

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE CIENCIAS MATEMÁTICAS
Departamento de Estadística e Investigación Operativa



TESIS DOCTORAL

**Problemas de segmentación y detención de bordes en
imágenes digitales con información borrosa**

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Carely Guada Escalona

Directores

Javier Montero de Juan
Daniel Gómez González
Juan Tinguaro Rodríguez González

Madrid
Ed. electrónica 2019

Problemas de segmentación y detección
de bordes en imágenes digitales
con información borrosa



MEMORIA PRESENTADA PARA OPTAR AL
GRADO DE DOCTOR POR

Carely Guada Escalona

Departamento de Estadística e Investigación Operativa
Facultad de Ciencias Matemáticas
Universidad Complutense de Madrid

Junio 2018

Problemas de segmentación y
detección
de bordes en imágenes digitales
con información borrosa

Memoria presentada para optar al grado de
Doctor en
Ingeniería Matemática, Estadística e Investigación Operativa

Dirigida por
Javier Montero de Juan
Daniel Gómez González
Juan Tinguaro Rodríguez González

Departamento de Estadística e Investigación Operativa
Facultad de Ciencias Matemáticas
Universidad Complutense de Madrid

Junio 2018

A mis padres

*La ciencia se compone
de errores, que a su vez son los
pasos hacia la verdad.
Jules Verne*

Índice

Agradecimientos	XIX
Resumen	XXI
Abstract	XXIII
1. Introducción	1
1.1. Objetivo general	4
1.2. Objetivos específicos	4
1.3. Metodología	4
1.4. Estructura de la memoria	7
2. Preliminares	9
2.1. Imagen digital	9
2.2. Teoría de grafos	10
2.2.1. Conceptos básicos y notación de grafos en el contexto de imágenes digitales	11
2.2.2. Subgrafo, árbol soporte y bosque soporte	14
2.3. Segmentación de imágenes	15
2.4. Segmentación jerárquica en redes	18
2.4.1. Algoritmo de segmentación jerárquica en redes	21
2.5. Detección de bordes	23
2.5.1. Operador sobel	26
2.5.2. Operador canny	26
2.6. Teoría de conjuntos borrosos	28
3. Segmentación borrosa basada en teoría de grafos	31
3.1. Nuevo concepto de segmentación grafo teórico	32
3.2. Concepto de segmentación jerárquica de imágenes	36
3.3. Concepto de segmentación borrosa de imágenes	38
3.3.1. Algoritmo para construir una segmentación borrosa basado en una segmentación jerárquica	44

3.4.	Métodologías de visualización	46
3.4.1.	Metodología de visualización de una segmentación nítida	46
3.4.2.	Metodología de visualización de una segmentación jerárquica	48
3.4.3.	Metodología de visualización de una segmentación borrosa	50
3.5.	Relaciones entre algunos problemas de tratamiento de imágenes	51
3.5.1.	Problemas de tratamiento de imágenes	51
3.5.2.	Relación entre problemas	54
3.5.3.	Nueva clasificación de los principales problemas	57
4.	Nuevo algoritmo de detección de bordes, <i>Image-Divide and Link (ID&L)</i>	59
4.1.	Algoritmo Image-Divide-and-Link (ID&L)	59
4.1.1.	Modelización de una imagen como una red	60
4.1.2.	Umbrales en el proceso iterativo	60
4.1.3.	Determinación de los mapas de regiones jerárquicos	62
4.1.4.	Redes gruesas en el algoritmo de partición jerárquica	63
4.1.5.	Transformación de un mapa de regiones a un mapa de bordes	67
4.1.6.	Salida del algoritmo Image Divide and Link	68
5.	Evaluación del ID&L	69
5.1.	Metodología de evaluación a través de bordes	69
5.2.	Nueva metodología de evaluación de bordes	72
6.	Resultados computacionales	77
6.1.	Resultados computacionales sobre la evaluación del algoritmo ID&L	77
6.1.1.	Configuración experimental	78
6.1.2.	Resultados en el conjunto de entrenamiento	80
6.1.3.	Resultados en el conjunto de pruebas	81
6.2.	Competición de Detección de Bordes en el congreso EUSFLAT 2017	83
6.2.1.	Resultados del ID&L	86
7.	Conclusiones y trabajo futuro	89
7.1.	Conclusiones	89
7.2.	Contribuciones relacionadas con este trabajo	91
7.2.1.	Artículos en revistas de impacto	91
7.2.2.	Artículos en otras revistas	91
7.2.3.	Contribuciones a congresos	92

7.3. Futuras líneas de investigación	94
Bibliografía	97

Índice de figuras

2.1. Ejemplo de la representación de una imagen	10
2.2. Grafo con cuatro nodos $\{a, b, c, d\}$ y cuatro aristas $\{e_{ab}, e_{ac}, e_{bc}, e_{cd}\}$	11
2.3. Algunas topologías de vecinos sobre un grafo	12
2.4. Ejemplo de la modelización de una imagen como una red de dimensiones 4×5	13
2.5. Segmentación mediante los nodos de una red $N(I)$	17
2.6. Salida del Normalized Cuts sobre la imagen 2.1a	19
2.7. Red $N(I)$	20
2.8. Segmentación jerárquica de la red $N(I)$	21
2.9. Salida del detector de bordes - Sobel	27
2.10. Salida del detector de bordes - Canny	28
3.1. (a) Red $N(I)$ y (b) segmentación mediante las aristas de una red $N(I)$	32
3.2. De la red de la Figura 3.1a: (a) bosque soporte F y el subconjunto de aristas $F' \subset F$; y (b) el subconjunto B derivado por esos subconjuntos	35
3.3. Segmentación jerárquica por nodos	37
3.4. Segmentación jerárquica por aristas	38
3.5. Imágenes de la base de datos de Berkeley (BSDS500) (Martin et al., 2001)	39
3.6. Segmentación realizada a una imagen de la base de datos de Berkeley (BSDS500) (Martin et al., 2001)	40
3.7. Ejemplo del orden en cómo ve un humano los objetos de una imagen	41
3.8. (a) Red $N(I)$ y (b) segmentación borrosa \tilde{B} del ejemplo 2.2.1	42
3.9. Segmentaciones nítidas $B(0.3)$ y $B(0.6)$ de la segmentación borrosa \tilde{B} del Ejemplo 2.2.1	42
3.10. (a) Red, (b) segmentación jerárquica por aristas y (c) segmentación borrosa del Ejemplo 2.2.1	47
3.11. Visualización de una segmentación por medio de un mapa de regiones	47

3.12. Visualización de una segmentación por medio de un mapa de bordes	48
3.13. Segmentación jerárquica de una imagen $N(I)$	49
3.14. Ejemplo de visualización de una segmentación borrosa	50
3.15. Ejemplo de transformación de una imagen segmentada a la detección de bordes	54
3.16. Ejemplo de transformación de la detección de bordes a una imagen segmentada	55
3.17. Segmentación jerárquica	56
3.18. Segmentación borrosa	56
3.19. Salidas de algunas técnicas de tratamiento de imágenes sobre la imagen de la iglesia 2.1a.	58
4.1. Contracción de nodos $d_{ab} \leq \alpha_\tau$	64
4.2. (a) Imagen en escala de grises, (b) la matriz asociada y (c) la red N	65
4.3. Iteraciones del proceso de partición jerárquica de regiones	65
4.4. (a) Red original y (b) red contraída	66
4.5. Contracción de nodos sobre la partición jerárquica de regiones; las aristas de cortes de color rojo	66
4.6. Detección de bordes obtenida con el algoritmo ID&L en la imagen del <i>oso</i> . (a) Imagen original. (b)-(h) Mapa de bordes automáticos resultantes del vector de umbrales $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8) = (47.79, 40.55, 33.29, 28.47, 23.63, 18.79, 11.55, 4.30)$ y con $L = 3$	68
5.1. Mapa de bordes sobre la imagen de la <i>iglesia</i>	70
5.2. Comparación de la salida del detector Canny con cada referencia humana sobre la imagen de la <i>iglesia</i> , sin considerar un nivel de tolerancia para el cálculo de los verdaderos positivos (píxeles verdes), los verdaderos negativos (píxeles negros), los falsos positivos (píxeles amarillos) y los falsos negativos (píxeles morados)	75
6.1. (I)-(VIII) Las distintas distribuciones consideradas para los valores α	80
6.2. Curvas P&R de todos los algoritmos y los valores mayores de P&R (íconos resaltados) obtenidos usando $\mathcal{A} = \max$ y $\sigma_s = 0$. Los mejores parámetros resultantes correspondientes a los íconos resaltados son: Sobel ($t = 0.30$), Canny2($\sigma_c = 2, t_1 = 0.156, t_2 = 0.39$), Canny4($\sigma_c = 4, t_1 = 0.124, t_2 = 0.31$), ID&L ($K = 6, \alpha = VII$ distribución, $\alpha_\tau = 0.09, L = 3$)	81

6.3.	Curvas de P&R de todos los algoritmos y los valores mayores de P&R (íconos resaltados) obtenidos usando $\mathcal{A} = \textit{media}$ y $\sigma_s = 0$. Los mejores parámetros resultantes correspondientes a los íconos resaltados son: Sobel ($t = 0.31$), Canny2($\sigma_c = 2, t_1 = 0.156, t_2 = 0.39$), Canny4($\sigma_c = 4, t_1 = 0.124, t_2 = 0.31$), ID&L ($K = 6, \alpha = VII$ distribución, $\alpha_\tau = 0.09, L = 3$)	82
6.4.	Curvas de P&R de todos los algoritmos y los valores mayores de P&R (íconos resaltados) obtenidos usando $\mathcal{A} = \textit{min}$ y $\sigma_s = 0$. Los mejores parámetros resultantes correspondientes a los íconos resaltados son: Sobel ($t = 0.31$), Canny2($\sigma_c = 2, t_1 = 0.156, t_2 = 0.39$), Canny4($\sigma_c = 4, t_1 = 0.112, t_2 = 0.28$), ID&L ($K = 6, \alpha = VII$ distribución, $\alpha_\tau = 0.09, L = 3$)	82
6.5.	(a) Imagen original y (b) <i>ground truth</i>	85
6.6.	Imagen resultante al ajustar los valores de intensidad de la imagen de ejemplo 6.5a antes de aplicar el ID&L	87
6.7.	Salida del ID&L	88

Índice de Tablas

3.1. Nueva clasificación de algunos problemas de tratamiento de imágenes	57
6.1. Descomposición esquemática de los algoritmos de detección de bordes utilizados en la configuración experimental, de acuerdo con la estructura de desglose de Bezdek (Bezdek et al., 1998)	78
6.2. Descomposición esquemática del algoritmo ID&L utilizado en la configuración experimental (Gómez et al., 2015a)	79
6.3. Resultados globales para cada uno de los algoritmos sobre el conjunto de datos de entrenamiento de la BSDS500	83
6.4. Resultados globales para cada uno de los métodos sobre el conjunto de datos de prueba de la BSDS500	84
6.5. Prueba de Wilcoxon de las comparaciones del algoritmo ID&L con todos los algoritmos de detectores de bordes considerados para $\mathcal{A} = \text{max}$ con $\sigma_s = 0$	84
6.6. Prueba de Wilcoxon de las comparaciones del algoritmo ID&L con todos los algoritmos de detectores de bordes considerados para $\mathcal{A} = \text{media}$ con $\sigma_s = 0$	85
6.7. Prueba de Wilcoxon de las comparaciones del algoritmo ID&L con todos los algoritmos de detectores de bordes considerados para $\mathcal{A} = \text{min}$ con $\sigma_s = 0$	85
6.8. Resultados de la competición de Edge Detection, EUSFLAT 2017	87

Agradecimientos

El científico encuentra su recompensa en lo que Henri Poincare llama el placer de la comprensión, y no en las posibilidades de aplicación que cualquier descubrimiento pueda conllevar

Albert Einstein

Esta memoria no hubiese sido posible sin la guía y el apoyo de mis supervisores Javier Montero, Daniel Gómez y Tinguaro Rodríguez, y del profesor Javier Yáñez. Muchas gracias por la dedicación y el tiempo de ser mis mentores.

También quiero expresar mi gratitud a los miembros del grupo GIARA de Pamplona y del IRAFM de Ostrava, quienes me orientaron en mis estancias de investigación.

Igualmente, quiero agradecer a mis compañeros del doctorado por acompañarme en el día a día.

Además, agradezco a todos mis amigos, en particular a los de Madrid y Mérida, que me han apoyado en lo posible durante estos años de investigación.

También doy las gracias a Luis y sobre todo a mi familia, en especial a mi hermano Victor, y por supuesto a mis padres Elydes y Carlos por su apoyo incondicional.

Carely

Resumen

*Uno nunca ve lo que se ha hecho, sino
que ve lo que queda por hacer.*

Marie Curie

En esta memoria se presenta en primer lugar, la definición de segmentación borrosa de imágenes basado en teoría de grafos. Para ello, también se define formalmente la segmentación nítida y jerárquica de imágenes. Asimismo, se plantea un algoritmo para construir una segmentación borrosa partiendo de una segmentación jerárquica. Y, se propone una metodología de visualización para la segmentación nítida, jerárquica y borrosa de imágenes. En segundo lugar, se establece una nueva clasificación sobre los problemas de tratamiento de imágenes. En tercer lugar, se diseña un nuevo algoritmo de detección de bordes basado en un agrupamiento jerárquico de grafos, es decir, el nuevo algoritmo genera una salida que corresponde a una secuencia de mapa de regiones jerárquicos representados a través del contorno dichas regiones como un mapa de bordes binario. Además, se formaliza una manera de evaluar las salidas de este nuevo algoritmo con el fin de compararlo con otros algoritmos relevantes y muy utilizados en el campo del tratamiento de imágenes.

Palabras claves: grafos, segmentación borrosa, detección de bordes, evaluación, conjuntos borrosos.

Abstract

Success is a science; if you have the conditions, you get the result.

Oscar Wilde

This work has the purpose, first to define the fuzzy image segmentation concept based on graph theory. To this aim, the concept of crisp segmentation and hierarchical segmentation are formally defined. Furthermore, an algorithm to construct a fuzzy segmentation through hierarchical segmentation was presented. And a visualization methodology for crisp segmentation, hierarchical segmentation and fuzzy segmentation were proposed. Secondly, a new classification of problems of image processing was presented. Thirdly, a new edge detection algorithm is also designed based on hierarchical clustering algorithm. This new algorithm generates an output corresponding to a sequence of hierarchical regions represented through the contour of these regions as a boundary map. In addition, a way to evaluate the outputs of this edge detection algorithm is proposed. This algorithm is evaluated it with other algorithms that are relevant and widely used in the field of image processing.

Keywords: graphs, fuzzy segmentation, edge detection, benchmarking, fuzzy sets.

Capítulo 1

Introducción

Investigación es lo que hago cuando no sé lo que estoy haciendo

Wernher von Braun

El tratamiento de imágenes es un campo de la inteligencia computacional, que consiste en imitar la cognición humana con el propósito de percibir y entender electrónicamente una imagen digital (Sonka et al., 1993). De este modo, al tratar de percibir y entender una imagen replicando la visión humana, el tratamiento de imágenes facilita o perfecciona la realización de diversas tareas, principalmente en grandes procesos, y es por ello que, actualmente, su uso en numerosas aplicaciones reales está muy extendido. Ejemplos de estas aplicaciones pueden ser, entre otras, la inspección por máquinas, el reconocimiento de imágenes médicas, la detección de rostro en imágenes fotográficas (ver por ejemplo Tellaache et al., 2011; Chaudhry y Chandra, 2017; Campomanes-Alvarez et al., 2016; Napalkova et al., 2014). Las tareas de tratamiento de imágenes se pueden dividir en dos grupos: aquellas que realizan tratamiento de imagen de bajo nivel y las que hacen entendimiento de imagen de alto nivel (Sonka et al., 1993). Ambas se diferencian de acuerdo al tipo de dato que utilizan, es decir, las tareas de tratamiento de imágenes de bajo nivel representan a las imágenes originales por matrices compuestas por la intensidad de brillo, por lo que usan muy poco conocimiento sobre el contenido de la imagen para comprenderla, y aplican métodos de tratamiento para el filtrado de ruido, extracción de bordes y objetos de la imagen. Por otro lado, las tareas de alto nivel intentan imitar la cognición y habilidad humana al extraer nueva información de las imágenes, comenzando con un modelo formal que se compara con la realidad percibida. En caso de encontrarse diferencias entre dichos modelos, se aplican tareas de tratamiento de imagen de bajo nivel para buscar información; esto se repite iterativamente hasta que el computador convierta la imagen en una información útil para el usuario (Sonka et al., 1993).

En general, el tratamiento de imágenes tiene el objetivo principal de resolver tareas complejas, para lo cual utiliza métodos o técnicas tales como captura de imagen, digitalización, preprocesamiento, detección de bordes, segmentación de regiones, clasificación de regiones, identificación de alto nivel, conclusiones cuantitativas/cualitativas, entre otros (Sonka et al., 1993). De esta forma, dependiendo de la tarea a desarrollar y de la salida deseada, es necesario aplicar una determinada técnica. En otras palabras, existe una variedad de técnicas de tratamiento de imágenes dirigidas a resolver determinados tipos de tareas, y que son ejecutadas mediante diferentes técnicas y algoritmos de tratamiento de información y aprendizaje automático. Asimismo, a veces es necesario la combinación de diversas técnicas para alcanzar el objetivo deseado.

En la literatura especializada, muchas de las tareas de tratamiento de imágenes de bajo nivel están claramente distinguidas de acuerdo a la salida que generan, aunque algunas de ellas, como por ejemplo técnicas que realizan segmentación de imágenes y detección de bordes, son frecuentemente confundidas como técnicas que abordan la misma tarea. Esta confusión puede deberse a la relación existente entre ellas, esto es, la segmentación de imágenes divide una imagen en múltiples regiones homogéneas no solapadas entre sí (Sonka et al., 1993); en cambio, la detección de bordes, como su nombre indica, detecta bordes en una imagen analizando la función de intensidad de cada píxel (Sonka et al., 1993; Canny, 1986), mas no necesariamente encierra regiones al identificar los bordes encontrados en la imagen. Sin embargo, la relación entre dichas tareas, subyace en que luego de realizar una segmentación de imágenes, se puede obtener una detección de bordes delimitando las fronteras de cada una de las regiones segmentadas. El caso inverso, sin embargo, no es siempre cierto.

Así, esta confusión respecto al objetivo principal que abordan algunas tareas muy conocidas de tratamiento de imágenes motivó una de las orientaciones iniciales del trabajo recogido en esta memoria, a saber, el desarrollo de una nueva clasificación de las principales tareas de tratamiento de imágenes, con el fin de distinguir las diferencias conceptuales entre ellas atendiendo en particular al tipo de salida que estas generan.

A medida que se realizaba esta clasificación sobre las principales tareas de tratamiento de imágenes, se observó que pocas investigaciones habían abordado la extensión del concepto de segmentación nítida a un contexto borroso (para más información, ver Bloch, 1993; Rosenfeld, 1979; Saha et al., 2000; Udupa y Samarasekera, 1996). Por ende, esto motivó a proponer en esta memoria una definición formal de segmentación borrosa de imágenes en términos grafo-teóricos, esto es, a través de la red de la imagen. Además, se plantea un algoritmo que lleve a cabo esta extensión de segmentación nítida a un contexto borroso. Para lograr esta extensión, se planteó utilizar como paso intermedio una segmentación jerárquica, el cual obtiene una secuencia de

segmentaciones nítidas con distintos niveles de detalle. Luego, se observó que no existía en la literatura consultada una definición formal matemática de la segmentación jerárquica de imágenes, tratada en algunas investigaciones previas (ver por ejemplo Arbeláez et al., 2011; Cheng y Sun, 2000; Lhermitte et al., 2008; Schroeter y Bigün, 1995) aunque no a partir de una base formal rigurosa. Esto motivó una tercera línea de trabajo en el desarrollo de esta memoria, consistente en proponer una definición matemática formal de la segmentación jerárquica también en términos de la red de la imagen.

A su vez, los desarrollos prácticos de estos dos conceptos están relacionados con la tesis de (Zarrazola, 2013), por lo que la continuación de su investigación también fue un motivo para el desarrollo de esta memoria. En su tesis, se desarrolló un algoritmo de segmentación jerárquica en redes denominado Divide-and-Link (D&L). Específicamente, el D&L se basa en un proceso jerárquico no supervisado, que expresa las particiones de su salida como regiones. Este algoritmo está enfocado en el análisis de redes, en particular se evaluó en términos de modularidad (Gómez et al., 2015b). El D&L llegó a ser aplicado para la segmentación de imágenes, pero, sin embargo, al no estar enfocado en el contexto de tratamiento de imágenes, presentaba problemas al evaluar las salidas del mismo. Entre estos problemas estaba que al momento de la publicación del algoritmo D&L, no existían metodologías de evaluación para algoritmos de segmentación en la literatura investigada. Además, al visualizar la salida generada por el algoritmo en donde cada región es coloreada de un color dentro de la escala de grises, se podía presentar el problema de que, al obtener una salida con más de 256 regiones, ya no existirían más colores para su visualización.

