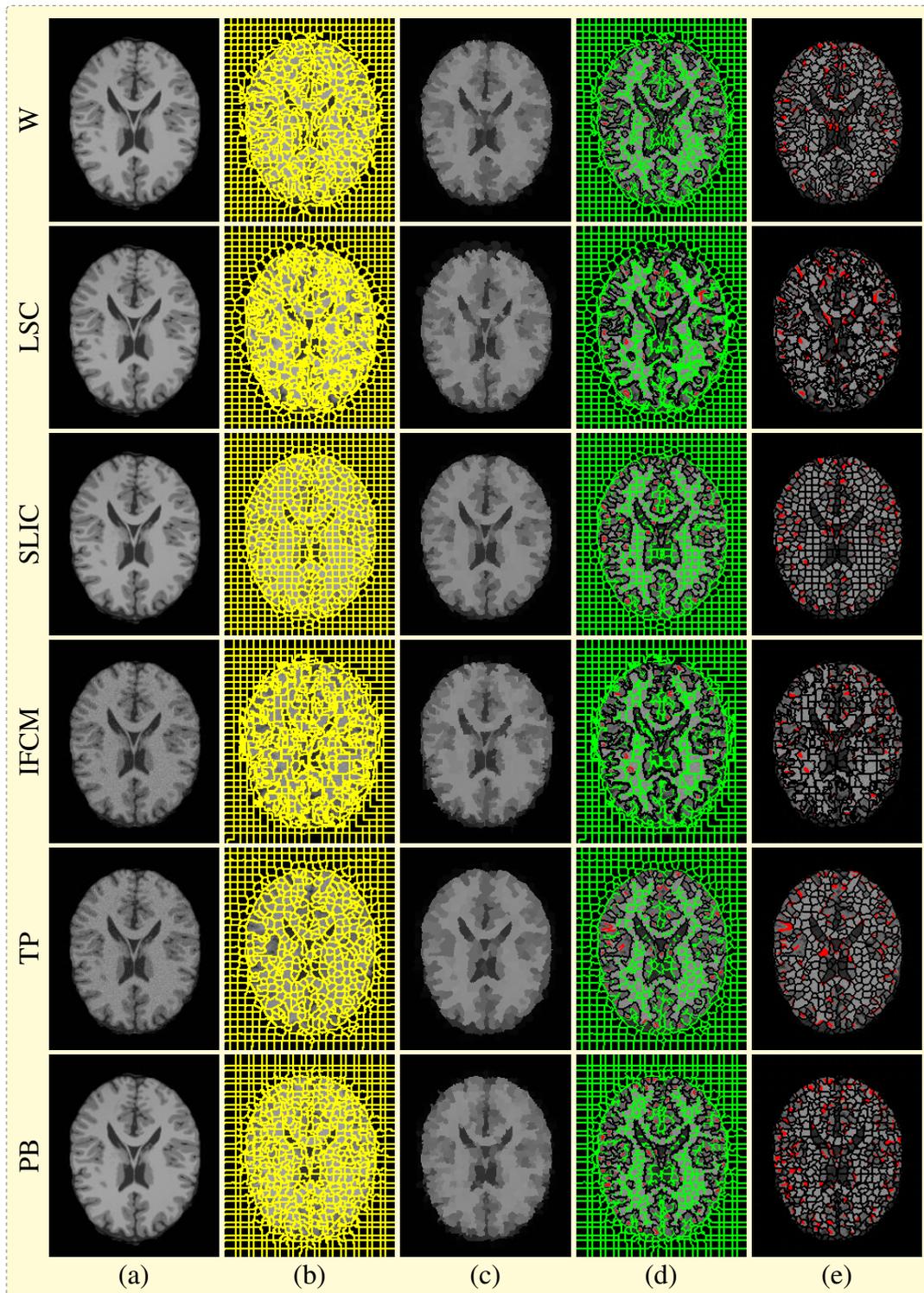
### Cualitativo

Los algoritmos considerados proporcionan una buena adherencia a los límites naturales de la imagen con números de superpíxeles grandes. Los algoritmos **SLIC**, **LSC** y **TP**, producen superpíxeles regulares y compactos. **LSC** produce una aproximación sólida de la imagen para números de superpíxeles pequeños. La adaptación a los bordes de la imagen mejora para los algoritmos **SLIC** y **TP** conforme se incrementa el valor de  $K$ . Sin embargo **TP** no logra mejorar a partir de  $K \geq 550$ . Los superpíxeles generados por **SLIC** logran preservar su compacidad a pesar de utilizar imágenes con ruido.

Los algoritmos **IFCM**, **W** y **PB** generan superpíxeles poco regulares, por lo cual la compacidad de los superpíxeles producidos son variantes. **IFCM** y **W** logran un nivel de adherencia buena, logrando preservar pequeños detalles de la imagen. En las Figuras 5.18 y 5.19 se pueden observar los resultados conceptuales.



**Figura 5.18:** Resultados cualitativos de la evaluación, en el conjunto de datos de BrainWeb T1 + 5% de ruido, con  $K \approx 200$ ; en la columna (a) se muestra la imagen original, (b) la representación con contornos, (c) representación con la intensidad media de los superpíxeles, (d) comparación con el segmento *ground truth* correspondiente para la métrica *Rec.* y (e) comparación con el segmento *ground truth* correspondiente para la métrica *UE*.



**Figura 5.19:** Resultados cualitativos de la evaluación, en el conjunto de datos de BrainWeb T1 + 5% de ruido, con  $K \approx 800$ ; en la columna (a) se muestra la imagen original, (b) la representación con contornos, (c) representación con la intensidad media de los superpíxeles, (d) comparación con el segmento *ground truth* correspondiente para la métrica *Rec.* y (e) comparación con el segmento *ground truth* correspondiente para la métrica *UE*.

### 5.4.3. Resultados con BSDS500

Para esta evaluación se utilizó BSDS500-test compuesta por 200 imágenes con una resolución de  $481 \times 321$  píxeles o  $321 \times 481$ . El objetivo de esta prueba es ver el rendimiento del algoritmo propuesto con imágenes a color.

#### Cuantitativo

Las pruebas se realizaron con la misma configuración de superpíxeles  $K$  como en las dos pruebas anteriores. Los resultados obtenidos con respecto a la métrica  $Rec$ , se muestran en la Tabla 5.14, el valor en negritas muestra el mejor resultado. El algoritmo **IFCM** obtuvo el mejor resultado en esta métrica, con un nivel de adherencia de 0.902, manteniendo su rendimiento con respecto a las dos pruebas anteriores.

El segundo mejor resultado fue obtenido por el algoritmo **LSC**, logrando un nivel de adherencia a los bordes de 0.883. A diferencia de las pruebas anteriores, **LSC** logra mejorar su rendimiento con forme se incrementa el valor de  $K$ . Se puede observar un cambio en el algoritmo **LSC** con imágenes a color, ya que muestra una tendencia creciente con respecto al número de superpíxeles.