De este modo, en esta memoria se transforma el D&L en un algoritmo de detección de bordes, que se denominará Image Divide and Link (ID&L), y se lleva a cabo su evaluación mediante la metodología más extendida de comparación de bordes. Esto hace entonces que la representación de las regiones del D&L se convierta en un problema de detección de bordes. No obstante, las salidas de este nuevo algoritmo de detección de bordes, que consisten en particiones jerárquicas, no podían ser evaluadas directamente con los métodos de evaluación de bordes ya existentes debido a que son métodos que consideran una sola imagen (no una jerarquía) como salida. Por lo tanto, esto llevó a adaptar la metodología de evaluación de bordes de cara a evaluar una salida jerárquica, generando así en esta memoria un nuevo método de evaluación de bordes que permita comparar algoritmos jerárquicos con otros jerárquicos y no jerárquicos.

Una vez planteados los motivos que impulsaron al desarrollo de esta memoria, se mencionan en los siguientes dos apartados los objetivos y la metodología de la investigación presentada en esta memoria.

1.1. Objetivo general

Proponer, formalizar y clasificar diversos conceptos y algoritmos relevantes de cara al desarrollo de nuevas técnicas de tratamiento de imágenes.

1.2. Objetivos específicos

A continuación, se mencionan los objetivos específicos a desarrollar en esta memoria de acuerdo al orden de su elaboración:

1. Definir formalmente los conceptos de segmentación nítida, segmentación jerárquica y segmentación borrosa de imágenes en términos grafo-teóricos.
2. Plantear un algoritmo para construir una segmentación borrosa basado en una segmentación jerárquica.
3. Proponer una metodología de visualización para la segmentación nítida, jerárquica y borrosa de imágenes.
4. Clasificar las principales tareas de tratamiento de imágenes de acuerdo a la salida que estas consiguen.
5. Desarrollar un nuevo detector de bordes a partir del algoritmo de segmentación jerárquica de redes D&L.
6. Establecer una metodología de evaluación de bordes para algoritmos con salidas jerárquicas.
7. Evaluar y comparar el rendimiento del nuevo detector de bordes con esta nueva metodología.

1.3. Metodología

Para el cumplimiento de los objetivos planteados, se siguió la siguiente metodología:

- Para el análisis y desarrollo del concepto de segmentación borrosa fue necesario definir previamente los conceptos de segmentación nítida y jerárquica en base a la red de una imagen digital, es decir, representado la imagen como un grafo, en donde los nodos representan los píxeles y las aristas representan la relación entre píxeles vecinos y, a su vez, considerando medidas de disimilaridad entre píxeles vecinos. Las definiciones de estos conceptos se han publicado en (Gómez et al., 2015a).

- Una vez formalizados estos conceptos, se propuso un algoritmo para construir una segmentación borrosa basado en una segmentación jerárquica. De este modo, se extiende de forma natural el concepto de segmentación nítida a un contexto borroso. Esto es, la segmentación nítida realiza una partición de píxeles de una imagen en regiones. Luego, con la segmentación jerárquica se obtiene una secuencia de particiones con distintos niveles de detalle ordenados jerárquicamente. Finalmente, con la segmentación borrosa se pueden obtener tantas particiones como sea posible indicando distintos grados de pertenencia de cada píxel a una partición.
- Se formalizaron metodologías de visualización de una segmentación nítida, jerárquica y borrosa. Específicamente, la metodología de visualización propuesta de segmentación borrosa se basa en la adaptación de la visualización de segmentación jerárquica de imágenes. Así, la salida de esta metodología de visualización de segmentación borrosa presenta gradaciones en la salida final y, no sólo en los pasos intermedios del procedimiento como sucede con otras metodologías de segmentaciones que utilizan técnicas borrosas pero obtienen una salida nítida. La propuesta de la metodología de visualización de segmentación borrosa también se ha publicado en (Gómez et al., 2015a).
- Para la nueva clasificación de algunos problemas de tratamiento de imágenes fue necesario realizar en primer lugar una revisión bibliográfica sobre las distintas técnicas que resuelven estos problemas. Luego, se analizó en cada una de las técnicas el tipo de salida que generaban. Posteriormente, se clasificaron algunos problemas de tratamiento de imágenes de acuerdo a la salida que generan y al objetivo que consiguen. Como resultado de esta investigación, se ha publicado en (Guada et al., 2016), una clasificación sobre las principales tareas de tratamiento de imágenes según la salida que cada una produce, en donde se muestra una descripción de cada una de ellas y, a su vez, dentro de esta nueva clasificación, las tareas fueron subclasificadas en base a su uso de métodos supervisados o no supervisados. Asimismo, dentro de esta clasificación se reflejó también el uso o no de un formalismo de representación borroso.
- La generación y visualización de los dos conceptos nuevos de segmentación jerárquica y borrosa basados en teoría de grafos, conllevaron también la creación de un algoritmo de detección de bordes. En resumen, el algoritmo propuesto primero divide jerárquicamente una imagen digital de cualquier tipo (escala de grises, RGB, CIELab, hiperespectrales, etc.) en regiones, representando cada elemento de la secuencia con un mapa de regiones, y luego transforma dicha secuencia de mapas de regiones en una secuencia de mapas de bordes binarios, en donde los

píxeles de color blanco representan el contorno de las regiones y los píxeles de color negro representan las regiones en sí. Este algoritmo está basado en la adaptación de un algoritmo de agrupamiento jerárquico para redes, llamado Divide-and-Link (D&L) (Gómez et al., 2015b), a un contexto de imágenes. De este modo, la idea principal del algoritmo propuesto es extender el algoritmo de partición jerárquica de redes D&L con el fin de obtener una secuencia jerárquica de mapas de regiones (Gómez et al., 2015a), y luego transformar dichos mapas de regiones en una secuencia de mapa de bordes. Este nuevo algoritmo se denominó Image-Divide-and-Link (ID&L) y se publicó en (Guada et al., 2018).

- En vista que la salida del algoritmo propuesto ID&L consiste en una secuencia de mapas de bordes, en los cuales los objetos encontrados son mostrados jerárquicamente, se puede considerar esta secuencia como un todo, de forma que esta se puede corresponder a una aproximación de los distintos niveles de detalle que varios humanos pueden identificar al reconocer objetos en una imagen¹. Además, otra de las ventajas de obtener una secuencia de bordes como salida de un algoritmo, es que permite escoger un elemento de la secuencia de interés, u obtener un mapa de bordes suavizado (*soft*) agregando todas las particiones de la secuencia y asignando un nivel de intensidad al borde de acuerdo a su nivel de jerarquía correspondiente, estableciendo así la metodología de visualización de la segmentación borrosa de imágenes propuesto en los objetivos de esta memoria (como se muestra en el artículo Gómez et al., 2015a). En consecuencia, en esta memoria también se plantea cómo evaluar la salida de un algoritmo jerárquico de detección de bordes o, en otras palabras, cómo evaluar la secuencia jerárquica de mapas de bordes como un todo. El método de evaluación propuesto en esta memoria, consiste en una adaptación del método de evaluación de bordes establecido por (Martin et al., 2004) y (Arbeláez et al., 2011), el cual utiliza la base de datos creada por (Martin et al., 2001). La idea general de este método consiste en hacer una comparación muchos-a-muchos (*many-to-many*) entre la secuencia de particiones jerárquicas y los distintos niveles de detalle identificados por humanos sobre las imágenes de la base de datos, que sirven como imágenes de referencia. Igualmente, una ventaja de la adaptación de este método de evaluación es que permite también comparar la secuencia de imágenes obtenida con un algoritmo jerárquico con la salida de otros algoritmos, bien sean

¹En un estudio realizado por (Martin et al., 2001), se presenta una base de datos que contiene una serie de imágenes de escenas naturales, en donde cada una viene acompañada por otras las cuales representan las segmentaciones realizadas por humanos sobre ellas y así usarlas como referencia (*ground truth*) al evaluar las salidas producidas por un algoritmo sobre las imágenes originales.

jerárquicos o no. En el apartado de los resultados computacionales de esta memoria, se muestran los resultados obtenidos en la evaluación del algoritmo propuesto de detección de bordes ID&L con respecto a las referencias humanas de la base de datos de (Martin et al., 2004), y se comparan las salidas obtenidas de dicho algoritmo con los algoritmos de detectores de bordes de Canny (Canny, 1986) y Sobel (Sobel, 2014). La publicación de este nuevo método de evaluación de bordes también fue publicado en (Guada et al., 2018; Rodríguez et al., 2016).

1.4. Estructura de la memoria

En el capítulo 2 se exhiben los conceptos base para una mejor comprensión de esta memoria. Específicamente, la sección 2.1 presenta el concepto matemático de una imagen digital, además, presenta las definiciones de imagen binaria, imagen en escala de grises e imagen en color RGB. Seguidamente, en la sección 2.2 se exponen los conceptos concernientes a la teoría de grafos y, cómo puede ser asociado un grafo a una imagen digital. En la sección 2.3 se recuerda la definición formal de segmentación nítida, así como el concepto de segmentación basado en la teoría de grafos. Luego, se describen brevemente la segmentación jerárquica en redes y un algoritmo afín, el D&L (sección 2.4). Posteriormente, en la sección 2.5 se detalla el concepto de detección de bordes y se explican dos operadores básicos para la detección de bordes. Y para finalizar este capítulo, se mencionan los conceptos relacionados con la teoría de conjuntos borrosos (sección 2.6).

Una vez esclarecidos los conceptos que sirven de base para el desarrollo de esta memoria, en el capítulo 3 se propone una definición formal de segmentación borrosa de imágenes basada en teoría de grafos. Para ello, es necesario trasladar el concepto de segmentación nítida a un contexto borroso. Por tanto, en la sección 3.1 se formula una definición equivalente de segmentación nítida de imágenes a través de las aristas de un grafo, la cual es extendida en la sección 3.2 para definir la segmentación jerárquica de imágenes. Y así, en la sección 3.3 se presenta la definición de segmentación borrosa basada en la teoría de grafos y un algoritmo para construir una segmentación borrosa basado en una segmentación jerárquica. En la sección 3.4, se presenta la metodología de visualización de estos tres conceptos. Por otro lado, en la sección 3.5 se muestra la clasificación realizada de las principales tareas de tratamiento de imágenes.

En el capítulo 4, se propone un nuevo algoritmo de detección de bordes, ID&L, basado en el algoritmo de segmentación jerárquica en redes D&L (sección 4.1). Específicamente, en la subsección 4.1.1, se recuerda cómo asociar un grafo a una imagen. Seguidamente, en la subsección 4.1.2 se describe cómo calcular los umbrales determinantes para realizar las particiones en el proceso iterativo. En la subsección 4.1.3 se detalla el procedimiento para determinar

mapas de regiones jerárquicos. Luego, en la subsección 4.1.4 se describe el procedimiento para contraer regiones homogéneas para que no se subdividan al realizar la partición jerárquica. Consecutivamente, en la subsección 4.1.5 se explica el procedimiento para transformar el mapa de regiones a un mapa de bordes.

En el capítulo 5, se propone una metodología de evaluación para secuencias jerárquicas de mapas de bordes a través de la adaptación de la extendida metodología de evaluación de bordes de (Arbeláez et al., 2011). Para ello, en la sección 5.1 se describe una de las metodologías de evaluación a través de bordes más utilizada en la literatura, que ha servido como base para proponer en la sección 5.2 una nueva metodología de evaluación de bordes para salidas jerárquicas.

En el capítulo 6, se presenta específicamente en la sección 6.1 el experimento computacional llevado a cabo para evaluar y comparar el rendimiento junto con la eficacia predictiva del nuevo algoritmo de detección de bordes con la de otros detectores. Y, en la sección 6.2 se muestran los resultados obtenidos en la competición de detección de bordes efectuada en el congreso EUSFLAT 2017.

Finalmente, en el capítulo 7 se presentan las conclusiones (sección 7.1), las contribuciones y publicaciones resultantes del desarrollo de esta memoria (sección 7.2) y las futuras líneas de investigación previstas para la continuación de la investigación reflejada en esta memoria (sección 7.3).

Capítulo 2

Preliminares

*La historia humana es en esencia una
historia de ideas*

Herbert George Wells

RESUMEN: En este capítulo se presentan todos los conceptos que han servido de base para el desarrollo de esta memoria. En primer lugar, se introducen formalmente los conceptos de imagen digital, imagen binaria, imagen en escala de grises e imagen RGB (sección 2.1). En segundo lugar, se mencionan los conceptos relacionados con la teoría de grafos y, cómo asociar un grafo a una imagen digital (sección 2.2). En tercer lugar, se presenta la definición de la segmentación de imágenes sobre los nodos de un grafo (sección 2.3). En la sección 2.4 se presenta un algoritmo de segmentación jerárquica en redes. Seguidamente, en la sección 2.5 se expone la definición de la detección de bordes. Finalmente, en la sección 2.6 se describen brevemente los conceptos relacionados con la teoría de conjuntos borrosos.

2.1. Imagen digital

Una imagen digital puede considerarse como una representación discreta de datos que poseen información espacial (disposición) e intensidad (color) (Solomon y Breckon, 2011).

Una imagen digital puede ser modelada por una función bidimensional $f(i, j)$, donde i y j representan las coordenadas en el plano y $f(i, j)$ representa cada elemento de la imagen o píxel (unidad de información mínima) por un número fijo de características medibles (González et al., 2009).

Definición 2.1.1 (Imagen digital) Una imagen digital I de dimensión $r \times s$ se puede modelizar como una función bidimensional $I = \{f(i, j), i = 1, \dots, r; j = 1, \dots, s\}$, donde (i, j) se refiere a la posición del píxel en la imagen, y $f(i, j)$ representa la información espectral visible asociada a la intensidad del píxel (i, j) .

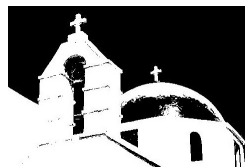
Una imagen digital I puede ser de varios tipos, algunos de ellos se describen a continuación:

Binaria $\equiv f(i, j) \in \{0, 1\}$: una imagen binaria es una función bidimensional que asigna un valor numérico del conjunto $\{0, 1\}$ a cada píxel de la imagen. Así, la imagen viene representada en dos tonos de colores: blanco y negro. Por lo general, el valor 0 se refiere al color negro (píxel de fondo) y el valor 1 al color blanco (píxel en primer plano) (Solomon y Breckon, 2011).

Escala de grises $\equiv f(i, j) \in \{0, \dots, 255\}$: una imagen en escala de grises consiste en una función bidimensional $f(i, j)$, en donde la amplitud de f en cualquier par de coordenadas (i, j) se refiere a la intensidad (nivel de gris) de la imagen en dicho punto (González et al., 2009).

RGB $\equiv f(i, j) \in \{0, \dots, 255\}^3$: una imagen RGB está formada por tres componentes de igual dimensión que constituyen la imagen: rojo, verde y azul; es decir, se tiene un vector de tres matrices $f(i, j) = [f(i, j)^1, f(i, j)^2, f(i, j)^3]$ que representa la intensidad de color de un píxel compuesto por esos tres colores.

En las Figuras 2.1a, 2.1b y 2.1c, se muestran las imágenes de una iglesia de la base de datos de Berkeley (BSDS500) (Martin et al., 2001) en escala binaria, en escala de grises y en RGB, respectivamente.



(a) Imagen en escala binaria



(b) Imagen en escala de grises



(c) Imagen en escala RGB

Figura 2.1: Ejemplo de la representación de una imagen

2.2. Teoría de grafos

Esta sección está dedicada a introducir los conceptos y las notaciones básicas de la teoría de grafos, específicamente en el contexto de imágenes digitales.

2.2.1. Conceptos básicos y notación de grafos en el contexto de imágenes digitales

Los grafos son conocidos por ser una estructura con una larga historia en las matemáticas y que han sido muy utilizados en la comunidad científica (Lezoray y Grady, 2012). Los grafos son muy útiles debido a que pueden modelar gráficamente cualquier situación del mundo real mediante un diagrama conformado por un conjunto de puntos y líneas (Bondy et al., 1976). Formalmente, los grafos representan un conjunto de elementos y un conjunto de relaciones entre pares de dichos elementos. Los elementos se denominan *nodos* o *vértices* y las relaciones entre ellos se llaman *aristas* (Lezoray y Grady, 2012).

Esta sección se centra en aquellos conceptos de la teoría de grafos que han sido utilizados en esta memoria, con especial énfasis en aplicaciones de la teoría de grafos en problemas de tratamiento de imágenes.

Definición 2.2.1 (Grafo) *Un grafo $G = (V, E)$ es una tupla compuesta por un conjunto finito de vértices o nodos V y un conjunto finito de aristas E , en donde cada arista enlaza un par de nodos no ordenados.*

En la Figura 2.2 se muestra un grafo con cuatro nodos y cuatro aristas.

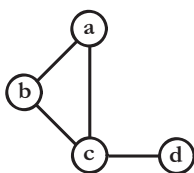


Figura 2.2: Grafo con cuatro nodos $\{a, b, c, d\}$ y cuatro aristas $\{e_{ab}, e_{ac}, e_{bc}, e_{cd}\}$

Una imagen digital puede ser asociada a un grafo, en donde los nodos representen a los píxeles de la imagen y las aristas las relaciones entre pares de píxeles. De esta manera, la imagen digital puede ser modelada por una estructura bidimensional de píxeles, en donde cada uno está conectado a su vecindad natural en la imagen (Gómez et al., 2007). El tipo de grafo asociado a una imagen variará en función de la vecindad o topología de las relaciones entre píxeles considerada (Gómez et al., 2015a).

Definición 2.2.2 (Grafo asociado a la imagen) *Dada una imagen digital I de dimensión $r \times s$ con conjunto de píxeles $\{(i, j), i = 1, \dots, r; j = 1, \dots, s\}$. Sea $G = (V, E)$ el grafo de la imagen I , en donde V representa el conjunto de píxeles y E el conjunto de aristas. Dos píxeles o nodos $(i, j), (i', j')$ son vecinos o adyacentes si y solo si comparten una coordenada y si son contiguos. Así, una arista $e = \{a, b\}$ (también denotada como e_{ab}) existe en E si y solo si los píxeles $a = (i, j)$ y $b = (i', j')$ son adyacentes.*

La adyacencia o vecindad considerada en la definición 2.2.1 corresponde con una topología de 4-vecinos (Figura 2.3(a)). No obstante, otras topologías como las que se muestran en las Figuras 2.3(b-d), también pueden ser utilizadas para representar la vecindad de los píxeles (Zarrazola, 2013).

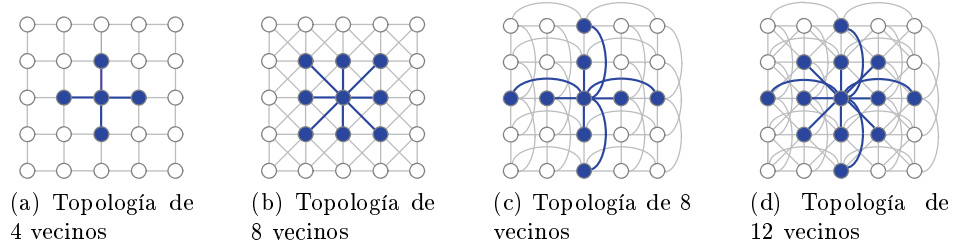


Figura 2.3: Algunas topologías de vecinos sobre un grafo

El grafo mencionado anteriormente (definición 2.2.2) permite modelar las relaciones que existen entre los píxeles dentro de una imagen. A este grafo se le pueden añadir las diferencias que existen entre píxeles adyacentes en base a sus características espectrales, intensidad, brillo, textura, forma o distancia espacial. Estas discrepancias o diferencias entre píxeles adyacentes se denominarán en esta memoria como *disimilitudes* y serán denotadas por d_{ab} siendo $a = (i, j)$ y $b = (i', j')$ dos píxeles adyacentes.

Por ejemplo, si tenemos los dos primeros píxeles $a = (1, 1)$ y $b = (1, 2)$ de una imagen en escala de grises con $f(1, 1) = 100$ y $f(1, 2) = 107$, una posible disimilitud asociada a la arista e_{ab} sería $|100 - 107| = 7$.

Una vez construida la medida de disimilitud entre píxeles adyacentes y sea incorporada al grafo asociado a la imagen, se tiene lo que se denominará red asociada a una imagen.

Formalmente, dada una arista e_{ab} que une a dos píxeles $a = (i, j)$ y $b = (i', j')$, sea $d_{ab} \geq 0$ el *grado de disimilitud* entre ellas. Además, sea $D = \{d_{ab} | e_{ab} \in E\}$ el conjunto de todas las disimilitudes.

De este modo, la información contenida en una imagen digital, puede ser resumida mediante una red $N(I) = \{G; D\}$ (Gómez et al., 2007).

Definición 2.2.3 (Red) Una red N es un grafo cuyos nodos y/o aristas tienen asociados unos atributos.

Otro concepto importante y que también será útil en lo que resta de este trabajo, es la partición de una red.

Definición 2.2.4 (Partición de una red) Dada una red $N = \{G = (V, E); D\}$, se dice que un conjunto $\mathcal{P} = \{C_1, \dots, C_t\}$ es una partición \mathcal{P} de la red N si se cumple lo siguiente:

1. Cubrimiento: $\cup_{j=1}^t C_j = V$.

2. No exista solapamientos: $C_j \cap C_j = \emptyset$ para todo $i \neq j$.
3. Conectividad: para todo i , el subgrafo $(C_i, E_{|C_i})$ es conexo.

Ejemplo 2.2.1 (Modelización de una imagen como una red) En la Figura 2.4 se muestra una imagen de tamaño 4×5 conformada por tres colores: blanco, azul y rojo, en donde el grado de disimilitud de las aristas son: 0 si los píxeles vecinos poseen el mismo color, 0.3 entre píxeles de colores blanco y azul, 0.6 entre píxeles de colores rojo y azul y 0.9 entre píxeles de colores rojo y blanco.

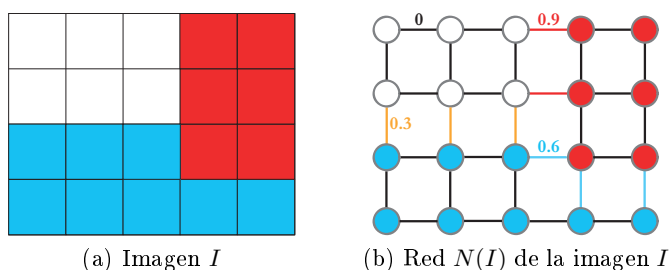


Figura 2.4: Ejemplo de la modelización de una imagen como una red de dimensiones 4×5

Cómo medir la disimilitud entre dos píxeles adyacentes en base a sus características espectrales, es un problema difícil de resolver. Aunque en el caso de imágenes en escala de grises se acepta de manera natural la diferencia en valor absoluto, no queda claro como extender esto al caso de tres dimensiones. Algunos autores usan la distancia Euclídea entre dos nodos como medida de disimilitud (Gómez et al., 2007), otros proponen otras métricas para construir dicha disimilitud. También hay autores que debido a la diversidad de posibilidades proponen agregar estas disimilitudes para construir números borrosos (ver Gómez et al., 2015a).

En este trabajo, el grado de disimilitud d entre dos nodos vecinos se calcula con la distancia *CIELab*. Para ello, las imágenes de tipo RGB se transforman a un espacio de color *CIELab* (también conocido como *CIE76*) (Ramanath et al., 2004), el cual es una buena medida de color por ser un espacio de color más uniforme (Connolly y Fleiss, 1997). Las siglas *Lab* se refieren al espacio de color tridimensional, en donde L representa la luminosidad de negro a blanco, A de rojo a verde y b el gradiente del azul. Entonces, sean dos píxeles de color $a = (i, j)$, $b = (i', j')$ en el espacio RGB $a^{\text{rgb}} = [r_a, g_a, b_a]$ y $b^{\text{rgb}} = [r_b, g_b, b_b]$, su representación en el espacio *CIELab* viene dado por $a^{\text{lab}} = [L_a, a_a, b_a]$ y $b^{\text{lab}} = [L_b, a_b, b_b]$ respectivamente. Luego, la distancia *CIELab* entre a^{lab} y b^{lab} se mide mediante la siguiente ecuación:

$$d_{ab}^{\text{lab}} = \sqrt{(\Delta L)^2 + (\Delta a)^2 + (\Delta b)^2} \quad (2.1)$$

donde $\Delta L = L_b - L_a$, $\Delta a = a_b - a_a$ y $\Delta b = b_b - b_a$.

2.2.2. Subgrafo, árbol soporte y bosque soporte

Las relaciones entre los nodos de los grafos involucran otros conceptos que se presentan en este apartado.

Definición 2.2.5 (Subgrafo) *Un grafo $G' = (V', E')$ es un subgrafo de $G = (V, E)$ si $V' \subseteq V$ y $E' = \{e_{ab} \in E \mid a \in V' \text{ y } b \in V'\}$.*

En particular, los árboles y bosques soporte o de extensión son conceptos muy importantes y utilizados en la teoría de grafos. Antes de mostrar sus definiciones, se presentan los conceptos de recorrido, camino, ciclo y conectividad.

Definición 2.2.6 (Recorrido) *Un recorrido en un grafo $G = (V, E)$ consiste en una secuencia de nodos y aristas cuyos elementos son alternativamente nodos y aristas de G .*

Definición 2.2.7 (Camino) *Un camino es un recorrido que no repite nodos.*

Definición 2.2.8 (Ciclo) *El ciclo es un camino cerrado en el que no se repite ningún nodo a excepción del primer nodo del camino que aparece dos veces como principio y fin del camino.*

Definición 2.2.9 (Conectividad) *Sea un grafo $G = (V, E)$. Dos nodos $(i, j), (i', j')$ están conectados si el grafo contiene al menos un camino desde el nodo (i, j) al nodo (i', j') . Un grafo G es conexo si todos los pares de nodos están conectados.*

Definición 2.2.10 (Árbol) *Un árbol es un grafo conexo que no posee ciclos y que posee las siguientes propiedades:*

- *Un árbol de n nodos contiene exactamente $n - 1$ aristas.*
- *Un árbol tiene al menos dos nodos hojas (i.e., nodos enlazados con una arista).*
- *Cada par de nodos de un árbol está conectado por un único camino.*

De las definiciones anteriores, se puede deducir que un camino siempre es un árbol. El concepto de árbol introduce el concepto de árbol de expansión, también conocido como árbol soporte, generador, o árbol recubridor (*spanning tree*).

Definición 2.2.11 (*Árbol soporte*) *Un árbol de soporte o de expansión de un grafo conexo $G = (V, E)$ es un árbol compuesto por todos los nodos de G .*

Una generalización del concepto de árbol soporte son los bosques soporte (*spanning forest*).

Definición 2.2.12 (*Bosque soporte*) *Un bosque soporte o de expansión consiste en una colección de árboles soporte en cada elemento conexo del grafo.*

2.3. Segmentación de imágenes

Alrededor de 1970, la segmentación de imágenes tuvo su auge como un tópico de investigación avanzado. El concepto de segmentación en tratamiento de imágenes puede verse como una técnica en la que se divide una imagen en múltiples regiones u objetos, con el fin de simplificar o modificar la representación de la imagen para que sea más significativa y fácil de analizar (Basavaprasad y Ravindra, 2014). Los píxeles que integran cada región u objeto poseen atributos similares (González et al., 2009). Los atributos más básicos para la segmentación, son la intensidad para una imagen monocromática y los componentes de color para una imagen de color. En resumen, las regiones de una imagen segmentada debieran ser uniformes y homogéneas en relación con algún atributo o propiedad (Pratt, 2001; Basavaprasad y Ravindra, 2014).

Formalmente, la segmentación de imágenes particiona una imagen en regiones no solapadas tales que cada región de píxeles conexos es homogénea (Pal y Pal, 1993; Fu y Mui, 1981).

Sin embargo, a pesar de que la segmentación es una técnica esencial para el tratamiento de imágenes y que tiene aplicaciones en muchos campos, no existe un método único aplicable a todos los tipos de imágenes. Las metodologías actuales de segmentación han sido clasificadas según determinadas características de los propios algoritmos (Ma et al., 2010; Basavaprasad y Ravindra, 2014).

Asimismo, el nivel de la subdivisión de la imagen en regiones es dependiente del problema a resolver, esto es, el número de regiones identificadas y aisladas se determina en función de los objetos de interés. No obstante, en la mayoría de los casos, las imágenes tratadas no son triviales y por ende la segmentación llega a ser más complicada. De este modo, existen técnicas de segmentación para detectar objetos con determinadas formas. Por ejemplo, está la segmentación de objetos curvilíneos, y un resumen de esta técnica en distintas aplicaciones se puede ver en (Bibiloni et al., 2016).

Actualmente existen muchas aplicaciones de segmentación de imágenes (Pal y Pal, 1993), por ejemplo en la visión por robot (Bruce et al., 2000),

detección de células cancerígenas (Al-Kofahi et al., 2010), identificación de cartílago de rodilla y segmentación de materia gris / materia blanca en imágenes de resonancia magnéticas (Grau et al., 2004), conducción autónoma (Sotelo et al., 2004), entre otros.

A continuación, se listan algunas técnicas básicas de segmentación de imágenes:

- Técnicas basadas en thresholding (Weszka et al., 1973; Sezgin y Sankur, 2004; Sahoo et al., 1988; Glasbey, 1993; Bustince et al., 2010).
- Técnicas basadas en watershed (Vincent y Beucher, 1989; Mangan y Whitaker, 1999; Rashedi y Nezamabadi-pour, 2013; Pratt, 2001; Basavaprasad y Ravindra, 2014; Beucher y Meyer, 1992; Roerdink y Meijster, 2000).
- Técnicas basadas en segmentación de regiones (Adams y Bischof, 1994; Pavlidis, 1982; Brice y Fennema, 1970; Trémeau y Colantoni, 2000; Horowitz y Pavlidis, 1974; Pavlidis, 1982).
- Técnicas basadas en partición de grafos (Felzenszwalb y Huttenlocher, 2004; Lladós et al., 2001; Grady, 2006).

Al modelar una imagen digital como un grafo, es posible extraer información significativa sobre la imagen tratada, comprobando las distintas propiedades del grafo. De este modo, los grafos resultan ser una herramienta extremadamente importante y útil para la segmentación de imágenes. En general, la teoría de grafos y la modelización de una imagen mediante un grafo permite de manera natural definir formalmente el concepto de segmentación tal y como se expone en la definición de (Pal y Pal, 1993). A continuación, se define formalmente la segmentación de imágenes sobre un grafo, específicamente sobre los nodos del grafo.

Definición 2.3.1 (*Segmentación sobre los nodos de un grafo*) Dada una red $N(I) = \{G; D\}$ asociada a una imagen I , se dice que la familia $S = \{R_1, \dots, R_t\}$ es una segmentación por nodos de la red de la imagen $N(I)$ si y solo si se cumplen las siguientes condiciones:

1. Las regiones no se solapan (i.e., para todo $i \neq j$, $R_i \cap R_j = \emptyset$).
2. Cubrimiento de todos los píxeles: $\bigcup_{j=1}^t R_j = V$ (todos los píxeles están cubiertos por regiones).
3. Conectividad en todas las regiones: para todo $j \in \{1, \dots, t\}$, el subgrafo $(R_j, E_{|R_j})$ es un grafo conexo.

La primera condición establece que las regiones deben ser disjuntas, es decir, que no haya solapamiento entre ellas. La segunda condición indica que

la segmentación debe ser completa, es decir, la unión de las regiones segmentadas R_i deben contener todos los píxeles de la imagen. La tercera condición indica que los nodos de cada región deben ser conexos (Zarrazola, 2013). Algunos autores añaden dos condiciones: $P(R_i) = VERDADERO$ para todo $i = 1, 2, \dots, n$ y $P(R_i \cup R_j) = FALSO$ para cualquier región adyacente R_i y R_j (ver González et al., 2009), para fijar algún tipo de homogeneidad de cada región y heterogeneidad entre regiones pero son condiciones muy poco operativas.

Además, se puede observar que desde un punto de vista matemático, el concepto de segmentación sobre los nodos de un grafo coincide con el concepto de partición de una red dado en la definición 2.2.4), para el caso particular en el que la red N sea un grafo asociado a una imagen.

Por otro lado, el conjunto de todas las segmentaciones por nodos sobre la red $N(I)$ vendrá denotado por $S^n(N(I))$.

En la Figura 2.5, se puede apreciar el concepto de segmentación basado en los nodos de la red de la Figura 2.4b; en ella se observan tres regiones separadas (blanca, azul y roja).

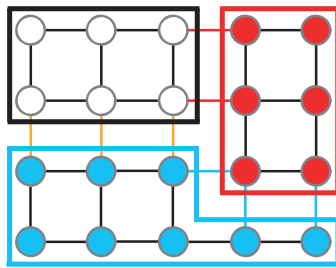


Figura 2.5: Segmentación mediante los nodos de una red $N(I)$

Los grafos han sido utilizados en muchas técnicas de segmentación con el fin de representar estructuras visuales complejas de visión por ordenador y reconocimiento de patrones. A continuación, se describen brevemente las técnicas más conocidas que utilizan una aproximación grafo teórica para realizar la segmentación:

Random walker: este algoritmo segmenta una imagen a partir de un número de píxeles predefinidos con etiquetas denominadas *semillas* que indican las distintas regiones a identificar (e.g., objeto y fondo). Estas semillas se encuentran etiquetadas en los píxeles correspondientes y, aquellos píxeles sin etiquetas serán marcados por el algoritmo mediante el siguiente planteamiento: “sea un *random walker* en un píxel sin etiquetar, ¿cuál es la probabilidad de que llegue primero a cada uno de los píxeles semillas?”. Para ello, se asigna un vector de K – *tuplas* a cada píxel, que especifique la probabilidad de que un *random walker* empiece desde cada píxel sin etiquetar, llegue primero a cada uno de

los K píxeles semillas. Luego, se selecciona de esas de K - *tuplas* la semilla más probable para el *random walker* (Grady, 2006). Más información se puede encontrar en (Wechsler y Kidode, 1979; Tetali, 1991; Grady, 2006).

Graph cuts: consiste en una herramienta de optimización multidimensional que se apoya en la teoría de grafos y, mediante el problema de minimización de la función de energía emplea el problema de flujo máximo de un grafo. De esta manera, a través del teorema de *max-flow min-cut* (flujo máximo - corte mínimo) se define un corte mínimo del grafo de forma que el tamaño del corte no sea más grande en ningún otro corte (para más detalle ver Greig et al., 1989; Boykov y Veksler, 2005; Boykov y Kolmogorov, 2004; Boykov y Funka-Lea, 2006; Sinha, 2004).

Normalized cuts: este procedimiento trabaja con grafos $G = (V, E)$ ponderados, en donde el peso de cada arista $w(i, j)$ es una función de la similitud entre los nodos i y j (Shi y Malik, 2000). Un grafo $G = (V, E)$ se divide óptimamente en dos subconjuntos disjuntos A y B , con $A \cup B = V$ y $A \cap B = \emptyset$, al minimizar su valor $Ncut$. El grado de disimilaridad de A y B se mide como una fracción de las conexiones de las aristas totales sobre todos los nodos en el grafo denominándose *normalized cut (Ncut)* (Shi y Malik, 2000; Tatiraju y Mehta, 2008).

Region adjacency graphs: este algoritmo proporciona una vista espacial de la imagen, para poder manipular la información contenida en ella de manera eficiente, a través de un grafo con el cual asocia cada nodo como una región y con las aristas enlazan cada par de regiones adyacentes (Trémeau y Colantoni, 2000). Una region R_i es definida por una cuádrupla $R_i = (k, x_k, y_k, A_k) | k \in (1, \dots, n)$, en donde k se refiere al número de índice de la región, (x_k, y_k) representa el centro de gravedad y A_k al área (Pavlidis, 1982; Trémeau y Colantoni, 2000; Flavell et al., 1991).

En la Figura 2.6, se puede observar la salida obtenida con el algoritmo Normalized Cuts conformada por seis particiones realizadas a la imagen 2.1a de la base de datos de Berkeley (BSDS500) (Martin et al., 2001).

2.4. Segmentación jerárquica en redes

Al igual que pasa con los problemas de agrupamiento o clustering estadístico en los que se busca una partición del conjunto de objeto de estudio (en nuestro caso píxeles de una imagen), la idea de segmentación de una imagen presenta algunos inconvenientes que pueden ser parcialmente resueltos por el concepto de jerarquía. Una partición jerárquica desde un punto de vista

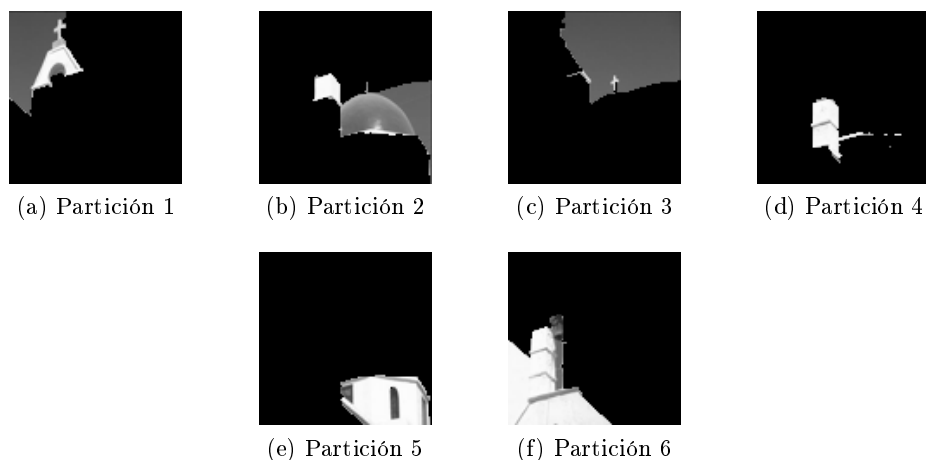


Figura 2.6: Salida del Normalized Cuts sobre la imagen 2.1a

matemático es una sucesión de particiones que muestra como los grupos de objetos (regiones de píxeles en nuestro caso) van formándose (aglomerativo) o rompiéndose (divisivo) de manera dinámica.

La segmentación jerárquica, específicamente en redes, consiste en agrupar nodos de la red en una serie de particiones conforme a un criterio partiendo desde un sólo grupo que incluya todos los individuos, hasta n grupos en donde cada uno contenga un único individuo (Zarrazola, 2013). Esto se puede realizar de dos formas, mediante *métodos aglomerativos* (Newman, 2004a) o *métodos divisivos* (Girvan y Newman, 2002a), en donde las agrupaciones vendrán representadas por un dendograma que se basa en un diagrama de dos dimensiones que expone la evolución de una agrupación jerárquica (Richards y Jia, 1999).

En más detalle, los métodos aglomerativos se encargan de fusionar sucesivamente los n individuos en grupos. En caso contrario, los métodos divisivos optan por separar los n individuos progresivamente en agrupaciones más reducidas. Con los algoritmos de segmentación jerárquica se tiene la ventaja de obtener una evolución de principio a fin de cómo se forman los grupos (métodos aglomerativos) o cómo se fraccionan (métodos divisivos) (Zarrazola, 2013).

Una de las ventajas de la segmentación jerárquica, es que permite al usuario predefinir el número de particiones necesarias y/o escoger de ellas aquella partición de interés. En más detalle, la secuencia obtenida muestra una secuencia anidada de agrupamientos que componen la jerarquía. Por tanto, el mejor corte para el usuario o para determinado problema estará en una determinada altura, en donde las agrupaciones por debajo de dicha altura son distantes entre sí (Zarrazola, 2013).

La segmentación jerárquica de redes involucra el problema de partición

jerárquica en redes (Zarrazola, 2013). Antes de exponer en esta sección el concepto de partición jerárquica, se recuerda que el concepto de partición de una red se expuso en la definición 2.2.4, luego, puede existir un orden parcial entre dos particiones.

Definición 2.4.1 (Relación de orden entre particiones) Dadas dos particiones \mathcal{P}, \mathcal{Q} de un conjunto de elementos V , se dice que \mathcal{P} es más fina que \mathcal{Q} (y se denota como $\mathcal{P} \subseteq \mathcal{Q}$) si para todo $C \in \mathcal{P}$, entonces existe $C' \in \mathcal{Q}$, tal que $C \subseteq C'$.

Definición 2.4.2 (Partición jerárquica) Dada la red $N = \{G; D\}$, y sea $\mathcal{S} = (\mathcal{P}^0, \mathcal{P}^1, \mathcal{P}^2, \dots, \mathcal{P}^K)$ una secuencia de particiones de dicha red. Entonces, \mathcal{S} es una partición jerárquica de N si se cumple lo siguiente:

- Existen dos particiones triviales: la primera \mathcal{P}^0 que contiene todos los nodos, y la última \mathcal{P}^K .
- $|\mathcal{P}^t| > |\mathcal{P}^{t-1}|$ para todo $t = 1, \dots, K$ (i.e., en cada iteración se incrementa el número de grupos en la red).
- $\mathcal{P}^t \subseteq \mathcal{P}^{t-1}$ para todo $t = 1, \dots, K$ (i.e., \mathcal{P}^t más fina que \mathcal{P}^{t-1}).

Esta definición se propone por primera vez en (Gómez et al., 2015b) y también se utiliza en (Guada et al., 2018). Por otro lado, una coloración binaria de grafos induce a una partición de un grafo, esto es, dado un nodo $a = (i, j)$ coloreado de binariamente (i.e. $col(a) \in \{0, 1\}$), el color que se asigna a un nodo vecino $b = (i', j')$ dependerá de la medida de disimilitud de $e_{a,b}$ comparado con un valor umbral establecido α (para más detalle ver Zarrazola, 2013). Un ejemplo de una segmentación jerárquica en redes se presenta a continuación.