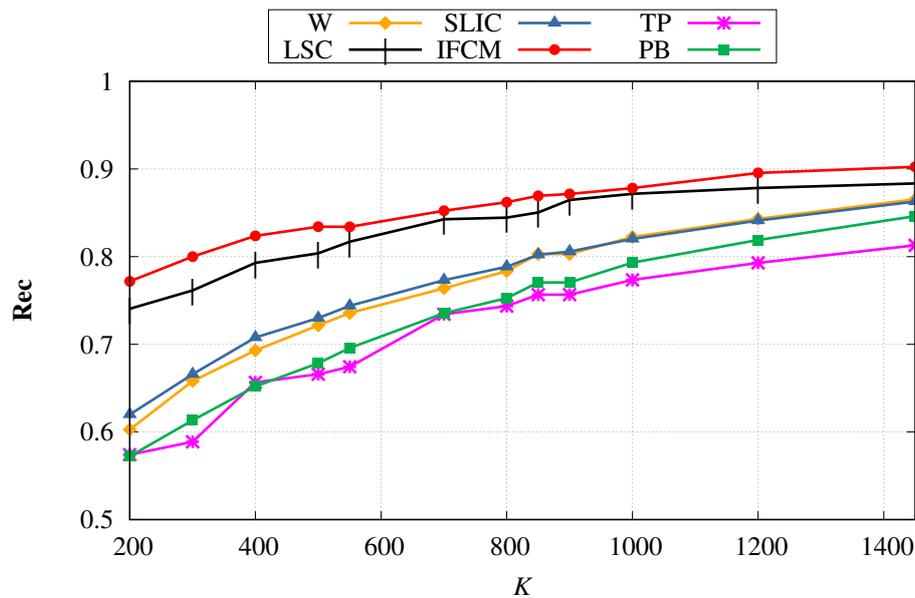
**Tabla 5.14:** Resultados de la evaluación con  $Rec$ , en el conjunto de datos BSDS500.

$K$	$Rec$					
	<b>W</b>	<b>LSC</b>	<b>SLIC</b>	<b>IFCM</b>	<b>TP</b>	<b>PB</b>
200	0.603	0.74	0.62	0.772	0.574	0.572
300	0.658	0.762	0.666	0.801	0.589	0.613
400	0.693	0.793	0.708	0.824	0.657	0.652
500	0.721	0.804	0.73	0.834	0.666	0.678
550	0.736	0.817	0.744	0.834	0.674	0.696
700	0.764	0.843	0.773	0.852	0.734	0.735
800	0.783	0.844	0.788	0.862	0.744	0.753
850	0.803	0.85	0.802	0.869	0.757	0.771
900	0.803	0.865	0.806	0.871	0.757	0.771
1000	0.822	0.872	0.82	0.878	0.773	0.793
1200	0.843	0.878	0.841	0.895	0.793	0.819
1450	0.865	0.883	0.863	<b>0.902</b>	0.813	0.846

Los algoritmos **W** y **SLIC** muestran un rendimiento similar, en general ocupan el tercer y cuarto lugar con un nivel de adherencia de 0.865 y 0.863 respectivamente.

## 5.4 Resultados con el conjuntos de datos Brain Web

**PB** y **TP** ocupan el quinto y sexto lugar con un nivel de adherencia de 0.846 y 0.813 respectivamente. **PB** muestra un rendimiento similar a los algoritmos **W** y **SLIC**, a partir de valores de ( $K \geq 700$ ). En la Figura 5.20, donde se observa la tendencia de los algoritmos para este conjunto de datos con respecto a la métrica *Rec*.



**Figura 5.20:** Resultados de la evaluación con *Rec*, en el conjunto de datos BSDS500.

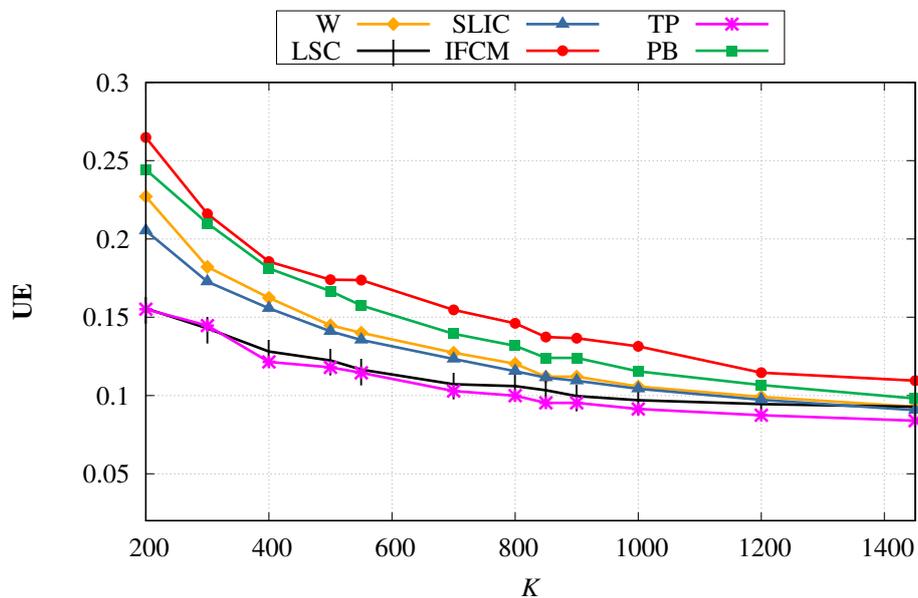
Los resultados con respecto a la métrica *UE* se muestran en la Tabla 5.15. El algoritmo **TP** obtuvo el mejor rendimiento con respecto a esta métrica, con un error de 0.089. **SLIC** obtuvo el segundo mejor resultado, con error de 0.091. Los algoritmos **W** y **LSC**, obtuvieron un error de 0.093, por lo cual ambos algoritmos ocupan el tercer lugar.

**PB** y **IFCM** ocupan el quinto y sexto lugar, con un error de 0.098 y 0.109 respectivamente. Los algoritmos **SLIC**, **W**, **LSC** y **PB**, muestran un rendimiento similar. En Figura 5.21 se puede observar la tendencia de los algoritmos con respecto a la métrica *UE*.

## 5.4 Resultados con el conjuntos de datos Brain Web

**Tabla 5.15:** Resultados de la evaluación con  $UE$ , en el conjunto de datos BSDS500.

$K$	$UE$					
	W	LSC	SLIC	IFCM	TP	PB
200	0.227	0.156	0.205	0.265	0.155	0.244
300	0.182	0.143	0.173	0.216	0.145	0.210
400	0.162	0.128	0.156	0.186	0.121	0.181
500	0.145	0.122	0.141	0.174	0.118	0.167
550	0.140	0.117	0.136	0.174	0.115	0.158
700	0.127	0.107	0.123	0.155	0.103	0.139
800	0.120	0.106	0.116	0.146	0.100	0.132
850	0.112	0.103	0.111	0.137	0.095	0.124
900	0.112	0.101	0.109	0.137	0.095	0.124
1000	0.106	0.097	0.104	0.131	0.091	0.115
1200	0.099	0.095	0.097	0.115	0.087	0.107
1450	0.093	0.093	0.091	0.109	<b>0.084</b>	0.098



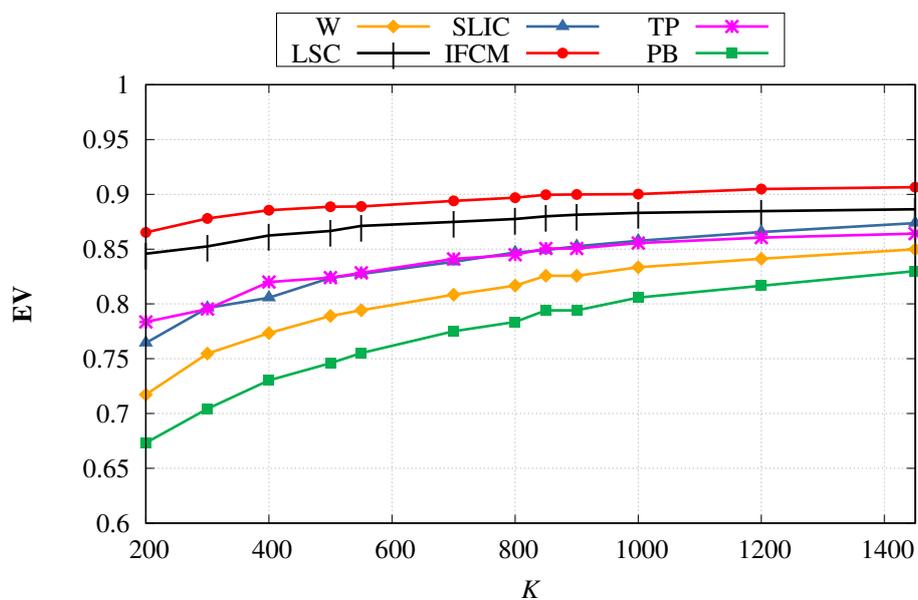
**Figura 5.21:** Resultados de la evaluación con  $UE$ , en el conjunto de datos BSDS500.