Ejemplo 2.4.1 Sea la red $N(I)$ dada en la Figura 2.7. Se aplica el proceso de segmentación jerárquica a dicha red utilizando el siguiente conjunto de valores para α : $\{\alpha = 6.2, \alpha = 3, \alpha = 1.2, \alpha = 1, \alpha = 0\}$.

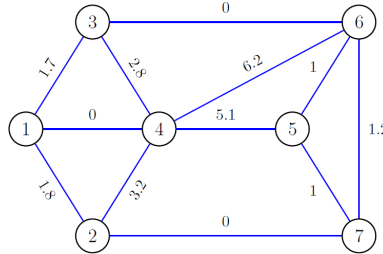


Figura 2.7: Red $N(I)$

Al ordenar el conjunto de umbrales en orden decreciente se tiene que $\alpha \in \{\alpha = 6.2 \geq \alpha = 3 \geq \alpha = 1.2 \geq \alpha = 1 \geq \alpha = 0\}$. El resumen de la secuencia de segmentación jerárquica (incluyendo un valor de $\alpha = 7$ en donde todos los nodos pertenecen al mismo grupo) se presenta en la Figura 2.8.

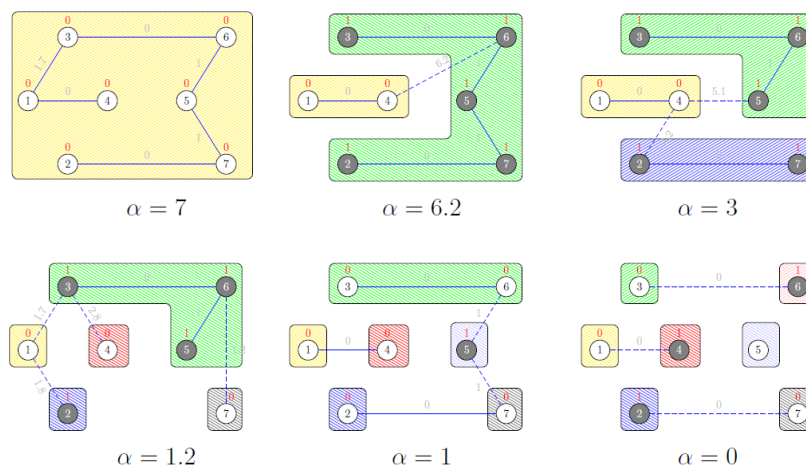


Figura 2.8: Segmentación jerárquica de la red $N(I)$

Una vez definido el concepto de segmentación jerárquica de una red, a continuación, se propone un algoritmo que será utilizado para obtener una segmentación de imágenes.

2.4.1. Algoritmo de segmentación jerárquica en redes

El análisis de redes complejas es un tema amplio que despierta un interés creciente en los últimos años. Específicamente, el estudio de la topología y las propiedades de las redes desempeñan un papel importante en ciencias de la computación, biología, ciencias sociales, entre otros campos (ver por ejemplo Fortunato, 2010; Sun et al., 2013; Girvan y Newman, 2002b; Gómez et al., 2013; Newman, 2004b; Wasserman y Faust, 1994). En particular, las redes sociales complejas representan un campo de aplicación interesante. Descubrir comunidades y estructuras inherentes en una red social debe ser un objetivo principal cuando se busca una mejor comprensión de una determinada red (Gómez et al., 2015b).

Sin embargo, en algunas situaciones, como por ejemplo en el proceso de división de una empresa o un grupo de amigos, este proceso se realiza de forma dinámica, por lo que sería relevante conocer todo el proceso de división en lugar de solo la imagen final. En tales casos, un enfoque de agrupamiento jerárquico parece más apropiado, ya que el proceso de división global se puede visualizar, trayendo ventajas específicas. Por ejemplo, como se señala en (Clauset et al., 2008), el conocimiento de la estructura jerárquica puede

usarse para predecir las conexiones faltantes de una manera más realista. Como se destaca en (Clauset et al., 2008), la jerarquía es un principio organizador central de redes complejas, capaz de ofrecer una visión de muchos fenómenos de red (véase también Espinilla et al., 2014), donde se enfatiza la importancia de comprender un problema frente a enfoques automáticos para problemas complejos de toma de decisiones). Por estas razones, en este trabajo, nos hemos centrado en las redes de agrupamiento jerárquico (Gómez et al., 2015b).

Un algoritmo que da una solución al problema de partición jerárquica para una red cualquiera, es el algoritmo Divide-and-Link (D&L). El algoritmo D&L es un algoritmo de segmentación jerárquica en redes que fue desarrollado en (Zarrazola, 2013) y extendido en (Gómez et al., 2015b). En detalle, consiste en un algoritmo iterativo binario no supervisado, que obtiene una partición jerárquica de una red representada en un dendograma, y está enmarcado en los procesos de partición de redes a través de grafos.

El algoritmo D&L parte de un grafo $G = (V, E)$ y dos pesos entre las aristas adyacentes del grafo: grado de disimilitud d y grado de similitud o afinidad l , obteniendo así la red $N = \{G; d, l\}$. Las aristas son clasificadas en:

Aristas de división: $e_d \subset E$ las cuales dividirán los nodos en el proceso de partición y se identifican ordenando las aristas de mayor a menor según su grado de disimilitud.

Aristas de enlace: $e_l \subset E$ que mantendrán los nodos enlazados en el proceso de partición y se identifican al ordenar las aristas de mayor a menor de acuerdo a su grado de afinidad.

Una vez ordenadas las aristas, se realiza un *coloreado binario de grafo* de forma que los nodos V se clasifican en dos clases: V_0 y V_1 , de acuerdo a las aristas que estén enlazados a ellos siguiendo el camino del bosque soporte. Esto es, los nodos unidos a una arista de división se asignan a diferentes clases, y aquellos nodos que estén unidos a aristas de enlace se asignan a la misma clase. Este procedimiento se realiza iterativamente hasta que se hayan *coloreado* todos los nodos de la red.

En resumen, el algoritmo D&L consta de los siguientes pasos (Gómez et al., 2015b):

1. Calcular los pesos de división d y enlace l para cada arista $e \in E$ de la red N denotado por e_d y e_l respectivamente.
2. Calcular $\alpha = \max_{e \in E} \{d_e\}$ y encontrar $E_{\max} = \{e \in E / d_e = \alpha\} \subset E$.
3. Disponer secuencialmente las aristas de la red N , seleccionando de primero las aristas en E_{\max} (i.e., las aristas que van a dividir la red). Ordenar las aristas restantes $l_{e_1} > l_{e_2} > \dots$ (i.e., las aristas de enlace).

- se ordenan desde la que tenga mayor grado de afinidad hasta el más bajo).
4. Construir un bosque soporte S , de manera similar al algoritmo de Kruskal (Kruskal, 1956), basados en la disposición encontrada en el paso previo.
 5. Construir la partición \mathcal{P}^t con los nodos conectados en el bosque soporte S^* que se han obtenido al eliminar las aristas de S pertenecientes a las aristas de división.
 6. Actualizar el conjunto de aristas E eliminando aquellas que unan los distintos grupos de \mathcal{P} .
 7. Repetir los pasos 1 hasta 6 siempre que $E \neq \emptyset$.

2.5. Detección de bordes

Los problemas de detección de bordes asociados a una imagen han sido un tema importante en el tratamiento de imágenes desde la década de los sesenta, por su capacidad de proporcionar información relevante de la imagen y proporcionar bordes de la superficie de objetos en una imagen.

En general, los métodos de detección de bordes buscan cambios en la función de intensidad de una imagen, de modo que los bordes detectados corresponden a los píxeles en donde esta función cambia abruptamente (Sonka et al., 2008). Así, esta técnica se considera la aproximación más común para detectar discontinuidades significativas en los valores de intensidades de una imagen, y esto se logra mediante derivadas de primer y segundo orden (típicamente con el gradiente y el Laplaciano, respectivamente). Esto es, los cambios en la función de intensidad de la imagen ($f(i, j)$) pueden ser determinados con las derivadas y así, los operadores que describen los bordes se expresan mediante derivadas parciales (González et al., 2009; Sonka et al., 1993). Cabe destacar que esta metodología de derivadas solo sirve para funciones de imagen escalares, es decir, sirve por ejemplo para imágenes en escala de grises mas no en imágenes vectoriales como las RGB.

Un detector de bordes tiene como entrada a imágenes discretas y digitalizadas y, produce como salida un mapa de bordes. El mapa de borde de algunos detectores incluye información explícita sobre la posición y la fuerza de los bordes, su orientación y la escala (Ziou et al., 1998).

Definición 2.5.1 (*Detección de bordes*) *Dada una imagen I , se dice que una imagen binaria I_{borde} es una posible solución/salida al problema de detección de bordes si I_{borde} es una imagen digital binaria donde los 1 corresponden al borde y el 0 al fondo.*

Por supuesto, para que una imagen I_{borde} sea una “buena solución” a la tarea de detección de bordes, esta salida debe satisfacer algunas propiedades deseables, entre ellas la más importante es que cuando se encuentre una variación entre dos píxeles vecinos en la imagen, se detecta un borde de la región (o límite) que contiene uno de estos píxeles. Estos límites se pueden representar en la imagen digital dibujando una línea blanca en estos píxeles seleccionados y configurando los otros como fondo negro. En (López-Molina et al., 2013a) se exponen una serie de propiedades de lo que debe ser una buena salida de la tarea de detección de bordes, según los criterios expresados por Canny en (Canny, 1986, 1983), las cuales se mencionan a continuación:

- Tasa de error baja: no deben perderse los bordes que aparecen en la imagen original y no debe haber respuestas espurias.
- Buena localización: la ubicación de los bordes debe estar lo más cerca posible de su posición en la imagen original.
- Unicidad: cada borde de la imagen original debe producir una sola respuesta. Por ello, los bordes deben ser de un píxel de grosor.

Previamente, se mencionó que la manera de detectar un salto de intensidad significativo entre la intensidad de un píxel con la de sus vecinos es mediante la primera y segunda derivada. La primera derivada se determina con el gradiente, el cual es capaz de determinar un cambio en la función de intensidad a través de un vector de dos componentes (magnitud y dirección) que apunta en la dirección de máximo crecimiento de la intensidad de la imagen (Sonka et al., 1993). Así, el gradiente para una función de argumento bidimensional $f(i, j)$ se define como:

$$G = \nabla f \equiv grad(f(i, j)) = [G_i G_j] = \begin{bmatrix} \frac{\partial f}{\partial i} \\ \frac{\partial f}{\partial j} \end{bmatrix} \quad (2.2)$$

La magnitud de este vector sería:

$$|G| = mag(\nabla f) = \sqrt{G_i^2 + G_j^2} = \sqrt{(\partial f / \partial i)^2 + (\partial f / \partial j)^2} \quad (2.3)$$

y la dirección viene dada por el ángulo:

$$\alpha(i, j) = \tan^{-1} \left[\frac{G_j}{G_i} \right] \quad (2.4)$$

Entonces, el valor del gradiente está relacionado con el cambio de intensidad en áreas en donde es variable, y cero en donde es constante.

En cuanto a la segunda derivada, generalmente se calcula utilizando el *Laplaciano*, el cual para una función de dos dimensiones $f(i, j)$ se forma a partir de las derivadas de segundo orden:

$$\nabla^2 f(i, j) = \frac{\partial^2 f(i, j)}{\partial i^2} + \frac{\partial^2 f(i, j)}{\partial j^2}$$

Cabe destacar que el Laplaciano raramente se utiliza directamente debido a su sensibilidad al ruido y a que su magnitud puede generar bordes dobles; además, es incapaz de detectar la dirección del borde, por lo que se utiliza combinándolo con otras técnicas (González et al., 2009).

En este sentido, el objetivo básico de la detección de bordes es encontrar los píxeles de la imagen en donde la función de intensidad o el brillo cambie bruscamente. Esto se puede detectar cuando la primera derivada de la intensidad es mayor en magnitud que un umbral predeterminado, o bien, si la segunda derivada de la intensidad es *zero crossing* (González et al., 2009; Sonka et al., 1993). Esto puede ser determinado a través de la convolución de la imagen con máscaras, que consiste en tomar una matrix 3×3 de números y multiplicar píxel por píxel con una sección o ventana de 3×3 de la imagen. Luego, se suman los productos y el resultado se coloca en el píxel central de la imagen (Phillips, 2000). Un análisis más detallado sobre la detección de bordes se puede obtener en (Marr y Hildreth, 1980).

Existen varios algoritmos que utilizan distintas máscaras y que implementan un operador gradiente, y posteriormente, mediante uno o varios umbrales predeterminados evalúa si se ha encontrado un borde o no (Pavlidis, 1982) como, por ejemplo, el operador Sobel (Sobel, 2014), el operador de Prewitt (Prewitt, 1970), el operador Roberts (Roberts, 1965), el operador Laplaciano Gaussiano (Marr y Hildreth, 1980), Moment-based operator (Reeves et al., 1983), entre otros.

En general, los algoritmos mencionados anteriormente realizan una detección de bordes a través de los siguientes pasos (Bezdek et al., 1998):

Acondicionamiento: en este primer paso se realiza un preprocesamiento a la imagen como suavizado, eliminación de ruidos, entre otros, para limpiar la imagen de forma que se puedan detectar los bordes con mayor facilidad.

Extracción de características: en este paso se obtienen para cada píxel las diferencias espectrales entre sus píxeles adyacentes.

Mezcla/agregación: en este paso se mezclan/agregan las características extraídas de cada píxel en un valor (*edginess*) que representa la variación total del píxel correspondiente.

Escalado: en este último paso se crea la imagen final con los bordes. De acuerdo a la variación total de cada píxel obtenido en la fase anterior, se determina para cada píxel si es un píxel borde o no, con la condición de que los bordes resultantes sean lo más finos posibles.

2.5.1. Operador sobel

El operador Sobel fue presentado por primera vez en 1968 por Irwin Sobel (Sobel, 2014) y luego fue descrito y acreditado formalmente en el libro de (Duda y Hart, 1973). Este algoritmo utiliza un operador diferencial discreto para detectar bordes calculando una aproximación al gradiente de la función de intensidad de una imagen en cada píxel. Para ello, utiliza dos máscaras¹ 3×3 para calcular los cambios horizontales y verticales (González et al., 2009):

$$G_i \Rightarrow \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} \qquad G_j \Rightarrow \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix}$$

En donde la vecindad estaría representada como:

$$\begin{vmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{vmatrix}$$

Por consiguiente, las estimaciones de los gradientes horizontal y vertical se combinan para obtener la magnitud del gradiente mediante la Ecuación 2.3:

$$\begin{aligned} |G| &= \sqrt{G_x^2 + G_y^2} \\ &= \sqrt{[(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)]^2 + [(z_1 + 2z_2 + z_3) - (z_7 + 2z_8 + z_9)]^2} \end{aligned}$$

De esta manera, se identifica a un píxel (i, j) como un píxel borde si en dicho punto $|G| \geq T$, en donde T denota a un umbral predefinido.

En la Figura 2.9, se muestra la salida obtenida con el operador Sobel sobre la Figura 2.1b.

En resumen, esta técnica estima el gradiente de la función de intensidad de una imagen en cada píxel en una vecindad de 3×3 (Sobel, 2014).

2.5.2. Operador canny

El algoritmo de Canny para la detección de bordes es muy popular y consta de múltiples etapas (Canny, 1986) que se describen a continuación:

¹Las máscaras, también conocidas como matriz de convolución o *kernel*, indican los coeficientes a aplicar sobre los píxeles de una área pequeña para calcular el valor del píxel central en función de la ponderación de sus píxeles adyacentes; esto con el fin de realizar desenfoco, nitidez, relieve, detección de bordes, entre otros.

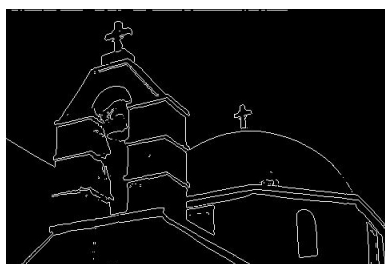


Figura 2.9: Salida del detector de bordes - Sobel

Reducción del ruido: se acondiciona la imagen utilizando un filtro basado en la primera derivada de una Gaussiana conocido como el filtro Gaussiano. El filtro Gaussiano suaviza la imagen gradiente en cada píxel. Un ejemplo de una máscara de 5×5 para realizar el filtro Gaussiano con $\mu = 0$ y $\sigma^2 = 1$ es el siguiente:

$$\frac{1}{273} \begin{vmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{vmatrix}$$

Determinación de la intensidad del gradiente: en cada píxel se determina la magnitud y dirección del gradiente con las Ecuaciones 2.3 y 2.4 respectivamente, de forma que se generan las estimaciones de los gradientes horizontal y vertical.

Supresión no máxima al resultado del gradiente: se afinan los bordes de las estimaciones de los gradientes horizontal y vertical obtenidos en el paso anterior. Esto se realiza asignando el valor 0 al píxel en donde la magnitud del gradiente sea más pequeña que al menos uno de sus dos vecinos en la dirección obtenida en el paso anterior, y se asigna el valor de su magnitud en el caso contrario (ver Rosenfeld y Thurston, 1971).

Selección de los umbrales: se seleccionan dos umbrales, estableciendo el primer umbral más pequeño que el segundo (para más información de cómo obtener estos valores ver Medina-Carnicer et al., 2009).

Histéresis: la histéresis asigna como píxel borde a todo píxel que se encuentre en un conjunto de píxeles con intensidades mayores o iguales al umbral menor conectados entre sí donde al menos uno de ellos tiene una intensidad mayor o igual al umbral mayor. Es decir, todos los píxeles con intensidad mayor o igual al umbral mayor son píxeles borde, mientras que aquellos con intensidad entre los dos umbrales, lo son

solamente si están conectados por un camino de píxeles con estas características a un píxel con intensidad mayor o igual al umbral mayor.

En la Figura 2.10, se observa la imagen resultante de detección de bordes utilizando el operador Canny sobre la Figura 2.1b.

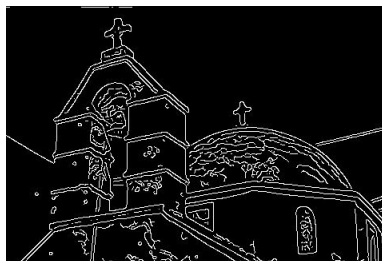


Figura 2.10: Salida del detector de bordes - Canny

2.6. Teoría de conjuntos borrosos

Los conjuntos borrosos se han venido aplicando en diferentes áreas como en la inteligencia artificial, el lenguaje natural, la toma de decisiones, los sistemas expertos, las redes neurales, la teoría de control, entre otras (Bustince et al., 2016).

El concepto matemático de conjuntos borrosos (*fuzzy sets*) se propuso en 1965 por el Prof. L. A. Zadeh en 1965 (Zadeh, 1965), basándose en investigaciones previas de Łukasiewicz. Zadeh introdujo la idea de distribuir los valores de verdad² uniformemente en el intervalo $[0, 1]$ (Bustince et al., 2016). En detalle, los conjuntos borrosos se refieren a conjuntos imprecisos en donde el conjunto en sí, el entorno y su complemento dentro del universo del discurso, no se pueden separar claramente el uno del otro.

Zadeh llegó a la convicción de que se encuentran clases o conjuntos de objetos en el mundo real que no tienen criterios de inclusión definidos con precisión (Seising, 2007). Zadeh (1965) expone: “Por ejemplo, la clase de animales incluye claramente perros, caballos, aves, etc. como miembros, y claramente excluye objetos tales como rocas, flóculos, plantas, etc. Sin embargo, objetos tales como estrellas de mar, bacterias, etc. tienen un estado ambiguo con respecto a la clase de animales”.

En resumen, la teoría de conjuntos borrosos permite introducir el grado en que un elemento x de un conjunto universo X pertenece a un conjunto borroso A (Terano et al., 1992).

²Los valores de verdad o el grado de pertenencia de un elemento a un conjunto en la teoría de conjuntos clásica se mide con un número real comprendido entre los valores $\{0, 1\}$.

Definición 2.6.1 (Conjunto borroso) *Un conjunto borroso A sobre un universo de discurso X es un objeto de la forma: $A = \{(x, \mu_A(x)) | x \in X\}$ donde $\mu_A : X \rightarrow [0, 1]$.*

El valor $A(x)$ se denomina *grado de pertenencia* del elemento x al conjunto borroso A . Luego, una definición equivalente de conjunto borroso en donde se muestra el grado de pertenencia de cada $x \in X$ es la siguiente: sea $A = \{(x, \mu_A(x)) | x \in X\}$ donde $\mu_A : X \rightarrow [0, 1]$ se conoce como la función de pertenencia del conjunto borroso A (Zadeh, 1965).

Estas funciones de pertenencia caracterizan la borrosidad de un conjunto borroso a través de una representación gráfica.

Las descripciones clásicas de los conjuntos borrosos se pueden encontrar en (Zadeh, 1965; Novák et al., 2012; Dubois et al., 2000; Dubois y Prade, 2012; Klir y Yuan, 1995; Kerre, 1993), y una cuenta histórica de los tipos de conjuntos borrosos y sus relaciones se puede encontrar en (Bustince et al., 2016).

Un concepto muy importante dentro de la teoría de conjuntos borrosos es el concepto de α -corte.

Definición 2.6.2 (α -corte) *Sea A un conjunto borroso de X y sea $\alpha \in [0, 1]$. El α -corte de A es simplemente el conjunto de esos $x \in X$ tal que $A(x) \geq \alpha$.*

Una manera de fusionar la información de distintas fuentes es a través de una *función de agregación*. Esto es, al agregar un conjunto de datos se reduce la cardinalidad de dicho conjunto, simplificando la información. El concepto de función de agregación ha sido muy estudiado en la Teoría de Conjuntos Borrosos (Beliakov, 2000; Bustince et al., 2007; Calvo et al., 2002b; Mayor y Torrens, 1988; Montero, 1994; Chiclana et al., 2003; Calvo et al., 2002a).

Definición 2.6.3 (Función de agregación) *Una función de agregación $\mathcal{A} : [0, 1]^n \rightarrow [0, 1]$ para $n \geq 2$, es una función que satisface lo siguiente:*

1. $\mathcal{A}(0, \dots, 0) = 0$, $\mathcal{A}(1, \dots, 1) = 1$.
2. Para todo par de tuplas (x_1, \dots, x_n) y (y_1, \dots, y_n) tales que $x_i \leq y_i$ para todo $i \in \{1, \dots, n\}$, entonces $\mathcal{A}(x_1, \dots, x_n) \leq \mathcal{A}(y_1, \dots, y_n)$.

Las funciones de agregación se pueden clasificar en distintas familias (Beliakov et al., 2007). A continuación, se mencionan dos de ellas con sus ejemplos prototípicos y que serán utilizadas posteriormente en esta memoria:

Mínimo y máximo: son funciones de agregación muy utilizadas en la teoría de conjuntos borrosos. Se definen para cualquier cantidad de argu-

mentos como:

$$\begin{aligned}\min(x) &= \min_{i=1,\dots,n} x_i, \\ \max(x) &= \max_{i=1,\dots,n} x_i,\end{aligned}$$

Media: las medias son promedios de las funciones de agregación. Formalmente, una media es simplemente una función f de la propiedad:

$$\min(x) \leq f(x) \leq \max(x)$$

La *media aritmética* es la función $media(x) = \frac{1}{n}(x_1 + x_2 + \dots + x_n = \frac{1}{n} \sum_{i=1}^n x_i$.

En el contexto de tratamiento de imágenes digitales, la teoría de conjuntos borrosos ha jugado un papel fundamental debido a que ha demostrado ser muy adecuada para abordar la imprecisión que surge naturalmente en el proceso de extraer información de una imagen (Russo, 1998). De esta manera, se han propuesto algoritmos que utilizan la teoría de conjuntos borrosos en todo el proceso o en alguna determinada fase del tratamiento de imágenes (López-Molina, 2012). Por ende, la teoría de conjuntos borrosos en el contexto de imágenes, ha extendido muchas de las técnicas clásicas conocidas a un contexto borroso. Entre ellas se encuentran la clasificación no supervisada borrosa de imágenes (*fuzzy clustering*), específicamente la técnica fuzzy c-means (Dunn, 1973; Bezdek, 1981; Bezdek et al., 1984); y la detección borrosa de bordes (Pal y King, 1983; Liang y Looney, 2003; Russo, 1992; Bustince et al., 2009; Barrenechea et al., 2011; López-Molina et al., 2013b; Bustince et al., 2006b; Senthilkumaran y Rajesh, 2009b).

Capítulo 3

Segmentación borrosa basada en teoría de grafos

Si no me podéis dar poesía, ¿me podéis dar ciencia poética?

Ada Lovelace

RESUMEN: Con el objetivo de definir en este capítulo el nuevo concepto de segmentación borrosa basado en teoría de grafos, es necesario presentar en primer lugar equivalencias al problema de segmentación nítida a través de un conjunto de aristas de un grafo (sección 3.1). Asimismo, es necesario definir formalmente el concepto de segmentación jerárquica tanto por los nodos como por las aristas de un grafo (sección 3.2). Una vez definidos los conceptos de segmentación nítida y jerárquica mediante las aristas de un grafo, en la sección 3.3 se presenta el concepto de segmentación borrosa mediante las aristas de un grafo. Específicamente, la definición formal de segmentación borrosa que se formula en este capítulo, se realiza mediante la relación que posee con la segmentación jerárquica. Además, se proporciona un algoritmo para obtener una segmentación borrosa a partir de cualquier segmentación jerárquica de una imagen. En la sección 3.4 se propone una metodología de visualización a través de grafos para la segmentación nítida, segmentación jerárquica y segmentación borrosa. Por último, se mencionan las relaciones entre algunos problemas de tratamiento de imágenes y, su clasificación de acuerdo al objetivo que consiguen (sección 3.5).

3.1. Nuevo concepto de segmentación grafo teórico

En el capítulo anterior, específicamente en la sección 2.3, se mostró la definición encontrada en la literatura sobre segmentación de imágenes basada en los nodos de un grafo. No obstante, este concepto también puede ser definido a través de las aristas del grafo. De este modo, en esta sección se propone formalmente una definición de segmentación basada en las aristas de un grafo, equivalente a la definición 2.3.1.

Así, la segmentación de imágenes se puede definir a través del conjunto mínimo de *aristas bordes* $B \subset C$ de una red $N(I)$ que separen las distintas regiones de una imagen segmentada.

Observación 3.1.1 *En esta memoria, se denotan aristas como las conexiones que unen o enlazan píxeles adyacentes y, a aristas bordes a aquellas aristas que conectan dos nodos que pertenecen a distintas regiones. Además, se recuerda que los bordes se refieren a los contornos de las regiones u objetos.*

Definición 3.1.1 (Segmentación sobre las aristas de un grafo) *Sea una red de una imagen $N(I) = \{G = (V, E); D\}$, un subconjunto $B \subset E$ caracteriza una segmentación por aristas de la red de una imagen $N(I)$ si y solo si el número de componentes conexos del grafo parcial generado por las aristas $E - B$, denotado como $G(E - B) = (V, E - B)$, decrece cuando se elimina cualquier arista de B .*

El conjunto de todas las segmentaciones por aristas sobre la red $N(I)$ vendrá denotado por $S^e(N(I))$.

En la Figura 3.1b, se puede observar el concepto de segmentación basado en las aristas de la red de la Figura 3.1a, en donde las tres regiones están separadas por las aristas más gruesas.

De este modo, la segmentación de una imagen $N(I)$, es caracterizada mediante el subconjunto de aristas $B \subset C$ que se encuentran entre dos regiones distintas (*aristas bordes*). Además, si una arista es eliminada de B ,

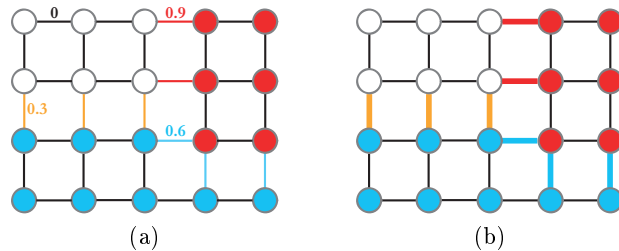


Figura 3.1: (a) Red $N(I)$ y (b) segmentación mediante las aristas de una red $N(I)$

entonces las dos regiones adyacentes correspondientes a los píxeles en los extremos de dicha arista son unidas en una misma región, ya que pasan a estar en la misma componente conexa.

Observación 3.1.2 *Dado un conjunto de aristas bordes B que verifiquen la definición 3.1.1, la familia de los componentes conexos $S = \{R_1, \dots, R_t\}$ del grafo parcial $G(E - B)$ es una segmentación.*

Un aspecto relevante de la definición propuesta de segmentación basada en las aristas de una red (definición 3.1.1), es que posee una biyección con la definición 2.3.1. Es decir, existe una biyección entre el conjunto de segmentaciones $S = \{R_1, \dots, R_t\}$ y el conjunto de aristas $B \subset E$ que producen segmentaciones de imágenes (ver Gómez et al., 2015a). De este modo, se puede decir que es equivalente encontrar una partición de un conjunto de píxeles V en el sentido de la definición 2.3.1, y encontrar el subconjunto de aristas en el sentido de la definición 3.1.1 (Gómez et al., 2015a).

Teorema 3.1.1 *Dada una red de una imagen $N(I) = \{G = (V, E); D\}$, sea el conjunto de todas las segmentaciones basadas en nodos $S^n(N(I))$ (en el sentido de la definición 2.3.1), y sea el conjunto de todas las segmentaciones basadas en aristas $S^e(N(I))$ (en el sentido de la definición 3.1.1). Entonces, existe una biyección $\phi : S^n(N(I)) \rightarrow S^e(N(I))$.*

Prueba.

- Por un lado, dada una segmentación definida mediante su conjunto de aristas bordes $B \in S^e(N(I))$, entonces se define la segmentación $\phi^{-1}(B)$ como la familia de componentes conexos del grafo parcial $G(E - B)$.

En general, dado cualquier subconjunto arbitrario $B \subset E$, los componentes conexos del grafo parcial $G(E - B)$ corresponden a una segmentación en el sentido de la definición 2.3.1. A partir de esta segmentación, se puede obtener el mínimo conjunto de aristas bordes que satisfacen la definición 3.1.1, borrando de B aquellas aristas que enlazan nodos en el mismo componente conexo.

- Por otro lado, dado $\mathcal{S} = \{R_1, \dots, R_t\} \in S^n(N(I))$, entonces el conjunto $\phi(\mathcal{S}) \in S^e(N(I))$ se construye de la siguiente manera:

$$\phi(\mathcal{S}) = E - \bigcup_{h=1}^t \{e = \{a, b\} \in E \mid a, b \in R_h\}$$

Por definición, los componentes conexos de:

$$G(E - \phi(\mathcal{S})) = G\left(\bigcup_{h=1}^t \{e = \{a, b\} \in E \mid a, b \in R_h\}\right)$$

son los subconjuntos $R_h \in \mathcal{S}$, siendo t el número de estos componentes conexos.

Sea $e = \{a, b\} \in \phi(\mathcal{S})$ una arista a ser eliminada de $\phi(\mathcal{S})$; entonces existen dos conjuntos distintos $R_h, R_{h'} \in \mathcal{S}$ de forma que $a \in R_h$ y $b \in R_{h'}$, y el grafo parcial

$$G(E - \phi(\mathcal{S}) - \{e\}) = G\left(\bigcup_{h=1}^t \{e = \{a, b\} \in E \mid p, q \in R_h\} \cup \{e\}\right)$$

tiene exactamente $t - 1$ componentes conexos (los conjuntos R_h y $R_{h'}$ están entonces en el mismo componente conexo). Por lo tanto, $\phi(\mathcal{S}) \in \mathcal{S}^e(N(I))$.

Por lo tanto, si un subconjunto de aristas $B \subset E$ caracteriza una segmentación de imagen de $N(I)$, entonces cualquiera de sus aristas enlaza dos regiones diferentes de la segmentación. Además, si se elimina una arista de B , las dos regiones adyacentes de sus extremos quedan unidas en una región. Luego, las condiciones necesarias y suficientes para que se de esta propiedad (i.e., una arista de B que enlace dos nodos de regiones diferentes) se expresan en el siguiente lema. Además, en este lema se introduce la estructura de *bosque soporte* en una red de imágenes, un grafo que no posee ciclos, el cual será útil para entender posteriormente el algoritmo de construcción de una segmentación borrosa de imágenes. \square

Lema 3.1.1 *Dada la red de una imagen $N(I) = \{G = (V, E); D\}$, un subconjunto de aristas $B \subset E$ es el borde de una segmentación si y sólo si existe un bosque soporte $G(F) = (V, F)$ de G , y existe un subconjunto de aristas $F' \subset F$ que verifican lo siguiente:*

1. $F' \subset B$
2. $(F - F') \not\subset B$
3. $\forall e = \{a, b\} \in E - F$, entonces $e \in B$ si y sólo si a y b pertenecen a diferentes componentes conexos del grafo parcial $G(F - F')$.

Prueba.

(\Rightarrow) Dado el borde $B \subset E$ de una segmentación $S = \{R_1, R_2, \dots, R_t\}$, en donde cualquiera de los conjuntos R_h es un componente conexo del grafo parcial $G(E - B)$, sean $\{F_1, F_2, \dots, F_t\}$ árboles soportes arbitrarios de estos componentes conexos. Sean $F = \bigcup_{h=1}^t F_h$ la inicialización del conjunto $F \subset E$.

Sea $F' \subset B$ el conjunto maximal (en el sentido de la inclusión) de esas aristas en B tales que $F \cup F'$ no incluyan un ciclo. El conjunto $F = F \cup F'$ se actualiza, y se cumplen las Propiedades 1 y 2 de la definición 2.3.1. Para

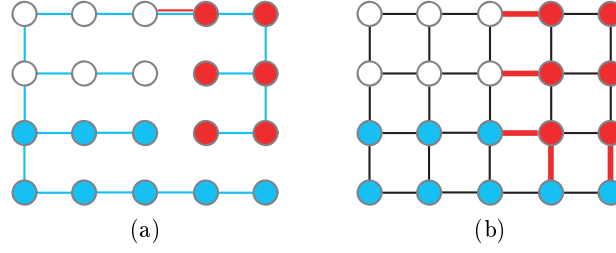


Figura 3.2: De la red de la Figura 3.1a: (a) bosque soporte F y el subconjunto de aristas $F' \subset F$; y (b) el subconjunto B derivado por esos subconjuntos

ver la Propiedad 3, entonces sea $e = \{a, b\} \in E - F$. Dos casos exhaustivos y exclusivos son posibles:

- $a, b \in R_h$ para algún $h \in \{1, 2, \dots, t\}$. En este caso, $e \notin B$.
- $a \in R_h$ y $b \in R_{h'}$ para algún $h \neq h'$ en $\{1, 2, \dots, t\}$. En este caso, $e \in B$ (de lo contrario, si $e \notin B$, R_h y $R_{h'}$ pertenecerán al mismo componente conexo de $G(E - B)$).

(\Leftarrow) Sea $F \subset E$ el subconjunto de aristas tales que el grafo parcial $G(F) = (V, F)$ define un bosque soporte de G . Sea $F' \subset F$ un subconjunto arbitrario, pudiendo ser el conjunto vacío; el conjunto B se inicializa como $B = F'$. Por la definición de un bosque soporte, el grafo parcial $G(F - F')$ define una partición $S = \{R_1, R_2, \dots, R_t\} \subset V$ de tal manera que los subgrafos inducidos por cualquiera de estos t subconjuntos, es un grafo conexo.

Entonces, el conjunto B se completa con todas las aristas $e = \{a, b\} \in E - F$ que enlazan los distintos subconjuntos de la partición, i.e. $a \in R_h$ y $b \in R_{h'}$ con $h \neq h'$. De esta manera, el conjunto B construido es $B = B(S)$, i.e., el conjunto mínimo de aristas bordes para la segmentación S introducida en la definición 3.1.1.

En relación con el lema 3.1.1 y continuando con la red de la Figura 3.1a del ejemplo 2.2.1, la Figura 3.2 muestra en (a) la relación que existe entre el bosque soporte $G = (V, F)$ (aristas coloreadas en azul) y un subconjunto seleccionado de aristas $F' \subset F$ (aristas coloreadas en rojo) que tuvieran un determinado peso (en este ejemplo $\alpha = 0.9$); en (b) se muestra el conjunto final de aristas B (de color rojo grueso) obtenidas a través de esta selección. Cabe destacar que al quitar la arista en rojo de la Figura 3.2, se separa una región que contiene nodos de dos distintos colores (azul y blanco), esto porque el peso de las aristas entre nodos de color azul y blanco es menor al establecido (i.e. $\alpha > 0.3$) para realizar la segmentación. \square

3.2. Concepto de segmentación jerárquica de imágenes

Se puede decir que la segmentación jerárquica de imágenes es una extensión de la segmentación de imágenes. Es decir, aunque estas dos técnicas generen distintos resultados se basan en el mismo principio, la segmentación de imágenes produce una imagen particionada en regiones no solapadas y la segmentación jerárquica de imágenes produce una secuencia jerárquica de imágenes particionadas en regiones. Entre algunos algoritmos de segmentación jerárquica están (Arbeláez et al., 2011; Lhermitte et al., 2008; Schroeter y Bigün, 1995; Cheng y Sun, 2000; Chamorro-Martínez et al., 2003), entre otros.

Como se mencionó en la sección 2.4, la principal ventaja de los algoritmos de segmentación jerárquica es que pueden mostrar una evolución de cómo se unen los grupos (en métodos aglomerativos) o división (en métodos divisivos), desde el comienzo del proceso hasta el paso final. Aunque los métodos de agrupamiento jerárquico pueden traer más problemas computacionales, tienen una ventaja importante porque los resultados son más informativos que los dados por algoritmos de agrupamiento particiones o no jerárquicos, que solo proporcionan una imagen final del proceso (Gómez et al., 2015b).

De este modo, una vez propuesta la definición de segmentación de imagen a través de las aristas de un grafo (definición 3.1.1), en este apartado se formula una definición de segmentación jerárquica de imágenes sobre un grafo. Para ello, primero es necesario definir un orden parcial entre las segmentaciones sobre una red siguiendo la definición 2.4.1.

Definición 3.2.1 (Relación de orden entre segmentaciones de un grafo) Dada la red de una imagen $N(I) = \{G = (V, E); D\}$, sean dos segmentaciones $\mathcal{S}, \mathcal{S}' \in \mathcal{S}^n(N(I))$, se dice que \mathcal{S} es más fina que \mathcal{S}' (y se denota como $\mathcal{S} \subseteq \mathcal{S}'$) si para todo $R \in \mathcal{S}$, entonces existe $R' \in \mathcal{S}'$, tal que $R \subseteq R'$.

La segmentación más fina será $\mathcal{S}^0 = \{\{v_1\}; \{v_2\}; \dots\}$, i.e. cada nodo de la red de la imagen, es un elemento de la partición; por el otro lado, la segmentación más gruesa será $\mathcal{S}^K = \{V\}$, la cual es una partición con un único elemento que contiene todos los nodos. Así, entre estas dos segmentaciones extremas, es posible construir diversas familias inclusivas de segmentaciones, ordenadas desde la partición más fina \mathcal{S}^0 hasta la partición \mathcal{S}^K con todos los nodos.

Luego, una vez establecido el orden parcial entre dos segmentaciones, se introduce una noción operativa de segmentación jerárquica grafo teórico.

Definición 3.2.2 (Segmentación jerárquica sobre los nodos de un grafo) Dada la red de una imagen $N(I) = \{G = (V, E); D\}$, se dice que la familia de segmentaciones $\mathcal{S} = (\mathcal{S}^0, \mathcal{S}^1, \mathcal{S}^2, \dots, \mathcal{S}^K)$ de la imagen $N(I)$,

es una segmentación jerárquica de $N(I)$ cuando se cumplen las siguientes propiedades:

1. $\mathcal{S}^t \in \mathcal{S}^n(N(I))$ para todo $t \in \{0, 1, \dots, K\}$, i.e. cada \mathcal{S}^t es una segmentación por nodos de $N(I)$.
2. Existen dos particiones triviales: la primera ($\mathcal{S}^0 = \{\{v\}, v \in V\}$), que contiene todos los píxeles como singletons, y la última, ($\mathcal{S}^K = \{V\}$) con un clúster para todos los píxeles.
3. $|\mathcal{S}^t| > |\mathcal{S}^{t+1}|$ para todo $t = 0, 1, \dots, K - 1$ (i.e., en cada iteración disminuye el número de regiones en la imagen).
4. $\mathcal{S}^t \subseteq \mathcal{S}^{t+1}$ para todo $t = 0, 1, \dots, K - 1$.

En la Figura 3.3, se puede observar el concepto de segmentación basado en las aristas de la red de la Figura 3.1a.

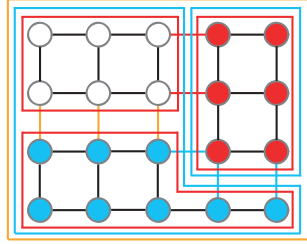


Figura 3.3: Segmentación jerárquica por nodos

Con la definición de segmentación por aristas, es posible establecer una definición de segmentación jerárquica por aristas, teniendo en cuenta que dada dos segmentaciones por aristas $B, B' \in \mathcal{S}^e(N(I))$, B es más fina que B' si $B \supset B'$.