## 5.4 Resultados con el conjuntos de datos Brain Web

Los resultados para la métrica  $EV$ , se muestran en la Tabla 5.16, el valor en negrita indica el mejor resultado. Los resultados de  $EV$  son muy similares a los obtenidos por la métrica  $Rec.$ . El primer lugar lo ocupa el algoritmo **IFCM** con un nivel de adherencia de 0.907. El segundo lugar lo ocupa **LSC** con 0.886, seguido de **SLIC** con 0.874. El cuarto, quinto y sexto lugar lo ocupan **TP**, **W** y **PB** con 0.864, 0.850 y 0.830 respectivamente. En la Figura 5.22 se muestra el rendimiento.

**Tabla 5.16:** Resultados de la evaluación con  $EV$ , en el conjunto de datos BSDS500

$EV$						
K	W	LSC	SLIC	IFCM	TP	PB
200	0.717	0.846	0.764	0.865	0.784	0.673
300	0.755	0.853	0.796	0.878	0.795	0.704
400	0.773	0.862	0.806	0.885	0.820	0.730
500	0.789	0.867	0.824	0.889	0.824	0.746
550	0.794	0.871	0.827	0.889	0.828	0.755
700	0.808	0.875	0.838	0.894	0.841	0.775
800	0.817	0.878	0.847	0.897	0.845	0.784
850	0.826	0.880	0.849	0.901	0.851	0.794
900	0.826	0.881	0.853	0.901	0.851	0.794
1000	0.833	0.883	0.858	0.901	0.856	0.806
1200	0.841	0.885	0.866	0.905	0.861	0.817
1450	0.850	0.886	0.874	<b>0.907</b>	0.864	0.830



**Figura 5.22:** Resultados de la evaluación con  $EV$ , en el conjunto de datos BSDS500.

## 5.4 Resultados con el conjuntos de datos Brain Web

Los resultados para el tiempo de ejecución se muestran en la Tabla 5.17. Todos los algoritmos presentaron un aumento de tiempo, debido a que la resolución de la imagen aumente de  $181 \times 217$  a  $481 \times 321$  píxeles. Los tiempos de ejecución en la mayoría de los algoritmos no exceden al segundo, a excepción de los algoritmos **IFCM** y **TP** que obtuvieron un tiempo de ejecución promedio de 1.133 y 1.530 respectivamente.

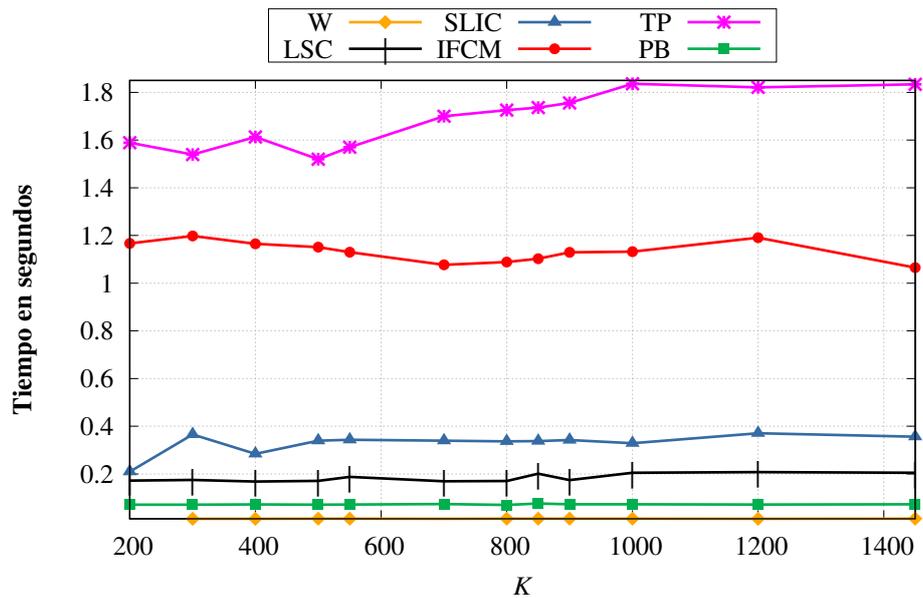
El algoritmo con mejor rendimiento corresponde al algoritmo **W**, con tiempo de ejecución promedio de 0.012 segundos. Los algoritmo **PB**, **LSC** y **SLIC**, tiene aproximadamente el mismo rendimiento con un tiempo de ejecución promedio de 0.072, 0.184 y 0.330 segundos respectivamente. A diferencia de la prueba con imágenes de resonancia magnética con 5% de ruido, donde el tiempo no cambia significativamente, los tiempos de ejecución obtenidos con el conjunto de datos BSDS500 se incrementan.

**Tabla 5.17:** Tiempos de ejecución en segundos para diferentes valores superpíxeles  $K$  en el conjunto de datos BSDS500

$K$	<b>W</b>	<b>LSC</b>	<b>SLIC</b>	<b>IFCM</b>	<b>TP</b>	<b>PB</b>
200	0.011	0.172	0.21	1.167	1.589	0.071
300	0.012	0.175	0.366	1.198	1.821	0.071
400	0.012	0.168	0.284	1.165	1.494	0.072
500	0.012	0.171	0.339	1.151	1.472	0.071
550	0.012	0.188	0.343	1.130	1.570	0.072
700	0.012	0.169	0.339	1.077	1.700	0.074
800	0.013	0.170	0.337	1.089	1.726	0.070
850	0.013	0.201	0.338	1.103	1.354	0.076
900	0.013	0.174	0.342	1.129	1.066	0.073
1000	0.012	0.205	0.329	1.132	1.182	0.073
1200	0.012	0.208	0.371	1.190	1.552	0.072
1450	<b>0.012</b>	0.205	0.356	1.065	1.834	0.073

En la Figura 5.23 se muestra graficamente el tiempo de ejecución obtenido para este conjunto de datos.

## 5.4 Resultados con el conjuntos de datos Brain Web



**Figura 5.23:** Resultados del tiempo de ejecución en segundos, en el conjunto de datos BSDS500.

### Cualitativo

La calidad visual de los superpíxeles obtenidos con el conjunto de datos BSDS500 se ve influenciada por el número de superpíxeles. Los algoritmos consiguen mejorar su adherencia a los bordes de la imagen, especialmente con valores de superpíxeles grandes. Los algoritmos notables que generan superpíxeles regulares y compactos son **SLIC**, **LSC** y **TP**. El algoritmo **PB**, a pesar de que no cuenta con un parámetro para controlar la compacidad, logra producir superpíxeles aproximadamente del mismo tamaño y forma.