Definición 3.2.3 (Segmentación jerárquica sobre las aristas de un grafo) Dada la red de una imagen $N(I) = \{G = (V, E); D\}$, se dice que la secuencia $\mathcal{B} = \{B^0, \dots, B^K\}$, con $B^t \subset E$ para todo $t \in \{0, 1, \dots, K\}$ es una segmentación jerárquica de $N(I)$ si y solo si se cumple lo siguiente:

- $B^t \in \mathcal{S}^e(N(I))$ para todo $t \in \{0, 1, \dots, K\}$, i.e. cada B^t es una segmentación jerárquica de $N(I)$.
- La secuencia $\{\phi^{-1}(B^0), \dots, \phi^{-1}(B^K)\}$ es una segmentación jerárquica de $N(I)$.

En consecuencia, se tiene que $B^0 = E \supset B^1 \supset \dots \supset B^K = \emptyset$.

En la Figura 3.4, se puede observar el concepto de segmentación basado en las aristas de la red de la Figura 3.1a.

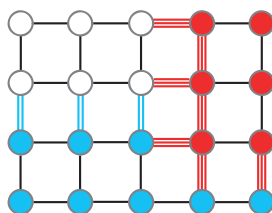


Figura 3.4: Segmentación jerárquica por aristas

En resumen, las segmentaciones jerárquicas mostradas en las Figuras 3.3 y 3.4, consideran en su primera partición todos los nodos en una misma región, la siguiente partición identifica las regiones que están más separadas de sus regiones adyacentes (y por tanto se identifican las regiones más claras) y, finalmente se detectan aquellas regiones y/o fronteras cuya disimilitud es más baja. En este sentido, una jerarquía permite ir identificando de manera dinámica las regiones de una imagen de las más claras a las menos obvias.

3.3. Concepto de segmentación borrosa de imágenes

En esta sección se propone la definición del concepto de segmentación borrosa de imágenes como resultado de la extensión de la segmentación nítida a un contexto borroso. Esta extensión se plantea de forma similar a la ocurrida con la extensión de la teoría de conjuntos clásica a un contexto borroso, en donde en primeras investigaciones se consideraron más valores de verdad que la clásica dupla binaria asimilada al conjunto $\{0, 1\}$ y, luego en investigaciones posteriores se llegaron a considerar tantos valores de verdad como valores posibles dentro del intervalo $[0, 1]$. De este modo, se propone dar una secuencia natural a la extensión de la definición de segmentación nítida a la definición de segmentación borrosa de imágenes, usando como paso intermedio la definición de la segmentación jerárquica.

Además, como se mencionó en la sección 2.3, algunas imágenes digitales no son muy nítidas y por ende, al realizar una segmentación en estas imágenes, ésta llega a ser más complicada. Muchos son los casos en donde las imágenes tratadas pueden tener ruido o tener poco marcado el contorno de los objetos. Las imágenes de las Figuras 3.5a y 3.5b tienen los contornos bien marcados, es decir, cada objeto de la imagen está bien diferenciado de los objetos de su alrededor. Contrariamente, en la imagen de la Figura 3.5c, se observa que el objeto principal (i.e. el gorila) está bien diferenciado del fondo, aunque algunas siluetas que aparecen en el fondo no están tan nítidas o están poco marcadas. Por último, en la imagen de la Figura 3.5d, se aprecia con poca diferencia los objetos de la imagen debido a que los colores de ellos son muy parecidos. En general, cuando una imagen no tiene los bordes de los

objetos tan definidos o los colores de los objetos son muy parecidos, es difícil demilitar el contorno de cada objeto en la imagen resultante al realizar una segmentación.

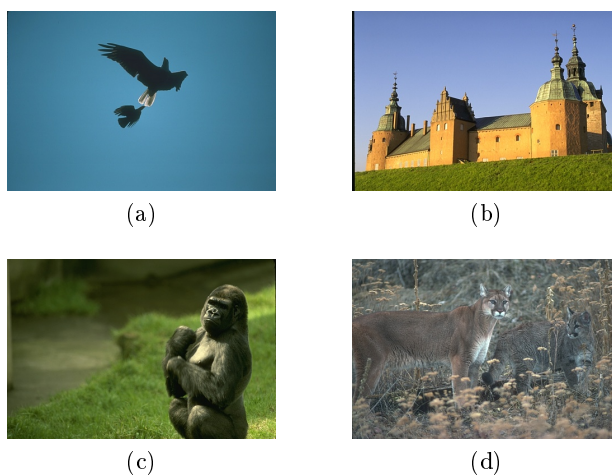


Figura 3.5: Imágenes de la base de datos de Berkeley (BSDS500) (Martin et al., 2001)

Al recordar la definición de segmentación de imágenes presentada en la sección 2.3, se dice que una segmentación es un proceso que divide o segmenta una imagen en múltiples regiones u objetos. De este modo, cuando la imagen a segmentar posee regiones muy diferenciadas, las aristas que están entre dichas regiones son aristas borde, por lo que una arista toma valores exhaustivos, es decir es borde o no-borde $(i, j) = \{0, 1\}$. Sin embargo, cuando los objetos de la imagen a tratar no poseen bordes tan marcados, es difícil realizar una segmentación por aristas en la imagen debido a que la asignación de una arista como borde no es exhaustiva debido a que alguna o todas las aristas pueden ser en cierto grado una arista borde. Así, la teoría de conjuntos borrosos presentada en la sección 2.6 facilita la asignación de las aristas a aristas bordes proporcionando una gradualidad en la clasificación de la arista como borde dentro de un intervalo $(i, j) = [0, 1]$.

En los últimos años, los métodos borrosos han atraído un enorme interés. Estos métodos han demostrado ser una herramienta poderosa para el modelado y control de procesos no fácilmente tratables con los enfoques convencionales (Casillas et al., 2013). En este sentido, las técnicas de tratamiento de imágenes con un enfoque borroso pueden lidiar con bordes entre regiones que sean imprecisos (Chamorro-Martínez et al., 2003). Pocos estudios se han dedicado a la extensión del concepto de segmentación nítida a un contexto borroso (ver por ejemplo Bloch, 1993; Rosenfeld, 1979; Saha et al., 2000; Udupa y Samarasekera, 1996). No obstante, es importante resaltar que muchas técnicas borrosas se han utilizado para afrontar estos ruidos en

las imágenes, pero las salidas de estas técnicas siguen siendo nítidas.

Una breve introducción de lo que se presentará en la siguiente sección y que está en consonancia con el párrafo anterior es, que la salida nítida de una segmentación puede ser visualizada de dos maneras. La primera de ellas consiste en que los objetos segmentados de la imagen de la Figura 3.6a, son coloreados en un color resultante de una agregación de los colores de los píxeles adyacentes (ver Figura 3.6b); y una segunda manera, es cuando la segmentación es caracterizada en términos de un conjunto de bordes que separan las regiones de forma binaria, en donde los píxeles negros representan los píxeles de una región y los píxeles blancos representan los píxeles entre dos regiones adyacentes manteniendo una misma intensidad (ver Figura 3.6c). En cambio, una salida es borrosa si al caracterizar una segmentación por medio de un conjunto de bordes, estos bordes presentan gradaciones en su intensidad de forma que sea más similar a la segmentación que haría un humano al resaltar con mayor intensidad aquellos bordes que ve con mayor nitidez, o bien, más similar a una imagen que agrega segmentaciones nítidas obtenidas por un conjunto de humanos en donde un borde de esta imagen con una intensidad baja corresponde a un borde que ha sido identificado por un solo experto, mientras que el resto de expertos no lo ha identificado (ver Figura 3.6d). Mayor formalismo en cuanto a estos tipos de visualización, serán dados en secciones posteriores.

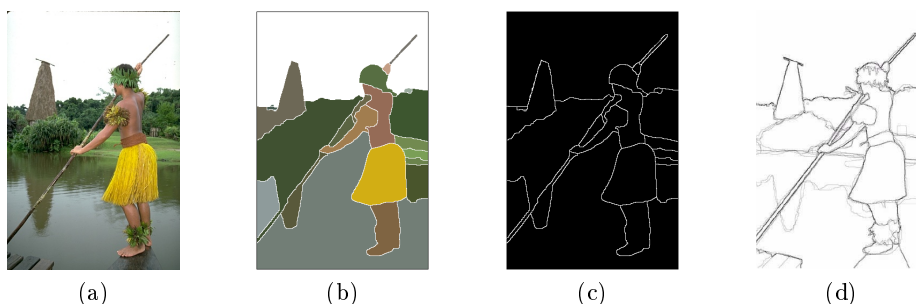


Figura 3.6: Segmentación realizada a una imagen de la base de datos de Berkeley (BSDS500) (Martin et al., 2001)

De acuerdo a Josef Albers, la única forma de ver un color es en relación con su entorno (Albers, 2013). El humano al visualizar una imagen digital, observa en primer lugar aquellos objetos que tengan más contraste con los objetos del alrededor. En la Figura 3.7 se muestra una escenificación de ello. El humano ve primero la flor que es el objeto que más contraste tiene con los objetos del alrededor, luego ve las hojas, seguidamente la silueta iluminada del jarrón, posteriormente el jarrón en sí y por último el fondo.

Por ello, la salida de una segmentación debe presentar gradaciones de color en los objetos o bordes de los objetos dependiendo de qué tan nítido o

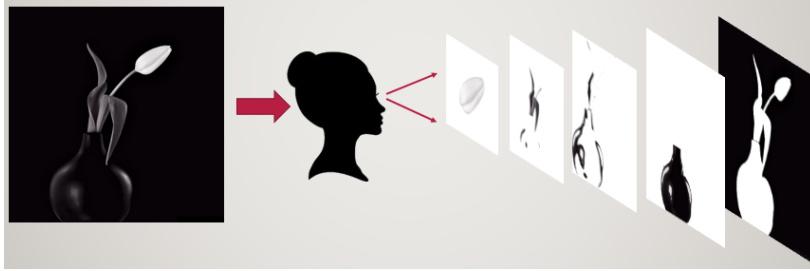


Figura 3.7: Ejemplo del orden en cómo ve un humano los objetos de una imagen

marcado sea el contorno de los mismos en la imagen original. De este modo, la salida será más realista e informativa. Estas gradaciones se pueden conseguir estableciendo α -cortes (ver definición 2.6.2) en la intensidad de cada región o borde de la región.

En resumen, en esta memoria se plantea una manera de definir el concepto de segmentación borrosa de imágenes como una extensión de la definición 3.1.1 de segmentación de imágenes basada en aristas (Gómez et al., 2014, 2015a). Esta propuesta se basa en el hecho de que una segmentación nítida puede ser caracterizada por medio de un conjunto de aristas que separan las regiones adyacentes de la segmentación a las que se les puede denominar como *borde borroso* de la imagen. Este concepto de borde borroso o frontera borrosa ha sido introducido en trabajos previos de (Bustince et al., 2006b, 2009; Senthilkumaran y Rajesh, 2009a). Es importante decir que, también se puede formalizar el concepto de segmentación borrosa basado en nodos, pero actualmente está siendo estudiado para ser presentado en trabajos futuros.

Entonces, a partir de la definición 3.1.1 se propone definir el concepto de segmentación borrosa de imágenes, esto es, a través de la fuzzificación del concepto de segmentación basado en aristas (Gómez et al., 2014, 2015a).

Definición 3.3.1 (*Segmentación borrosa de imágenes basado en aristas*) Dada una red $N(I) = \{G = (V, E); D\}$, se dice que el conjunto borroso $\tilde{B} = \{(e, \mu_B(e)), e \in E\}$ genera una segmentación borrosa de imágenes basado en aristas si y solo si para todo $\alpha \in [0, 1]$ el conjunto nítido $B(\alpha) = \{e \in E : \mu_B(e) \geq \alpha\}$ produce una segmentación de imágenes en el sentido de la definición 3.1.1.

Como se puede apreciar en la definición anterior, la función de pertenencia del conjunto borroso \tilde{B} para una arista dada, representa el grado de separación entre dos píxeles vecinos en el proceso de segmentación. En resumen, se propone utilizar *bordes borrosos* para representar las disimilitudes entre los píxeles.

Observación 3.3.1 Una segmentación borrosa \tilde{B} es una segmentación por

aristas B en el sentido nítido de la definición 3.1.1 si se cumple: $\tilde{B} = B \iff \mu_B(e) = 1 \forall e \in B$ (i.e., una arista e está en B si tiene grado 1, no está en B si tiene grado 0, y no hay aristas con grados intermedios).

El Ejemplo 2.2.1 presentado en la Subsección 2.2.1 que utiliza la red de la Figura 3.8a, se puede vincular a una segmentación borrosa \tilde{B} basado en aristas como se observa en la Figura 3.8b. En esta última imagen, se asume que las aristas e finas (coloreadas en negro) tienen $\mu_B(e) = 0$, las líneas azules más gruesas tienen un grado algo mayor, por ejemplo $\mu_B(e) = 0.3$ y las líneas rojas más gruesas tienen un grado aún mayor, por ejemplo $\mu_B(e) = 0.6$.

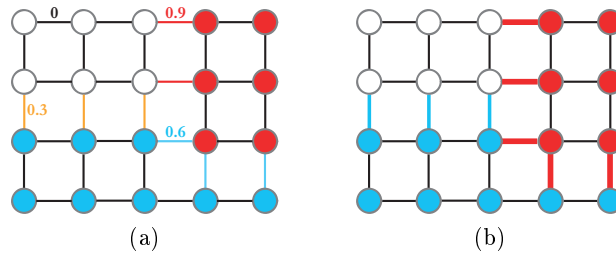


Figura 3.8: (a) Red $N(I)$ y (b) segmentación borrosa \tilde{B} del ejemplo 2.2.1

Dos segmentaciones nítidas de esta segmentación borrosa \tilde{B} vendrían dadas por $B(0.3)$ y $B(0.6)$, las cuales se pueden observar en la Figura 3.9. En realidad, existen cuatro segmentaciones diferentes asociadas a los siguientes intervalos:

$$\alpha \in \{[0, 0]; (0, 0.3]; (0.3, 0.6]; (0.6, 1.0]\}$$

las cuales están asociadas con la segmentación trivial $B^0 = E$; $B^1 = B(0.3)$; $B^2 = B(0.6)$ y $B^3 = \emptyset$ respectivamente.

Observación 3.3.2 En este apartado se ha definido el concepto de segmentación borrosa de una imagen basada en la clase de borde borroso sobre el conjunto de aristas de la red de la imagen. En este sentido, la función de

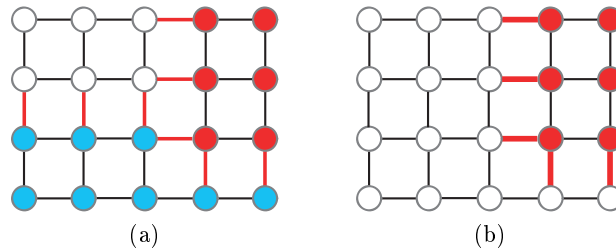


Figura 3.9: Segmentaciones nítidas $B(0.3)$ y $B(0.6)$ de la segmentación borrosa \tilde{B} del Ejemplo 2.2.1

pertenencia de esta clase representa, para una arista particular, el grado de separación entre dos píxeles adyacentes en el proceso de segmentación.

Es importante distinguir entre este concepto de borde borroso y el concepto clásico de disimilitud entre dos píxeles adyacentes. Es decir, aunque estos dos conceptos estén relacionados, no son equivalentes, debido a que las medidas de disimilitud entre dos píxeles adyacentes se obtiene a partir de la información espectral de pares de píxeles (o nodos), sin considerar el proceso de segmentación. Esto se puede observar en los métodos de detección de borde o detección de borde borroso (ver López-Molina et al., 2010, para más detalles), en donde la medida asociada a cada arista representa la disimilitud (en términos de información espectral) entre los píxeles en los extremos de esa arista, que no necesariamente cumple la idea de borde borroso entre regiones. Por este motivo, no es posible la construcción directa de una segmentación borrosa de imágenes basada en la medida de disimilitud (Gómez et al., 2015a); un ejemplo de ello se plantea a continuación.

Ejemplo 3.3.1 *Sea \tilde{D} una clase borrosa sobre el conjunto de aristas E basada en las disimilitudes iniciales mostradas en la red de Figura 3.8a. La función de pertenencia $\mu_{\tilde{D}} : E \rightarrow [0, 1]$ de esta clase borrosa, coincide con el grado de disimilitud entre dos píxeles adyacentes. Entonces, $\mu_{\tilde{D}}(e) = d(e)$, toma los valores 0, 0.3, 0.6 y 0.9. Luego, es fácil observar que esta clase borrosa no produce una segmentación borrosa de imágenes.*

- *Para $\alpha > 0.9$, $\tilde{D}(\alpha) = \{e \in E / \mu_{\tilde{D}}(e) \geq \alpha\} = \{\emptyset\}$. La segmentación de imagen producida por este conjunto de aristas es toda la figura $S = \{(i, j), i = 1, \dots, 4; j = 1, \dots, 5\}$.*
- *Para $0.6 < \alpha \leq 0.9$, $\tilde{D}(\alpha) = \{(1, 3), (1, 4)\} \cup \{(2, 3), (2, 4)\}$, i.e. sólo las aristas rojas), las cuales no producen una segmentación de imagen en el sentido de la definición 3.1.1.*
- *Para $0.3 < \alpha \leq 0.6$, $\tilde{D}(\alpha) = \{(1, 3), (1, 4)\} \cup \{(2, 3), (2, 4)\} \cup \{(3, 3), (3, 4)\} \cup \{(3, 4), (4, 4)\} \cup \{(3, 5), (4, 5)\}$ (i.e. las aristas azules y rojas de la Figura 3.8a). La segmentación producida por este conjunto de aristas es $S = \{R_1, R_2\}$, en donde R_1 es el conjunto de nodos blancos y azules, y R_2 se refiere al conjunto de nodos rojos de la Figura 3.8a.*
- *Para $0 < \alpha \leq 0.3$, $\tilde{D}(\alpha) = E$. La segmentación de imagen producida por este conjunto de aristas es $S = \{(i, j) \mid \forall i, j\}$.*

Dado que el conjunto nítido $\tilde{D}(\alpha)$ no es un conjunto de aristas que induce una segmentación de imagen para $\alpha \in (0.6, 0.9]$, se puede concluir que \tilde{D} no es una segmentación borrosa de imagen.

En consecuencia, debido a que usualmente la medida de distancia o disimilitud entre píxeles adyacentes no es suficiente para construir una segmentación borrosa de imágenes, hace que este problema no sea una tarea trivial. No obstante, existe una relación fuerte entre el problema de la segmentación borrosa de imágenes y la segmentación jerárquica de imágenes, y por ende, sería posible generar una solución de segmentación borrosa de imágenes a través de una segmentación jerárquica de imágenes, y viceversa.

3.3.1. Algoritmo para construir una segmentación borrosa basado en una segmentación jerárquica

Como se mencionó en la sección anterior, es posible construir una segmentación borrosa a partir de una segmentación jerárquica de imágenes debido a que existe una biyección entre dichos conceptos (Gómez et al., 2015a). De este modo, en esta Subsección se presenta un algoritmo que construye una segmentación borrosa a partir de una segmentación jerárquica.

Teorema 3.3.1 *Dada la red de una imagen $N(I) = \{G = (V, E); D\}$, si $\tilde{B} = \{(e, \mu_B(e)), e \in E\}$ es una segmentación borrosa de imágenes, entonces para cualquier secuencia $\alpha_0 = 0 < \alpha_1 < \dots < \alpha_{K-1} = 1 < \alpha_K$, el conjunto $\{B(\alpha_0) = E, B(\alpha_1), \dots, B(\alpha_{K-1}), B(\alpha_K) = \emptyset\}$ produce una segmentación jerárquica de imágenes.*

Prueba. Dada una secuencia $\alpha_0 = 0 < \alpha_1 < \dots < \alpha_{K-1} = 1 < \alpha_K$, se denota por $\tilde{S} = \{S^0 = \phi^{-1}(B(\alpha_0)), \dots, S^K = \phi^{-1}(B(\alpha_K))\}$ a la secuencia de segmentaciones inducidas por el conjunto $\{B(\alpha_0) = E, \dots, B(\alpha_K) = \emptyset\}$. Con el fin de probar que \tilde{S} es una segmentación jerárquica de $N(I)$, se deben verificar las cuatro propiedades exigidas en la definición 3.2.2.