Los algoritmos **IFCM** y **W**, generan superpíxeles irregulares, con alta adherencia a los bordes de la imagen. Estos algoritmos logran capturar detalles pequeños en la imagen con números de superpíxeles grandes ( $K \geq 1000$ ). En la Figura 5.24 se muestra los resultados de la generación de superpíxeles con los algoritmos considerados. Se puede apreciar las diferentes representaciones de los superpíxeles como: contornos (Figura 5.24-(b)), valor medio de los superpíxeles (Figura 5.24-(c)), así como la comparación con su segmento *ground truth* correspondiente.

## 5.4 Resultados con el conjuntos de datos Brain Web



**Figura 5.24:** Resultados cualitativos de la evaluación, en el conjunto de datos de BSDS500, con  $k \approx 1000$ ; en la columna (a) se muestra la imagen original, (b) la representación con contornos, (c) representación con la intensidad media de los superpíxeles, (d) comparación con el segmento *ground truth* correspondiente para la métrica *Rec* y (e) comparación con el segmento *ground truth* correspondiente para la métrica *UE*

## 5.4 Resultados con el conjuntos de datos Brain Web

La Tabla 5.18 muestra la comparación final del rendimiento de los algoritmos considerados, en los conjuntos de datos BrainWeb y BSD500. Los mejores resultados se muestran en negritas. Los acrónimos “BST1” y “BST1+r” corresponde los estudios de resonancia magnética del cerebro, normal y con ruido, utilizados en las dos primeras evaluaciones.

**Tabla 5.18:** Comparación final de los algoritmos para extraer superpíxeles, con respecto *Rec*, *UE* y *EV* en los conjuntos de datos BrainWeb y BSDS500.

Conjunto de datos	Métricas	Algoritmos					
		<b>W</b>	<b>LSC</b>	<b>SLIC</b>	<b>IFCM</b>	<b>TP</b>	<b>PB</b>
BWT1	<i>Rec</i>	0.996	0.976	0.994	<b>0.998</b>	0.92	0.986
	<i>UE</i>	0.057	0.073	0.048	<b>0.047</b>	0.096	0.063
	<i>EV</i>	0.979	0.965	<b>0.988</b>	0.987	0.963	0.98
BWT1+r	<i>Rec</i>	0.994	0.958	0.993	<b>0.998</b>	0.902	0.986
	<i>UE</i>	0.069	0.096	<b>0.055</b>	0.059	0.097	0.075
	<i>EV</i>	0.972	0.951	0.984	<b>0.977</b>	0.947	0.975
BSDS500	<i>Rec</i>	0.865	0.883	0.863	<b>0.902</b>	0.813	0.846
	<i>UE</i>	0.093	0.093	0.091	0.109	<b>0.084</b>	0.098
	<i>EV</i>	0.85	0.886	0.874	<b>0.907</b>	0.864	0.83

El algoritmo **IFCM** tiene un buen rendimiento con respecto a las métricas de recuperación de límites. Logró conservar su rendimiento en la métrica *Rec* para los conjuntos de datos BWT1 y BWT1+r. Obtuvo el mejor rendimiento con la métrica *UE* en el conjunto de datos BWT1. Con respecto a la métrica *EV*, aunque no tuvo el mejor valor en el conjunto BWT1, pudo recuperarse con BWT1+r, obteniendo el mejor resultado. En el conjunto de datos BSDS500, obtuvo los mejores resultados en las métricas *Rec* y *EV*. **IFCM** genera superpíxeles irregulares con alta adherencia a los bordes, esta característica le permite preservar detalles pequeños en las imágenes, en los conjuntos de datos BWT1, BWT1+r y BSDS500.

**SLIC**, es uno de los algoritmo notables que generan superpíxeles regulares y compactos, en general tuvo un buen rendimiento en todas las pruebas, teniendo resultados similares al algoritmo **IFCM**. En la primera prueba, con el conjunto de datos BWT1, se desempeño bien, obteniendo el mejor valor en la métrica *EV*. En la segunda evaluación, con el conjunto de datos BWT1+r, tuvo el mejor resultado en la métrica *UE*.

**SLIC** trata de generar superpíxeles de igual tamaño y forma, esto provoca que se pierdan algunos detalles de la imagen, especialmente cuando las imágenes tienen detalles pequeños, con regiones con una alta variabilidad en su intensidad.

Los algoritmos **W**, **LSC** y **PB** tuvieron un buen desempeño en las evaluaciones con los conjuntos de datos BWT1, BWT1+r y BSDS500. El algoritmo **LSC**, generó superpíxeles regulares y compactos que se adaptan bien a los bordes de la imagen, obteniendo una buena aproximación de la imagen original con números de superpíxeles pequeños, sin embargo no consigue incrementar su nivel de adherencia con números de superpíxeles grandes.

**W**, generó superpíxeles irregulares y al igual que el algoritmo **IFCM**, logró preservar pequeños detalles de la imagen, con números de superpíxeles grandes. **PB**, produce superpíxeles aproximadamente del mismo tamaño, con poca adherencia a los bordes. **PB** perdió muchos detalles de las imágenes en las evaluaciones con los conjuntos de datos BWT1 y BWT1+r. El rendimiento de **PB** mejora conforme el número de superpíxeles aumenta.

El algoritmo **TP**, obtuvo un rendimiento bajo en los conjuntos de datos BWT1 y BWT1+r, en comparación al resto de los algoritmos considerados. En la evaluación con imágenes a color obtuvo un buen desempeño, obteniendo el mejor valor de adherencia a los bordes, con respecto a la métrica *UE*.

## 5.5. Discusión

La comparación cualitativa se basó en la recuperación de los límites a través de las métricas: *Rec* (Recuerdo de límites), *UE* (Error de subsegmentación) y *EV* (Variación Explicada). Estas métricas captan la estabilidad de los algoritmos para generar superpíxeles.

En general, los algoritmos **LSC**, **SLIC** y **PB** proporcionaron un buen rendimiento en los conjuntos de datos considerados; generaron superpíxeles de tamaño regular, que siguen los bordes de la imagen. **PB** tiene un buen desempeño cuando el número de superpíxeles es considerablemente grande ( $K \geq 700$ ). Los algoritmos basados en agrupamiento tienen comportamiento similar.

**IFCM** y **W** producen superpíxeles altamente irregulares, sin embargo son capaces de preservar pequeños detalles y los bordes importantes. **TP** tuvo el menor desempeño, aunque generó superpíxeles regulares y compactos, no logra preservar los pequeños detalles en las imágenes.

Los algoritmos que controlan la compacidad tratan de generar superpíxeles de tamaño y forma similar, esto afecta principalmente cuando el número de superpíxeles es pequeño, ya que las regiones iniciales presentan una alta variabilidad en la intensidad, causando que los superpíxeles resultantes pierdan los detalles en la imagen. En general, los algoritmos considerados, dependen en gran medida de los datos de la imagen y no de las restricciones de compacidad o de distancia. Los algoritmos considerados, reducen el espacio de búsqueda en regiones limitadas. Esta estrategia permite que los algoritmos tengan un costo computacional bajo.

# Capítulo 6

## Conclusiones

En este capítulo se realiza un análisis detallado de las conclusiones a las que se llegaron después de la evaluación del algoritmo y de estudiar los resultados obtenidos, con respecto a los objetivos y alcances planteados. También se hace un análisis para trabajos futuros y sugerencias sobre el tema.

### 6.1. Objetivos y alcances logrados

En la Tabla 6.1, se muestran las actividades realizadas en relación a los objetivos y alcances planteados para el desarrollo de la tesis.

**Tabla 6.1:** Objetivos y alcances realizados.

Objetivo	Actividad
Estudiar y comprender los algoritmos de agrupamiento difuso (intuitivo) y su utilización para segmentar imágenes.	Se realizó un estudio detallado de los conceptos básicos y específicos relacionados con el algoritmo de agrupamiento difuso y su extensión intuitiva. También se implementó el algoritmo <b>IFCM</b> convencional para la segmentación de imágenes.
Estudiar y comprender el algoritmo <b>SLIC</b> para la extracción de superpíxeles.	Se realizó un análisis del algoritmo <b>SLIC</b> tanto en su enfoque convencional así como otras mejoras al algoritmo encontradas en el estado del arte. También se realizó la implementación del algoritmo con el cual se realizaron varias pruebas.

## 6.1 Objetivos y alcances logrados

Tabla 6.1: Objetivos y alcances realizados (continuación).

Objetivo	Actividad
Adaptar el algoritmo agrupamiento difuso intuitivo para extraer superpíxeles utilizando imágenes en escala de grises.	El método propuesto es una adaptación del algoritmo <b>IFCM</b> para extraer superpíxeles. La implementación del enfoque propuesto se realizó en el Lenguaje C.
Evaluar el desempeño del algoritmo implementado utilizando bases de datos especializadas.	Para la evaluación del algoritmo propuesto con imágenes de resonancia magnética cerebral, se contempló la base de datos BrainWeb. Adicionalmente se utilizó la base de datos BSDS500 para la experimentación con imágenes a color. Ambos conjuntos de datos se describen detalladamente en la Sección 5.3.3.
Comparar el desempeño del algoritmo implementado con otras técnicas existentes en la literatura.	Se realizó una evaluación comparativa contemplando 5 algoritmos ( <b>W</b> , <b>SLIC</b> , <b>LSC</b> , <b>TP</b> , <b>PB</b> ) encontrados en el estado del arte.
Alcances	Actividad
Implementación de una metodología para la extracción de superpíxeles basado en <b>SLIC</b> utilizando el algoritmo de agrupamiento difuso intuitivo.	La implementación del enfoque propuesto se basa en una adaptación del algoritmo de agrupamiento difuso intuitivo para generar superpíxeles. Se tomó como base la inicialización del algoritmo <b>SLIC</b> .
Evaluar el algoritmo con las métricas definidas en la literatura.	Se utilizaron 3 métricas ( <i>Rec</i> , <i>UE</i> y <i>EV</i> ) encontradas en el estado del arte para la evaluación cuantitativa de los métodos superpíxel. Estas métricas miden el nivel de adherencia de los superpíxeles a los bordes de la imagen.
Evaluar el algoritmo con bases de datos especializadas en imágenes de resonancia magnética del cerebro.	Se utilizó la base de datos BrainWeb para simular los estudios de resonancia magnética del cerebro en modo T1. Se generaron dos estudios, el primer estudio se compone de 182 imágenes, libres de ruido. El segundo estudio se generó agregando 5% de ruido.

## 6.2. Resultados del trabajo

### 6.2.1. Productos

1. Como parte de los entregables, se redactó un informe del Estado del Arte donde se describen los trabajos relacionados con la sobresegmentación en imágenes de resonancia magnética a través de superpíxeles, así como el algoritmo de agrupamiento difuso intuitivo para el procesamiento de imágenes médicas. Estos trabajos se emplearon para dar sustento al presente trabajo de investigación.
2. Se redactó un informe sobre el algoritmo de agrupamiento difuso intuitivo, como parte de los productos entregables. El resumen de este informe fue incluido en la Sección 2.3 y 4.1 del presente documento.
3. Implementación del método propuesto se realizó en el lenguaje de programación C. Esta implementación tiene la ventaja de poder reutilizarse como un conjunto de funciones externas para otros estudios y aplicaciones.
4. Resultados de la evaluación comparativa del método propuesto con otros existentes en la literatura. Se elaboró el presente documento, como parte de los productos entregables descritos en la propuesta aceptada.
5. Participación como ponente de la conferencia “Extracción superpíxeles utilizando agrupamiento intuicionista local”, en el marco del evento: Escuela de inteligencia artificial y Robótica 2018, llevado a cabo en las instalaciones de la Universidad Tecnológica Emiliano Zapata, Cuernavaca, Morelos, México. Ver Anexo A.
6. Participación como asistente, en el Congreso Internacional de Ingeniería Mecánica, Electrónica y Automotriz, realizado del 27 al 30 de Noviembre del 2018, en la ciudad Cuernavaca, Morelos, México. Ver Anexo A.
7. Artículo: *2D Superpixel Oversegmentation of Images Evaluated by Different Color Spaces*, publicado en la revista JCYTA VOL. 2, CENIDET 2019, pp.30-34, ISSN en trámite. Ver Anexo A. La revista está disponible en el siguiente enlace: <https://www.cenidet.edu.mx/revistajornadacienciaytecnologiaaplicada.php>

### 6.2.2. Aportaciones

1. Desarrollo de una metodología basada en una adaptación del algoritmo de agrupamiento difuso intuitivo para la extracción de superpíxeles en imágenes de resonancia magnética del cerebro. El método propuesto se basó en el método de inicialización del algoritmo **SLIC**.
2. El método propuesto también puede usarse con imágenes a color, teniendo una buena adaptación de los superpíxeles a los bordes de la imagen.
3. El método propuesto tuvo un buen rendimiento en las experimentaciones realizadas en el Capítulo 5.3, con los conjuntos de datos BrainWeb y BSDS500. El método propuesto logró superar los 5 algoritmos considerados para la evaluación, en la mayoría de las pruebas, con respecto a las métricas *Rec*, *UE* y *EV*.

### 6.2.3. Conclusiones

En esta tesis se presenta un nuevo método para generar superpíxeles basado en una adaptación del algoritmo de Agrupamiento Difuso Intuitivo (**IFCM**). Las características del algoritmo **IFCM** permiten manejar con mayor eficiencia la incertidumbre, lo cual beneficia en la asignación de los píxeles a un determinado superpíxel, ya que la asignación no es binaria. El algoritmo se planteó inicialmente para imágenes en escala de grises, sin embargo, se implementó adicionalmente una extensión para imágenes a color.

El algoritmo propuesto tuvo un buen rendimiento en las pruebas con los conjuntos de datos de BrainWeb y BSDS500. Obtuvo los mejores resultados en la mayoría de las métricas de recuperación de límites. El algoritmo propuesto generó superpíxeles irregulares con alta adherencia a los bordes, esta característica le permite preservar detalles pequeños en las imágenes, en los conjuntos de datos considerados. Los resultados de la evaluación se detallan a continuación.

El algoritmo propuesto se comparó con 5 algoritmos existentes en la literatura. Se contemplaron dos algoritmos basados en agrupamiento (**SLIC** y **LSC**), un algoritmo basado en grafos (**PB**), un algoritmo basado en evolución del contorno (**TP**) y un algoritmo basado en cuencas hidrográficas (**W**). La evaluación se realizó utilizando 2 estudios simulados en BrainWeb en modo T1, el primer estudio se

genero libre de ruido y el segundo estudio se le agrego 5% de ruido.