A partir de la definición de segmentación borrosa de imágenes, es claro que la Propiedad 1 se cumple, ya que se requiere que para todos $\alpha \in (0, 1)$, el conjunto de aristas $B(\alpha)$ produzca una segmentación de imagen. También se cumple la Propiedad 2, debido a que el conjunto de aristas $B(0) = E$ produce la partición trivial en el que todos los píxeles pertenecen a un clúster *singleton* (i.e., $R_0 = \{\{v\}, v \in V\}$) y el conjunto \emptyset producen una partición con un único clúster que contiene todos los píxeles. En cuanto a las Propiedades 3 y 4, estas también se cumplen porque para cualquier iteración t , las segmentaciones $B(\alpha_t)$ y $B(\alpha_{t+1})$ generadas por las particiones S^t y S^{t+1} satisfacen que $B(\alpha_t) \supset B(\alpha_{t+1})$ (Gómez et al., 2015a). \square

Teorema 3.3.2 *Dada la red de una imagen $N(I) = \{G = (V, E); D\}$, sea $\mathcal{B} = \{B^0 = E, B^1, \dots, B^K = \emptyset\}$ una segmentación jerárquica de imágenes, y para todo $t \in \{0, 1, \dots, K\}$. Sea $\mu^t : E \rightarrow \{0, 1\}$ la función de pertenencia asociada al conjunto de bordes $B^t \subset E$. Entonces, el conjunto borroso \tilde{B}*

definido como:

$$\mu_B(e) = \sum_{t=0}^K w_t \mu^t(e) \quad \forall e \in E$$

induce una segmentación borrosa de imágenes de $N(I)$ para cualquier secuencia de pesos $w = (w_0, w_1, \dots, w_K)$ tal que

$$w_t \geq 0 \quad \forall t \in \{0, 1, \dots, K\} \quad \text{y} \quad \sum_{t=0}^K w_t = 1$$

Prueba. Con el fin de probar que el conjunto borroso \tilde{B} produce una segmentación borrosa de imágenes, sólo se debe verificar que cualquier α -corte sobre el conjunto borroso \tilde{B} produce una segmentación de imagen.

Se puede apreciar que las segmentaciones más simples son directas, con $\alpha = 0$ y $\alpha = 1$.

Sea $\alpha \in (0, 1)$ el valor en el intervalo $(0, 1)$, y sea $B(\alpha) \subset E$ el subconjunto de aristas con $\mu_B(e) \geq \alpha$.

Sea $h_0 \in \{0, 1, \dots, K\}$ el primer nivel jerárquico tal que $\sum_{t=0}^{h_0} w_t \geq \alpha$, se ve entonces que $B(\alpha) = B^{h_0}$.

- $B(\alpha) \subset B^{h_0}$

Si se toma en cuenta que \mathcal{B} es una segmentación jerárquica de una imagen $N(I)$, se cumple lo siguiente: $B^0 \supset B^1 \dots \supset B^{h_0} \supset \dots \supset B^K$.

Sea $e \in B(\alpha)$ una arista arbitraria, entonces $\mu_B(e) = \sum_{t=0}^K w_t \mu^t(e) \geq \alpha > 0$. Dado que $\mu_B(e) > 0$ existe al menos un nivel jerárquico $i \in \{0, 1, \dots, K\}$ tal que $e \in B^i$; sea i_e el mayor de estos niveles jerárquicos con esta propiedad, i.e., $e \in B^{i_e}$ y $e \notin B^{i_e+1}$.

Por definición de i_e y considerando la propiedad jerárquica de \mathcal{B} , entonces:

$$\begin{aligned} \alpha \leq \mu_B(e) &= \sum_{t=0}^K w_t \mu^t(e) \\ &= \sum_{t=0}^{i_e} w_t \mu^t(e) + \sum_{t=i_e+1}^K w_t \mu^t(e) \\ &= \sum_{t=0}^{i_e} w_t 1 + \sum_{t=i_e+1}^K w_t 0 \end{aligned}$$

Por tanto, $\sum_{t=0}^{i_e} w_t \geq \alpha$, y tomando en cuenta que h_0 es el primer nivel que satisface $\sum_{t=0}^{h_0} w_t \geq \alpha$, se tiene que $i_e \geq h_0$. Además, puesto que \mathcal{B} es jerárquico, entonces $B^{h_0} \supset B^{i_e}$.

Luego, $e \in B^{i_e} \subset B^{h_0}$ y está demostrado que $B(\alpha) \subset B^{h_0}$.

- $B^{h_0} \subset B(\alpha)$

Ahora, sea $e \in B^{h_0}$ una arista genérica; dado que \mathcal{B} es jerárquica, $e \in B^t$ para cualquier $t \leq h_0$; de este modo:

$$\begin{aligned} \mu_B(e) &= \sum_{t=0}^K w_t \mu^t(e) \\ &= \sum_{t=0}^{h_0} w_t \mu^t(e) + \sum_{t=h_0+1}^K w_t \mu^t(e) \geq \sum_{t=0}^{h_0} w_t 1 + \sum_{t=h_0+1}^K w_t \mu^t(e) \geq \alpha \end{aligned}$$

Entonces, $e \in B(\alpha)$ y queda demostrado que $B^{h_0} \subset B(\alpha)$.

Por lo tanto, para cualquier valor α existe un $t \in \{0, \dots, K\}$ tal que $B(\alpha) = B^t$. Y el resultado se consigue, dado que B^t produce una segmentación nítida de imagen por definición. \square

Observación 3.3.3 *Obsérvese que el teorema anterior muestra una manera de construir una familia (que depende de los pesos) de las segmentaciones borrosas a partir de una segmentación jerárquica de imágenes.*

Ejemplo 3.3.2 *A partir de la segmentación jerárquica $\mathcal{B} = \{B^0, B^1, B^2, B^3\}$ mostrada en la Figura 3.10b correspondiente a la red de la Figura 3.10a, se construye la segmentación borrosa mostrada en la Figura 3.10c con los siguientes pesos: ($w_0 = 0.4; w_1 = 0.3; w_2 = 0.3, w_3 = 0$).*

3.4. Metodologías de visualización

Una vez presentados los conceptos de segmentación nítida, segmentación jerárquica y segmentación borrosa, en esta sección se muestra la metodologías de visualización correspondiente a cada una de esas tareas.

3.4.1. Metodología de visualización de una segmentación nítida

Generalmente, una metodología muy común de visualizar una segmentación de imágenes es a través de un mapa de regiones, es decir, una imagen similar a la original en la que los píxeles que pertenecen a una misma región se colorean en un color que se obtiene como una agregación de los colores de los píxeles que pertenecen a dicha región, tal y como se puede ver en la Figura 3.11, en donde las regiones de la imagen segmentada se pueden observar fácilmente.

No obstante, esta forma de visualización presenta una serie de inconvenientes, las cuales se mencionan a continuación:

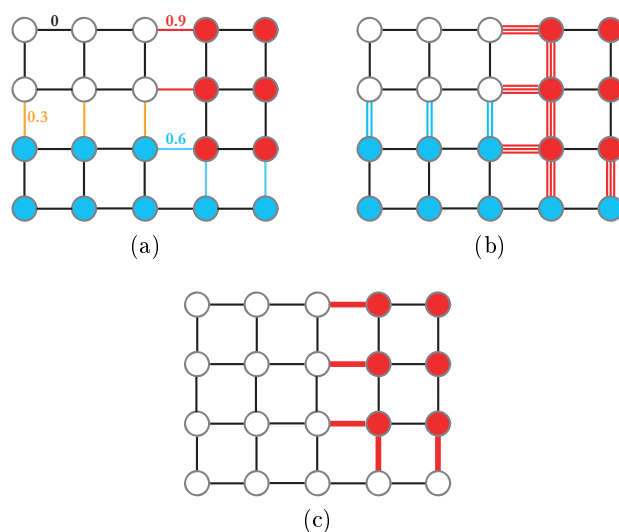


Figura 3.10: (a) Red, (b) segmentación jerárquica por aristas y (c) segmentación borrosa del Ejemplo 2.2.1

1. No todas las imágenes a segmentar son de tipo RGB, de forma que si es una imagen en escala de grises presenta el problema de que no se puede visualizar la segmentación con esta metodología si la imagen se encuentra segmentada con más de 256 regiones.
2. Si el número de regiones homogéneas es grande, entonces la imagen segmentada es casi igual a la original.
3. Dos regiones adyacentes pero segmentadas como diferentes incorrectamente, podrían visualizarse como una sola región.

Otra manera de visualizar una segmentación de una imagen es mediante el contorno de las regiones obtenidas (Figura 3.12). De forma que, la segmentación se puede representar de forma binaria o *mapa de bordes* clasificando los píxeles en dos clases $V = \{\text{negro}, \text{blanco}\}$, es decir, como una imagen en donde los píxeles coloreados en blanco representan los bordes de las regiones

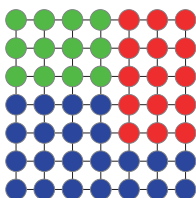


Figura 3.11: Visualización de una segmentación por medio de un mapa de regiones

(píxeles localizados en la frontera entre dos regiones distintas) y los píxeles coloreados en negro son aquellos en que sus vecinos pertenecen a su misma región.

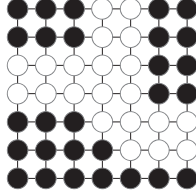


Figura 3.12: Visualización de una segmentación por medio de un mapa de bordes

Formalmente, si denotamos por $S(p)$ la región a la que pertenece un píxel p , se propone clasificar el conjunto de píxeles en $V = \text{negro} \cup \text{blanco}$ como se presenta a continuación:

Definición 3.4.1 Dada la red de una imagen $N(I) = \{ G = (V, E); D \}$, y dada una segmentación $S = \{R_1, \dots, R_k\}$, se definen las clases Negra y Blanca como:

1. Negra = $\{p \in V \text{ tal que } S(p) = S(p') \forall e = (p, p') \in E\}$
2. Blanca = $\{p \in V \text{ tal que existe } e = (p, p') \in E \text{ cons } S(p) \neq S(p')\}$

De este modo, la definición previa dice que si un píxel es coloreado blanco, es porque tiene al menos un vecino que pertenece a otra región. Por otro lado, si un píxel es coloreado negro, es porque todos sus vecinos pertenecen a su misma región. Sin embargo, esta metodología de visualización genera bordes de dos píxeles de grosor. Esto puede ser un problema al evaluar y comparar las salidas de una técnica de segmentación con las salidas de técnicas de detección de bordes cuyos bordes son de un píxel de grosor. Por lo que con esta metodología simplemente se pretende visualizar las regiones identificadas en el problema de segmentación, delimitando las fronteras que existen entre dichas regiones. En este sentido, una solución para obtener bordes de un píxel de grosor es asignar como borde a los píxeles pertenecientes a la región más grande. Es decir, se traspasa del problema de segmentación a uno de detección de bordes; este procedimiento se explicará en más detalle en el capítulo 4.

3.4.2. Metodología de visualización de una segmentación jerárquica

La visualización de una segmentación jerárquica se puede realizar a través de los nodos y de las aristas de una red, considerando las definiciones planteadas

anteriormente (definición 3.2.2 y definición 3.2.3). Esto se puede apreciar con el ejemplo que se presenta a continuación.

Ejemplo 3.4.1 *La Figura 3.13 ilustra tanto la segmentación jerárquica por nodos como por aristas correspondientes a la red del Ejemplo 2.2.1 considerando los siguientes parámetros:*

$$K = 4 \quad (\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4) = (0.0, 0.3, 0.6, 0.9, 1.0)$$

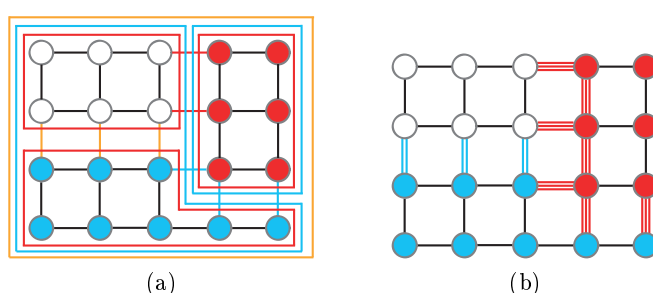


Figura 3.13: Segmentación jerárquica de una imagen $N(I)$

Así, se obtiene la segmentación jerárquica con cuatro particiones, la cual se puede apreciar en la Figura 3.13 (en este caso las segmentaciones obtenidas para α_2 y α_3 son las mismas).

En detalle, la Figura 3.13a, representa una segmentación jerárquica mediante los nodos de la red:

- S^0 , veinte segmentos (los nodos).
- S^1 , los tres grupos delimitados por líneas rojas.
- S^2 , los dos grupos delimitados por líneas azules.
- S^3 , sólo un grupo con todos los nodos, delimitado por la línea naranja.

Asimismo, en la Figura 3.13b, se aprecia la misma segmentación jerárquica pero en términos de las aristas de la red:

- B^0 , dado por los bordes rojo, azul y negro, lo que lleva a veinte clústeres (los nodos).
- B^1 , dado por los bordes rojo y azul, dando lugar a tres grupos.
- B^2 , dado por los bordes rojos, definiendo dos segmentos.
- $B^3 = \emptyset$, un segmento único contiene todos los nodos.

3.4.3. Metodología de visualización de una segmentación borrosa

Para visualizar una segmentación borrosa en base a los bordes se propone extender a un contexto borroso la definición 3.4.1 correspondiente a la construcción de un mapa binario. Es decir, se propone introducir una gradualidad en los bordes de las regiones segmentadas como se observa en la Figura 3.14b. Para ello, la clave está en clasificar los píxeles en una *Clase borrosa blanca* sobre el conjunto de nodos, y en una *Clase borrosa negra* la cual será construida como negación de la *Clase borrosa blanca*.

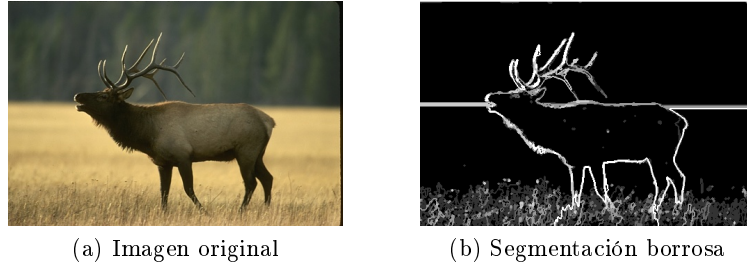


Figura 3.14: Ejemplo de visualización de una segmentación borrosa

Sea \tilde{B} una segmentación borrosa de imágenes, con función de pertenencia $\mu_B : E \rightarrow [0, 1]$. Se puede construir de forma natural las funciones de pertenencia μ_{blanca} y μ_{negra} como sigue:

Definición 3.4.2 Dada una red de imagen $N(I) = \{G = (V, E); D\}$ y dada una segmentación borrosa de imagen \tilde{B} con función de pertenencia $\mu_B : E \rightarrow [0, 1]$, definimos las clases borrosas Blancas y Negras de la siguiente manera:

1. $\mu_{Blanca}(p) = \mathcal{A}\{\mu_B(e), \text{para todo } e = (p, q) \in E\} \forall p \in V.$
2. $\mu_{Negra}(p) = N(\mu_{blanca}(P)) = 1 - \mu_{blanca}(P) \forall p \in V.$

Observación 3.4.1 Nótese que en la definición anterior, el grado de pertenencia de un píxel dado p a la clase borrosa blanca, es el resultado de la agregación \mathcal{A} de los valores de $\{\mu_{\tilde{B}}(p, q_i), i = 1, \dots, 4\}$, donde q_1, \dots, q_4 son los vecinos del píxel p . En general, el operador de agregación $\mathcal{A} : [0, 1]^4 \rightarrow [0, 1]$ utilizado debe satisfacer las siguientes propiedades establecidas en la definición 2.6.3 y se recuerdan a continuación:

1. \mathcal{A} es simétrica.
2. $\mathcal{A}(x_1, \dots, x_4) = 0$ si y solo si $x_i = 0, \forall i = 1, \dots, 4$, i.e., un píxel que está siempre en la misma región que sus vecinos, debería tener de grado cero de pertenencia a la clase frontera.

3. $\mathcal{A}(x_1, \dots, x_4) = 1$ si y solo si existe $i \in \{1, \dots, 4\}$ con $x_i = 1$, i.e., un píxel que está siempre en el borde debido a que tiene un píxel adyacente en una región diferente, debe tener grado uno de pertenencia a la clase frontera.

En esta memoria se ha utilizado el operador de agregación $\mathcal{A} = \text{máx}$, pero otra función de agregación puede ser considerada.

3.5. Relaciones entre algunos problemas de tratamiento de imágenes

Como se ha visto en las secciones anteriores, con el problema de la visualización de un problema de segmentación de imágenes, es muy habitual confundir problemas relacionados de tratamiento de imágenes debido a su alta correlación y similitud en las salidas. Por este motivo, en esta sección se establece una clarificación de algunos de los diferentes problemas de tratamiento de imágenes en base a la salida que proporcionan dichos problemas y los objetivos que persiguen.

Así, en esta sección se propone una nueva clasificación de problemas de tratamiento de imágenes de acuerdo a la salida que generan. Específicamente, en una primera Subsección, se presentan los distintos problemas de tratamiento de imágenes (Subsección 3.5.1). En la segunda Subsección se comenta brevemente sobre la relación entre los principales problemas (Subsección 3.5.2). Y, en la última Subsección se propone la nueva clasificación de los principales problemas y las técnicas que abordan estos problemas (Subsección 3.5.3).

3.5.1. Problemas de tratamiento de imágenes

Cada problema de tratamiento de imágenes es apropiado para obtener una salida específica, por ejemplo, de segmentación o de detección de bordes. En general, cada tarea o problema de tratamiento de imágenes se puede entender en términos de problemas de clasificación, en donde cada píxel de la imagen se clasifica según las características propias de la tarea correspondiente. A continuación se presentan formalmente las salidas que se pueden obtener con los principales problemas de tratamiento de imágenes:

3.5.1.1. Clasificación de píxeles con clases definidas a priori

En este apartado estarían incluidas aquellas situaciones en las que los píxeles deben ser clasificados en clases que han sido definidas a priori por el usuario. Si se denotan por C_1, \dots, C_t las clases a las que puede pertenecer un píxel, el problema busca identificar un mecanismo que permita clasificar cada píxel en las clases de interés.

Matemáticamente dada una imagen I y su red $N(I) = \{G = (V, E); D\}$, la clasificación asigna cada píxel a una de las t clases definidas a priori.

$$C_{clas} : V \rightarrow \{C_1, \dots, C_t\} \quad (3.1)$$

Se pueden distinguir dos tipos de clasificación de imágenes, supervisada y no supervisada. En la clasificación supervisada se utiliza un conjunto de entrenamiento en donde se clasifican algunos píxeles que se utilizan para clasificar el resto de los píxeles. En cambio, la clasificación no supervisada o *clustering* no necesita información previa sobre los píxeles a clasificar sino que identifican grupos de píxeles que posean características comunes.

Entre algunos problemas relacionados con la clasificación están la identificación de hierba cardo en cultivos de cereales mediante imágenes aéreas (Franco et al., 2018), reconocimiento facial (Ahonen et al., 2004), teledetección (Wacker y Landgrebe, 1972).

3.5.1.2. Segmentación de imágenes

Los problemas de segmentación de imágenes tratan de identificar regiones homogéneas. No parten de ninguna información a priori sobre el número de regiones, posibles objetos o incluso el tipo de imagen que se está analizando. La idea, al igual que pasa con el *clustering* de objetos en estadística, es establecer una partición de la imagen de manera que los elementos de dicha partición representen regiones con características similares.

Dada la red de una imagen $N(I)$, la segmentación clasifica cada píxel a una de las t regiones siguiendo las propiedades expuestas en la definición 2.3.1.

$$C_{seg} : V \rightarrow \{R_1, \dots, R_t\} \quad (3.2)$$

3.5.1.3. Detección de bordes

Aunque los problemas de detección de bordes tal y como se han definido anteriormente (definición 2.5.1), podrían ser incluidos en la sección 3.5.1.1 ya que el objetivo fundamental es el de clasificar a los píxeles en bordes y no-bordes, las restricciones que se imponen a este tipo de problemas (ver Canny, 1986, 1983) así como su relevancia hacen que puedan ser categorizados como un problema diferente.

Sea $N(I)$ la red de una imagen, la detección de bordes clasifica cada píxel como borde o no-borde.

$$C_{edge} : V \rightarrow \{0, 1\} \quad (3.3)$$

La salida que obtiene cualquier técnica de detección de bordes corresponde a una imagen binaria, en donde aparecen en blanco (1) aquellos píxeles

en donde se han detectados cambios de intensidades en la imagen original, mientras que los demás píxeles se mantienen negros (0). Esta imagen resultante simplifica la información de la imagen contenida en la imagen original, para hacerla más interpretable y fácil de analizar preservando propiedades estructurales importantes de la imagen como son los bordes de la imagen.

3.5.1.4. Clasificación borrosa de píxeles con clases definidas a priori

Los problemas de clasificación borrosa están presente cuando los píxeles deben ser clasificados en clases definidas a priori denotadas por C_1, \dots, C_t y el problema busca identificar un mecanismo que permita clasificar cada píxel en las clases de interés con un grado de pertenencia entre los valores $[0, \dots, 1]$.

Dada la red de una imagen $N(I)$, la clasificación asigna cada píxel a una de las t clases con sus respectivos grados de pertenencia $(\mu_{C_1}, \dots, \mu_{C_t})$.

$$C_{fuzzyclas} : V \rightarrow [0, 1] \times \dots \times [0, 1] = [0, 1]^t \quad (3.4)$$

Del mismo modo que los problemas de clasificación, los problemas de clasificación borrosa de píxeles pueden estar conformados por métodos supervisados y no supervisados.