Se contaron con 181 imágenes con una resolución de  $181 \times 127$  píxeles. Adicionalmente se utilizó el conjunto de datos BSDS500 para la evaluación con imágenes a color. Las métricas que se utilizaron cuantifican la adherencia a los bordes de la imagen. Los resultados de la evaluación con imágenes de resonancia magnética del cerebro libre de ruido destacan los siguientes puntos:

- El agrupamiento difuso intuitivo para extraer superpíxeles tiene un buen desempeño y logra adaptarse bien a los límites naturales de la imagen, siendo el algoritmo con mejor rendimiento en la métrica *Rec*.
- El algoritmo propuesto, no tiene control sobre la compacidad de los superpíxeles generados lo cual da como resultado superpíxeles poco irregulares, los cuales hacen que el algoritmo se vea diezmado en las métricas *UE* y *EV*.
- El algoritmo propuesto logra superar a **SLIC** en 2 métricas (*UE* y *Rec*).
- La calidad de los superpíxeles generados por los algoritmos considerados tiende a mejorar conforme el número de superpíxeles  $K$  se incrementa; en las pruebas se observa que a partir de  $K \approx 800$ , los algoritmos logran una buena aproximación de la imagen original.
- Considerando que el algoritmo propuesto sólo utiliza la intensidad de los píxeles como criterio de homogeneidad para agrupar los píxeles tiene un buen desempeño, en comparación con otros algoritmos como **SLIC**, que utiliza 3 características (la intensidad del píxel y la relación espacial) para imágenes en escala de grises.
- En cuanto al tiempo de ejecución el algoritmo es bastante rápido ya que en las pruebas con imágenes de  $181 \times 127$  tiene un tiempo de ejecución de entre 200 a 400 mili segundos.

Las pruebas con el estudio de resonancia magnética del cerebro con 5% de ruido se llegaron a las siguientes conclusiones:

- **IFCM** mantiene su rendimiento con respecto a la métrica *Rec*, a diferencia del resto de los algoritmos.
- En cuanto al tiempo de ejecución no se tiene un cambio significativo con respecto al las pruebas con RM sin ruido.

- Las dos pruebas anteriores muestran que la sobresegmentación tiene buenos resultados con las imágenes de RM al obtener una buena aproximación de la imagen original, logrado adaptarse bien a las regiones internas del cerebro, especialmente cuando  $K$  es considerablemente alto. Esto se debe a que los algoritmos consideran información local para agrupar los píxeles; además, algunos algoritmos como **SLIC**, **LSC** y **W** usan el mínimo local para descartar que un centro de agrupación sea un borde.

Finalmente, en las pruebas adicionales con imágenes de color, se observó que:

- El algoritmo propuesto mantiene un buen rendimiento teniendo resultados similares al algoritmo **SLIC** tanto para valores de  $K$  pequeños y grandes superando a los algoritmos **W**, **LSC**, **TP** y **PB** en las métricas *Rec* y *EV*.

### 6.2.4. Trabajo futuro

El trabajo futuro de esta investigación se enfocará en la incorporación de la información espacial como otra característica para la agrupación local. La incorporación de información espacial en el enfoque propuesto podría mejorar la regularidad de los superpíxeles generados, característica importante para los algoritmos superpíxel que contribuye en una buena adaptación a los bordes de la imagen. También se podría plantear la utilización de la información de textura como una característica, lo cual podría generar segmentos con mayor homogeneidad. En cuanto a la mejora del tiempo de ejecución, se podría trabajar en una versión paralelizada para aplicaciones en tiempo real.

# Bibliografía

- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2010). Slic superpixels. Technical report.
- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., et al. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282.
- Atanassov, K. T. (1986). Intuitionistic fuzzy sets. pages 87–96.
- Atanassov, K. T. (2017). *Intuitionistic fuzzy logics*. Springer.
- Avellaneda, A. D. (2009). *Caracterización de Texturas Naturales*. Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico.
- Bai, X., Cao, Z., Wang, Y., Ye, M., and Zhu, L. (2014). Image segmentation using modified slic and nyström based spectral clustering. *Optik-International Journal for Light and Electron Optics*, 125(16):4302–4307.
- Boemer, F., Ratner, E., and Lendasse, A. (2018). Parameter-free image segmentation with slic. *Neurocomputing*, 277:228–236.
- Bromiley, P. A., Thacker, N. A., and Courtney, P. (2001). Colour image segmentation by non-parametric density estimation in colour space. In *BMVC*, pages 1–10. Citeseer.
- Buysens, P., Gardin, I., Ruan, S., and Elmoataz, A. (2014). Eikonal-based region growing for efficient clustering. *Image and Vision Computing*, 32(12):1045–1054.
- Caponetti, L. and Castellano, G. (2017). *Fuzzy logic for image processing: a gentle introduction using Java*. Springer.
- Cervantes, A. J.-F. (2006). *Metodología para la Estructuración y uso de Conocimiento en Segmentación de Imágenes Digitales*. Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico.
- Chaira, T. (2010). Intuitionistic fuzzy segmentation of medical images. *IEEE Transactions on Biomedical Engineering*, 57(6):1430–1436.
- Chaira, T. (2015). *Medical image processing : advanced fuzzy set theoretic techniques*. CRC Press, 1 edition.

- Cheng, W., Ping, Y., Zhang, Y., Chuang, K.-H., and Liu, Y. (2013). Magnetic resonance imaging (mri) contrast agents for tumor diagnosis. *Journal of healthcare engineering*, 4(1):23–46.
- Cho, H., Kang, S.-J., Cho, S. I., and Kim, Y. H. (2014). Image segmentation using linked mean-shift vectors and its implementation on gpu. *IEEE Transactions on Consumer Electronics*, 60(4):719–727.
- Choi, K.-S. and Oh, K.-W. (2016). Subsampling-based acceleration of simple linear iterative clustering for superpixel segmentation. *Computer Vision and Image Understanding*, 146:1–8.
- Cocosco, C. A., Kollokian, V., Kwan, R. K.-S., Pike, G. B., and Evans, A. C. (1997). Brainweb: Online interface to a 3d mri simulated brain database. In *NeuroImage*. Citeseer.
- Cyganek, B. (2009). Framework for object tracking with support vector machines, structural tensor and the mean shift method. In *International Conference on Neural Information Processing*, pages 399–408. Springer.
- Davis, L. S. (1975). A survey of edge detection techniques. *Computer graphics and image processing*, 4(3):248–270.
- Despotović, I., Goossens, B., and Philips, W. (2015). Mri segmentation of the human brain: challenges, methods, and applications. *Computational and mathematical methods in medicine*, 2015.
- Despres, P. and Jia, X. (2017). A review of gpu-based medical image reconstruction. *Physica Medica*, 42:76–92.
- Diniz, P. H. B., Valente, T. L. A., Diniz, J. O. B., Silva, A. C., Gattass, M., Ventura, N., Muniz, B. C., and Gasparetto, E. L. (2018). Detection of white matter lesion regions in mri using slic0 and convolutional neural network. *Computer methods and programs in biomedicine*, 167:49–63.
- Dong, L. and Zhou, J. (2016). Noise-robust slic superpixel for natural images. In *Cloud Computing and Big Data (CCBD), 2016 7th International Conference on*, pages 335–340. IEEE.
- Dougherty, G. (2009). *Digital image processing for medical applications*. Cambridge University Press.
- Dubey, Y. K., Mushrif, M. M., and Mitra, K. (2016). Segmentation of brain mr images using rough set based intuitionistic fuzzy clustering. *Biocybernetics and Biomedical Engineering*, 36(2):413–426.
- El Aziz, M. A., Ewees, A. A., and Hassanien, A. E. (2016). Hybrid swarms optimization based image segmentation. In *Hybrid Soft Computing for Image Segmentation*, pages 1–21. Springer.

- Fabijańska, A. and Gołąbski, J. (2016). An application of the supervoxel-based fuzzy c-means with a gpu support to segmentation of volumetric brain images. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 777–785. IEEE.
- Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181.
- Fisher, R. B., Breckon, T. P., Dawson-Howe, K., Fitzgibbon, A., Robertson, C., Trucco, E., and Williams, C. K. (2013). *Dictionary of computer vision and image processing*. John Wiley & Sons.
- Fu, H., Cao, X., Tang, D., Han, Y., and Xu, D. (2014). Regularity preserved superpixels and supervoxels. *IEEE Transactions on Multimedia*, 16(4):1165–1175.
- Funke, J., Zhang, C., Pietzsch, T., and Saalfeld, S. (2017). The candidate multi-cut for cell segmentation. *arXiv preprint arXiv:1707.00907*.
- Ghadiri, F., Bergevin, R., and Bilodeau, G.-A. (2018). From superpixel to human shape modelling for carried object detection. *arXiv preprint arXiv:1801.03551*.
- Ghanem, T. F., Elkilani, W. S., Abdelkader, H. M., and Hadhoud, M. M. (2015). Fast dimension-based partitioning and merging clustering algorithm. *Applied Soft Computing*, 36:143–151.
- Han, C.-y. (2017). Improved slic image segmentation algorithm based on k-means. *Cluster Computing*, 20(2):1017–1023.
- He, J., Kim, C.-S., and Kuo, C.-C. J. (2013). *Interactive Segmentation Techniques: Algorithms and Performance Evaluation*. Springer Science & Business Media.
- Hu, Z., Zou, Q., and Li, Q. (2015). Watershed superpixel. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 349–353. IEEE.
- Humayun, A., Li, F., and Rehg, J. M. (2015). The middle child problem: Revisiting parametric min-cut and seeds for object proposals. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1600–1608.
- Irving, B. (2016). maskslic: regional superpixel generation with application to local pathology characterisation in medical images. *arXiv preprint arXiv:1606.09518*.
- Kacprzyk, J. (2013). *Studies in Fuzziness and Soft Computing*, volume 295. Springer.
- Krig, S. (2014). *Computer vision metrics: Survey, taxonomy, and analysis*. Apress.
- Laal, M. (2013). Innovation process in medical imaging. *Procedia - Social and Behavioral Sciences*, 81:60–64.
- Langone, R. and Suykens, J. A. (2017). Fast kernel spectral clustering. *Neurocomputing*, 268:27–33.

- Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., and Siddiqi, K. (2009). Turbopixels: Fast superpixels using geometric flows. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2290–2297.
- Li, D.-F. (2014). *Decision and game theory in management with intuitionistic fuzzy sets*, volume 308. Springer.
- Li, Z. and Chen, J. (2015). Superpixel segmentation using linear spectral clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1356–1363.
- Liu, Z., Zou, W., and Le Meur, O. (2014). Saliency tree: A novel saliency detection framework. *IEEE Transactions on Image Processing*, 23(5):1937–1952.
- Machairas, V., Decenciere, E., and Walter, T. (2014). Waterpixels: Superpixels based on the watershed transformation. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 4343–4347. IEEE.
- Machairas, V., Faessel, M., Cárdenas-Peña, D., Chabardes, T., Walter, T., and Decencière, E. (2015). Waterpixels. *IEEE Transactions on Image Processing*, 24(11):3707–3716.
- Maghsoudi, O. H. and Spence, A. (2017). 3d based landmark tracker using superpixels based segmentation for neuroscience and biomechanics studies. *arXiv preprint arXiv:1711.08785*.
- Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423.
- Martin, D. R., Fowlkes, C. C., and Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5):530–549.
- Michielsen, K. and De Raedt, H. (2000). Morphological image analysis. *Computer Physics Communications*, 132(1-2):94–103.
- Moore, A. P., Prince, S. J., Warrell, J., Mohammed, U., and Jones, G. (2008). Superpixel lattices. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.
- Morales, X. M. (2018). *Segmentación no Paramétrica de Tejidos Cerebrales Mediante una Arquitectura Paralela de Redes Neuronales Convolucionales*. Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico.
- Mújica, V. D. (2013). *Segmentación de Imágenes Utilizando Algoritmos Robustos de Agrupamiento Difuso*. Tesis doctoral, Instituto Politécnico Nacional.
- Mújica-Vargas, D., Gallegos-Funes, F. J., and Rosales-Silva, A. J. (2013). A fuzzy clustering algorithm with spatial robust estimation constraint for noisy color image segmentation. *Pattern Recognition Letters*, 34(4):400–413.

- Neubert, P. and Protzel, P. (2014). Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 996–1001. IEEE.
- Pătrașcu, V. (2007). Fuzzy image segmentation based on triangular function and its n-dimensional extension. In *Soft Computing in Image Processing*, pages 187–207. Springer.
- Preston, D. (2006). Magnetic resonance imaging (mri) of the brain and spine: Basics. *School of medicine, Case Western Reserve University*.
- Ren, C. Y., Prisacariu, V. A., and Reid, I. D. (2015). gslicr: Slic superpixels at over 250hz. *arXiv preprint arXiv:1509.04232*.
- Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *null*, page 10. IEEE.
- Schick, A., Fischer, M., and Stiefelhagen, R. (2014). An evaluation of the compactness of superpixels. *Pattern Recognition Letters*, 43:71–80.
- Shi, C. and Pun, C.-M. (2018). Superpixel-based 3d deep neural networks for hyperspectral image classification. *Pattern Recognition*, 74:600–616.
- Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107.
- Strassburg, J., Grzeszick, R., Rothacker, L., and Fink, G. A. (2015). On the influence of superpixel methods for image parsing. In *VISAPP (2)*, pages 518–527.
- Stutz, D., Hermans, A., and Leibe, B. (2017). Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*.
- Suárez, S. D.-G. (2018). *Sintonización de una Red Totalmente Conectada para Segmentación de Dos Clases de Objetos en Imágenes*. Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico.
- Szmidt, E. (2014). *Distances and similarities in intuitionistic fuzzy sets*, volume 307. Springer.
- Tamajka, M. and Benesova, W. (2016). Automatic brain segmentation method based on supervoxels. In *Systems, Signals and Image Processing (IWSSIP), 2016 International Conference on*, pages 1–4. IEEE.
- Tasli, H. E., Cigla, C., and Alatan, A. A. (2015). Convexity constrained efficient superpixel and supervoxel extraction. *Signal Processing: Image Communication*, 33:71–85.
- Tian, Z., Liu, L., Zhang, Z., and Fei, B. (2016). Superpixel-based segmentation for 3d prostate mr images. *IEEE transactions on medical imaging*, 35(3):791–801.
- Tighe, J. and Lazebnik, S. (2010). Superparsing: scalable nonparametric image parsing with superpixels. In *European conference on computer vision*, pages 352–365. Springer.

- Verma, H., Agrawal, R., and Sharan, A. (2016). An improved intuitionistic fuzzy c-means clustering algorithm incorporating local information for brain image segmentation. *Applied Soft Computing*, 46:543–557.
- Verma, N., Cowperthwaite, M. C., and Markey, M. K. (2013). Superpixels in brain mr image analysis. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 1077–1080. IEEE.
- Wang, M., Liu, X., Gao, Y., Ma, X., and Soomro, N. Q. (2017a). Superpixel segmentation: a benchmark. *Signal Processing: Image Communication*, 56:28–39.
- Wang, X.-Y., Wu, C.-W., Xiang, K., and Chen, W. (2017b). Efficient local and global contour detection based on superpixels. *Journal of Visual Communication and Image Representation*, 48:77–87.
- Xiang, S., Pan, C., Nie, F., and Zhang, C. (2010). Turbopixel segmentation using eigen-images. *IEEE Transactions on Image Processing*, 19(11):3024–3034.
- Xu, Z. (2013). *Intuitionistic fuzzy aggregation and clustering*, volume 279. Springer.
- Xu, Z. and Cai, X. (2012). Intuitionistic fuzzy information aggregation. In *Intuitionistic Fuzzy Information Aggregation*, pages 1–102. Springer.
- Yang, Y., Wang, Y., and Xue, X. (2016). A novel spectral clustering method with superpixels for image segmentation. *Optik-International Journal for Light and Electron Optics*, 127(1):161–167.
- Yao, J., Boben, M., Fidler, S., and Urtasun, R. (2015). Real-time coarse-to-fine topologically preserving segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2947–2955.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3):338–353.
- Zhang, Y., Hartley, R., Mashford, J., and Burn, S. (2011). Superpixels via pseudo-boolean optimization. In *2011 International Conference on Computer Vision*, pages 1387–1394. IEEE.

# Anexo A

## Documentos correspondientes a participaciones



Figura A.1: Escuela de Inteligencia Artificial y Robótica 2018.

**ICMEAE**  
INTERNATIONAL CONFERENCE ON MECHATRONICS, ELECTRONICS AND AUTOMOTIVE ENGINEERING

El Centro de Investigación en Ingeniería y Ciencias Aplicadas de la Universidad Autónoma del Estado de Morelos y El Instituto de Ingenieros Electrónicos y Eléctricos de Morelos A.C.

otorgan el presente

# RECONOCIMIENTO

**A: David Domínguez López**

Por su destacada participación como:

## Asistente

En el marco del Congreso Internacional de Ingeniería Mecatrónica, Electrónica y Automotriz realizado del 27 al 30 de Noviembre del 2018 en la ciudad de Cuernavaca, Morelos, México.

**Ing. Leoncio Aguilar Negrete**  
Presidente del Instituto de Ingenieros Electrónicos y Eléctricos de Morelos A.C.

**Dra. Elsa Carmina Menchaca Campos**  
Directora Interina del Centro de Investigación en Ingeniería y Ciencias Aplicadas

Logos: CIICAp, IEEE, IIEEM, IEEE Robotics & Automation Society, UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS, IEEE Computational Intelligence Society.

Figura A.2: ICMEAE 2018.

## 2-D Superpixel Oversegmentation of Images Evaluated by Different Color Spaces

David Domínguez López \* Dante Mújica-Vargas \*\*  
Diego C. Luna-Márquez \*\*\*

\* CENIDET-TecNM, Cuernavaca, México (e-mail:  
david.dominguez17ce@cenidet.edu.mx.)

\*\* CENIDET-TecNM, Cuernavaca, México (e-mail:  
dantemv@cenidet.edu.mx.)

\*\*\* CENIDET-TecNM, Cuernavaca, México (e-mail:  
diego.luna18ca@cenidet.edu.mx.)

**Abstract:** The image oversegmentation by using superpixels is an efficient way to simplify it through pixel regions perceptually similar; in other words, it reduces their complexity in order to develop a posterior global segmentation. This feature has allowed superpixels to be used in areas such as image processing and computer vision. SLIC (Simple Linear Iterative Clustering) is one of the most efficient superpixel algorithms in the literature, it has a low runtime and produces superpixels with dynamical adjust to objects in size and shape, in addition to its low computational cost. In this paper an evaluation of the SLIC algorithm considering the CIE  $L^*a^*b^*$  (original one used), HSV, YCbCr and the orthonormal IJK color spaces. The performance of the algorithm and the quality of the superpixel segmentation are evaluated using the metrics proposed in the literature.

**Keywords:** Oversegmentation, superpixel, SLIC, color spaces, image processing.

### 1. INTRODUCTION

The oversegmentation by means of super pixels is considered a subproblem of image segmentation in order to provide a simpler, more compact and efficient representation of an image without losing the detail between pixels. This allows to represent an image with only a couple of hundred segments instead of tens of thousands of pixels Boemer et al. (2018), Schick et al. (2014). Superpixels are homogeneous regions of pixels grouped by low-level characteristics. They were introduced by Ren and Malik (2003), as more natural and coherent entities of an image while at the same time reducing the number of primitives.

The image redundancy is downscale when the graphical nodes are built with superpixels, this causes that both complexity and calculation time decrease substantially. The reduction of complexity allows superpixels to be very useful in various applications of computer vision and image processing Tasli et al. (2015). For example for biomedical image segmentation Ensafi et al. (2016), medical image analysis Verma et al. (2013), to mention just a few.

In recent years, several superpixel algorithms have been introduced in the state-of-art. In relation to the approach used to generate superpixels, the following categorization can be done:

**Based on watershed.** Some super-pixel algorithms based on this approach control compactness, for example *Compact Watershed* controls compactness with a single parameter by using a distance metric that combines the distance based on the conventional aspect and the Euclidean distance from the pixel to the seed of the segment Neubert and Protzel (2014).

**Based on clustering.** These algorithms are based on clustering algorithms to generate superpixels. SLIC Achanta et al. (2010) is inspired by the *k-means* algorithm for grouping pixels. The number of superpixels and the compactness are controllable through the parameters  $k$  and  $m$ , respectively. To group pixels, it uses a distance metric that combines the spatial and color information of each pixel. This algorithm requires further processing to reinforce the connectivity between the superpixels.

**Based on the energy optimization.** These algorithms iteratively optimize a formulated energy. For example in Yao et al. (2015), the image is divided into a regular grid such as initial segmentation of superpixels, and the pixels are exchanged between neighboring superpixels with respect to energy. The number of superpixels and the compactness can be controlled. The algorithm employs a single iteration to converge on a better minimum of energy, this allows it to be fast.



**EL TECNOLÓGICO NACIONAL DE MÉXICO  
A TRAVÉS DEL CENTRO NACIONAL DE INVESTIGACIÓN  
Y DESARROLLO TECNOLÓGICO**

OTORGAN LA PRESENTE

**CONSTANCIA**

A

**David Domínguez López, Dante Mújica-Vargas, Diego C.  
Luna Márquez**

POR SU PARTICIPACIÓN CON EL ARTÍCULO: **“2-D SUPERPIXEL  
OVERSEGMENTATION OF IMAGES EVALUATED BY DIFFERENT  
COLOR SPACES”**, PRESENTADO EN LA SEGUNDA JORNADA DE  
CIENCIA Y TECNOLOGÍA APLICADA, QUE SE LLEVO A CABO LOS  
DÍAS 4 Y 5 DE ABRIL DE 2019, EN LA CIUDAD DE CUERNAVACA,  
MORELOS, MÉXICO.

CUERNAVACA, MOR., A 5 DE ABRIL DE 2019.



S. E. P.

CENTRO NACIONAL DE INVESTIGACION  
Y DESARROLLO TECNOLÓGICO

**DR. VÍCTOR HUGO OLIVARES PEREGRINO**  
DIRECTOR



**Figura A.4:** Constancia del artículo publicado en la revista JCYTA VOL. 2, CENIDET 2019.