Algunos problemas relacionados con la clasificación borrosa son los problemas de clasificación borrosa en la teledetección (Gómez et al., 2008; Gómez y Montero, 2008), los sistemas de asistencia al conductor (Madrid y Hurtik, 2016).

3.5.1.5. Detección borrosa de bordes

Las técnicas de detección borrosa de bordes pueden considerarse extensiones de la técnica clásica de detección de bordes.

Sea $N(I)$ la red de una imagen de tamaño $r \times s$, la detección borrosa de bordes clasifica en cierto grado de pertenencia cada píxel como borde o no-borde .

$$C_{fuzzyedge} : V \rightarrow [0, 1] \times [0, 1] \quad (3.5)$$

Donde el primer intervalo corresponde al grado de pertenencia de borde y el segundo al grado de pertenencia de no-borde.

3.5.1.6. Segmentación borrosa por aristas

Dada la red de una imagen $N(I)$, la segmentación borrosa por aristas clasifica con un grado de pertenencia cada arista como borde borroso.

$$C_{fuzzyseg} : E \rightarrow [0, 1] \quad (3.6)$$

3.5.2. Relación entre problemas

Cabe destacar que, se puede convertir la salida de una técnica que resuelve una determinada tarea en la salida característica de una técnica que resuelve otra tarea. Por este motivo, se puede decir que existen relaciones entre las diferentes tareas y técnicas. Por ejemplo, la salida de un algoritmo de detección de bordes puede servir de entrada para una segmentación de imagen, si se encuentra una partición del conjunto de píxeles en regiones conexas. De forma inversa, también es posible construir los bordes de un conjunto de objetos a través de un conjunto de píxeles que conectan píxeles de distintas regiones.

La segmentación de imágenes divide una imagen en objetos no solapados, por lo que la salida puede ser representada principalmente por objetos coloreados de distintos colores (ver Figura 3.15a), o bien, colorear el borde de cada objeto (ver Figura 3.15b).

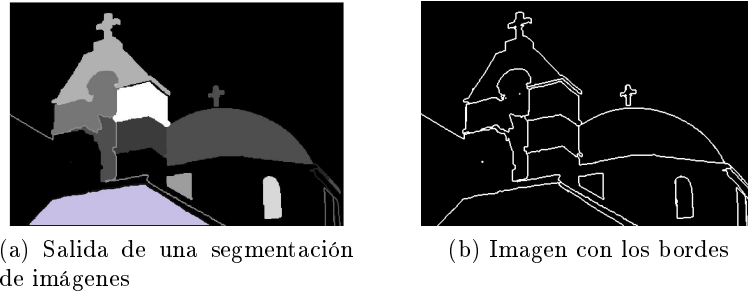


Figura 3.15: Ejemplo de transformación de una imagen segmentada a la detección de bordes

Matricialmente, la diferencia entre los objetivos o salidas obtenidas con la segmentación de imágenes y la detección de bordes se puede observar a continuación:

$$C_{seg} = \begin{vmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{vmatrix} \quad C_{edge} = \begin{vmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

Transformar la salida de un algoritmo de detección de bordes a una segmentación de imágenes no es tan trivial como en el caso inverso. Esto debido

a que la técnica de detección de bordes necesariamente no encierra los objetos a medida que va identificando los cambios de intensidad en los píxeles. Si se impone la condición de cerrar los bordes en la Figura 3.16a obtenida mediante un algoritmo de detección de bordes en regiones, se puede obtener la salida de la Figura 3.16b.

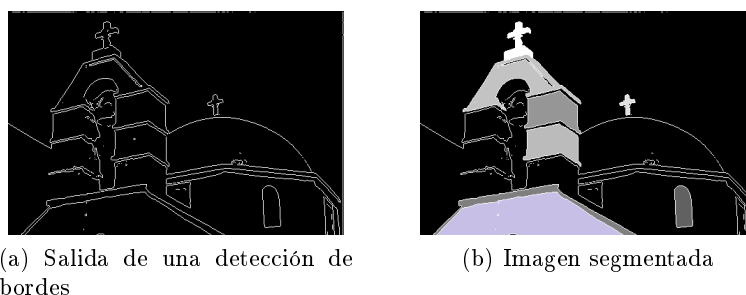


Figura 3.16: Ejemplo de transformación de la detección de bordes a una imagen segmentada

Por otro lado, si en la salida de un algoritmo de segmentación de imágenes dos o más regiones distintas y no adyacentes comparten características similares, éstas pueden pertenecer a la misma clase al aplicar una técnica de clasificación de imágenes. De manera contraria, desde la salida producto de una clasificación de imágenes, se puede obtener una segmentación de imágenes al asignar cada una de las regiones no adyacentes clasificadas en la misma clase como un único objeto. Igualmente, la salida de la técnica de *clustering* puede ser utilizado como paso previo para conseguir una segmentación de imágenes.

Además, también la segmentación borrosa y jerárquica están relacionadas como se mencionó en la sección 3.3. En la Figura 3.18 se muestra la salida de una segmentación borrosa producto de la agregación de la secuencia jerárquica obtenida con el algoritmo de segmentación jerárquica D&L representada por los bordes de las regiones (Figura 3.17) aplicado a la Figura 2.1c.

Una ventaja de la segmentación borrosa es que, al representar su salida se pueden observar graduaciones en los bordes de los objetos. En este sentido, se tendría una salida con bordes graduales y a la vez más informativa que las salidas binarias que ofrecen otros métodos clásicos.

En consecuencia, muchas más relaciones se pueden encontrar entre el resto de las tareas y técnicas de procesamiento de imágenes, pero es un tema que merece ser dedicado con más detalle y formalismo en futuras investigaciones debido a que es un tema extenso y novedoso.

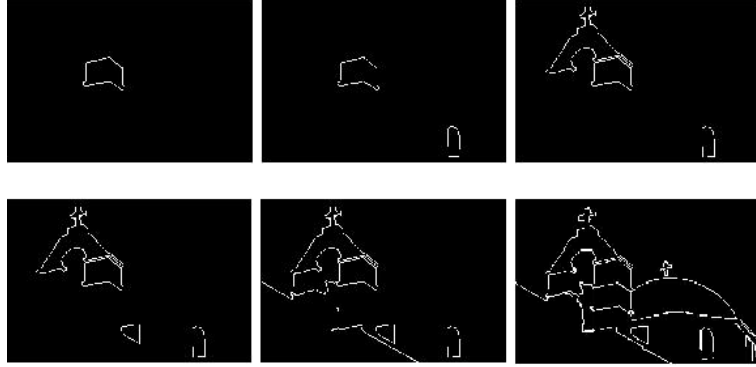


Figura 3.17: Segmentación jerárquica



Figura 3.18: Segmentación borrosa

3.5.3. Nueva clasificación de los principales problemas

Las principales tareas más conocidas de tratamiento de imágenes pueden diferenciarse de acuerdo al objetivo o salida que persiguen, además, pueden diferenciarse respecto a si se refieren a una técnica clásica o su versión extendida a un contexto borroso. Del mismo modo, algunas de estas tareas conocidas de tratamiento de imágenes, pueden ser distinguidas dentro de dicha clasificación según si se tratan de métodos supervisados o no supervisados.

De esta manera, teniendo en cuenta estas consideraciones mencionadas anteriormente, se propone una nueva clasificación sobre algunos problemas de tratamiento de imágenes, la cual fue documentada en (Guada et al., 2016). En la Tabla 3.1, se presenta resumidamente la clasificación realizada sobre algunos problemas de tratamiento de imágenes según la salida que consiguen (Guada et al., 2016).

Problemas	Técnicas
Clasificación de imagen (Lu y Weng, 2007)	· Métodos supervisados: algoritmos genéticos, MVS, redes neuronales, entre otros. · Métodos no-supervisado: clustering.
Clasificación borrosa de imagen (Nachtegaele et al., 2007)	· Métodos supervisados: versión borrosa de algoritmos genéticos, redes neuronales, entre otros. · Métodos no-supervisados: Clustering borroso.
Detección de bordes (Marr y Hildreth, 1980)	· Métodos no-supervisados: Canny, Prewitt, Sobel, entre otros.
Detección borrosa de bordes (Bustince et al., 2006b)	· Métodos no-supervisados: función de Epanechnikov como una función de pertenencia, entre otros.
Segmentación de imágenes (Peng et al., 2013)	· Métodos no-supervisados: <i>Thresholding</i> , <i>Watershed</i> , <i>Region segmentation</i> , <i>Graph partitioning</i> .
Segmentación jerárquica de imagen (Arbeláez et al., 2011)	· Métodos no-supervisados: Divide & Link (D&L), <i>gPb-owt-ucm</i> , entre otros.
Segmentación borrosa por aristas (Gómez et al., 2015a)	· Métodos no-supervisados: segmentación jerárquica algoritmo de segmentación, como el ID&L, utilizado para producir segmentación borrosa por aristas.

Tabla 3.1: Nueva clasificación de algunos problemas de tratamiento de imágenes

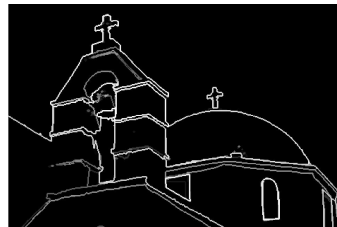
En la Figura 3.19 y así, como en las presentadas en las dos secciones anteriores (Figuras 2.6, 2.9 y 2.10), se pueden observar algunas de las salidas de distintos algoritmos de tratamiento de imágenes. Cada una de dichas figuras, son distintas debido a que la técnica utilizada para obtener cada una de ellas abordaban distintos problemas, como por ejemplo de clasificación, de segmentación, de detección de bordes, entre otros.



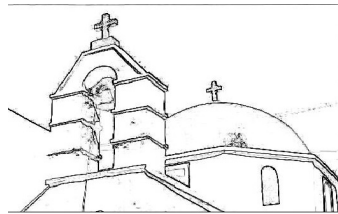
(a) Clustering



(b) Clustering borroso

(c) *Thresholding*

(d) Segmentación borrosa de imágenes



(e) Detección borrosa de bordes

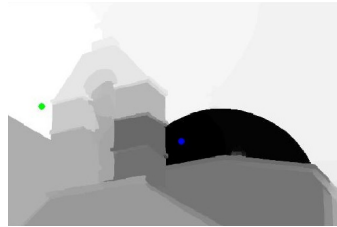
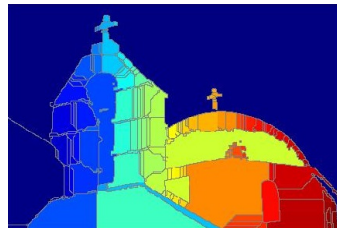
(f) *Random walker*(g) *Region grow*(h) *Watersheed*

Figura 3.19: Salidas de algunas técnicas de tratamiento de imágenes sobre la imagen de la iglesia 2.1a.

Capítulo 4

Nuevo algoritmo de detección de bordes, *Image-Divide and Link* (ID&L)

Si no conozco una cosa, la investigaré

Louis Pasteur

RESUMEN: Este capítulo está dedicado a introducir un nuevo algoritmo de detección de bordes basado en un enfoque de agrupación jerárquica de grafos. En detalle, este algoritmo produce una serie de particiones jerárquicas de regiones que vienen representadas a través de sus bordes. Este nuevo algoritmo está basado en la adaptación del algoritmo de segmentación jerárquica en redes Divide-and-Link (Gómez et al., 2015b) a un contexto de detección de bordes. En la sección 4.1, se describe detalladamente el algoritmo propuesto.

4.1. Algoritmo Image-Divide-and-Link (ID&L)

Siguiendo con los objetivos planteados en esta memoria y, en base a las relaciones entre problemas que existen entre la segmentación de imágenes, medidas de disimilitud entre píxeles y los problemas de detección de bordes, en esta sección se propone un algoritmo competitivo para el problema de detección de bordes. Este nuevo algoritmo de detección de bordes trabaja con cualquier tipo de imagen digital, como las mencionadas en la sección 2.1 del capítulo 2, y se basa en el algoritmo no supervisado de partición jerárquica de redes Divide-and-Link (D&L) (Gómez et al., 2015b) presentado en la sección 2.4 del capítulo 2. La idea general del algoritmo que se propone

consiste en extender el algoritmo D&L para obtener una secuencia jerárquica de mapa de regiones de una imagen y luego, transformar esta secuencia en una secuencia de mapa de bordes. Este nuevo algoritmo de detección de bordes se denominará Image-Divide-and-Link (ID&L) (Guada et al., 2018).

El ID&L consta de una serie de pasos, las cuales se describen en las siguientes Subsecciones. Específicamente, en la subsección 4.1.1, se recuerda cómo modelar una imagen a través de una red. Posteriormente, en el apartado de la subsección 4.1.2 se establece cómo determinar los umbrales a utilizar en el proceso iterativo para la partición jerárquica. Seguidamente, se describe cómo obtener los mapas de regiones jerárquicos (subsección 4.1.3). Después, se muestra cómo un mejoramiento a la red a tratar mediante una contracción de píxeles con un umbral preestablecido logra mejorar el ID&L reduciendo su tiempo computacional al evitar tratar regiones muy homogéneas (subsección 4.1.4). Luego, en la subsección 4.1.5, se muestra cómo transformar un mapa de regiones jerárquico en una secuencia de mapas de bordes. Finalmente, en la subsección 4.1.6 se presentan dos salidas generadas por el algoritmo ID&L.

4.1.1. Modelización de una imagen como una red

El primer paso del algoritmo consiste en modelizar una imagen I como una red $N(I) = \{G; D\}$. Para ello, se asocia un grafo a la imagen siguiendo la definición 2.2.2 y considerando una topología de 4-vecinos y, posteriormente se calculan los grados de disimilitud entre las aristas del grafo.

En esta memoria, las disimilitudes entre los píxeles se calculan a través de la distancia Euclídea entre la información espectral visible de los píxeles (ver Ecuación 2.1), como se verá posteriormente en el capítulo 6.1 de experiencias computacionales. Sin embargo, en la práctica esta medida puede ser calculada de otra manera, la cual depende de la tarea específica a realizar y del tipo de imagen a analizar.

Cabe destacar que, el algoritmo propuesto ID&L sólo necesita una medida de disimilitud simple entre los píxeles adyacentes mas no pares de medidas de similitud/disimilitud, como sí lo necesita el algoritmo D&L. Además, este tipo de disimilitud calculada para el algoritmo ID&L, no se puede actualizar en cada iteración a diferencia de lo que sucede con el algoritmo D&L para redes (Gómez et al., 2015b; Guada et al., 2018).

4.1.2. Umbrales en el proceso iterativo

Es importante resaltar que, el objetivo principal del algoritmo D&L para la agrupación jerárquica de redes es obtener una partición completa o dendrograma, en la que el número de grupos o clústeres aumenta sólo una unidad en cada iteración hasta que todos los nodos pertenezcan a clústeres *singleton*. Esto es significativo en el contexto del análisis de redes sociales, debido a que

proporciona una descripción del proceso por el cual una determinada red de elementos, se rompe en comunidades sucesivamente más pequeñas, hasta el punto en que las comunidades se identifican con elementos individuales. No obstante, en el contexto de procesamiento de imágenes, puede que tal descripción de profundidad máxima o dendrograma no sea realmente interesante y significativa, porque se busca separar los objetos en una imagen los cuales generalmente están compuestos de muchos píxeles, y por ende, una vez que se alcanza un cierto nivel de detalle, puede que no sea significativo persistir en la separación de píxeles en objetos muchos más pequeños.

En este sentido, la principal diferencia del algoritmo ID&L propuesto en esta memoria con respecto al algoritmo D&L, es que con el ID&L el dendrograma es prefijado mediante un parámetro pre-establecido K , el cual establece el número de iteraciones repetidas del proceso divisivo. En sí, este parámetro K representa el número de los distintos niveles de detalle de las regiones a encontrar en el proceso iterativo, y que a su vez, son representadas como mapa de bordes, de forma que cada repetición del proceso divisivo proveerá mayor detalle en comparación con la partición anterior obtenida. Por lo tanto, se considera que el valor K debe tomar valores pequeños, por lo general menor a 10, pero sobretodo tiene que ser un valor mucho más bajo que el número de píxeles en la imagen $N = r \times s$, el cual constituiría el límite superior en el caso de requerir un dendrograma completo. En general, este parámetro tiene que ser elegido teniendo en cuenta dos cuestiones, por un lado, que ayude a obtener una secuencia jerárquica que consiga los distintos niveles de detalle necesarios y, por otro lado, que el coste computacional de la obtención de las particiones se mantenga tan bajo como sea posible.

Otra gran diferencia, producto de esta profundidad máxima prefijada del proceso de partición, es que los umbrales $\alpha_1, \dots, \alpha_K$ -que son los que determinan en cada iteración cuáles serán los nodos pertenecientes a las distintas regiones- también se deben prefijar una vez que se ha establecido el valor de K .

Así, la idea principal es que estos umbrales α tienen que seguir un esquema decreciente, de forma que, en los primeros pasos del algoritmo, α debe tomar valores relativamente altos para que así, queden separados únicamente los píxeles adyacentes fuertemente disímiles. Esto permite reconocer primero los objetos o regiones más contrastantes de la imagen. Seguidamente, a medida que va decreciendo el valor α , más píxeles van quedando separados en las regiones ya detectadas, permitiendo identificar más detalles en los objetos. Obviamente, el rango en el que α puede variar, depende de los valores de las disimilitudes asociadas a las aristas; si $\alpha > \bar{d}$, en donde $\bar{d} = \max(D)$, todos los píxeles se asignarían al grupo trivial V . Por el otro lado, si $\alpha \leq \underline{d}$, en donde $\underline{d} = \min(D)$, cada píxel se asignaría a un grupo trivial con sólo un elemento (i.e. *singletons*).

En general, se pueden adoptar muchas soluciones al problema de interés

seleccionando un número apropiado de niveles de detalle K y obteniendo una secuencia adecuada de K umbrales $\alpha_1 > \dots > \alpha_K$, de tal manera que se produzca un proceso de división, adquiriendo niveles de detalle adicionales y relevantes en cada paso. Así, ambas decisiones tienen que depender del tamaño y las características del problema de interés. Es importante resaltar que en este apartado no se propone una manera concreta de asignar estos umbrales, sin embargo, en la subsección 6.1.1 se propone un esquema de aprendizaje sobre una colección de distribuciones de los umbrales usando un conjunto de imágenes de entrenamiento. Por otro lado, cuando no existan imágenes de entrenamiento, se puede recurrir a un método no supervisado como las *funciones de disimilitud restringida* (Bustince et al., 2006a), pero este tema se abordará como trabajo futuro.

4.1.3. Determinación de los mapas de regiones jerárquicos

Una vez que se han escogido la profundidad del dendograma K y los umbrales $\underline{d} = \alpha_0 > \alpha_1 > \dots > \alpha_{K+1} = \bar{d}$, se obtienen las particiones jerárquicas de la red de una imagen de forma similar a los pasos a seguir del algoritmo D&L descrito en la sección 2.4.1. El proceso de la determinación de los mapas de regiones jerárquicos se describe a continuación:

Para cada t de 0 a $K + 1$ hacer lo siguiente:

1. Considerar $\alpha = \alpha_t$ y dividir el conjunto de aristas E en $Div = \{e \in E / d_e \geq \alpha\}$ y $Link = \{e \in E / d_e < \alpha\}$.
2. Ordenar todas las aristas en E en dos pasos. En primer lugar, ordenar las aristas en Div desde la disimilitud más alta a la más baja d_e . En segundo lugar, añadir las aristas en $Link$ desde la disimilitud más baja hasta la más alta d_e .
3. Sobre esta disposición de aristas, se construye un bosque soporte S siguiendo el algoritmo de Kruskal (Kruskal, 1956).
4. La partición \mathcal{P}^t generada en esta etapa se construye como los componentes conexos del bosque soporte S^* obtenido después de la supresión de los bordes de E que pertenecen a las aristas divisivas Div (i.e. $\mathcal{P}^t = \{\text{componentes conexos de } S - Div\}$).
5. Actualizar el conjunto de aristas E eliminando aquellas aristas que unan las diferentes regiones de \mathcal{P}^t .

Nótese que dos particiones sucesivas, \mathcal{P}^{t-1} y \mathcal{P}^t , serán diferentes si existe una arista $e_{ab} \in E$ tal que $\alpha_{t-1} > d_{ab} \geq \alpha_t$. De esta manera, el número de umbrales diferentes que pueden proporcionar efectivamente particiones diferentes es $|\{d_e | e \in E\}| \leq m = |E|$. Sin embargo, el número de umbrales

establecidos es importante para determinar las particiones obtenidas después de la última iteración, debido a que la partición conseguida después de utilizar un único umbral, digamos α_2 , no es necesariamente el mismo que se obtendría, por ejemplo, estableciendo dos umbrales $\alpha_1 > \alpha_2$. También, observe que el algoritmo anterior tiene un mayor bucle basado en la profundidad jerárquica K , cuyo límite superior es $n = |V|$, que se alcanza cuando $|\mathcal{P}^i| = |\mathcal{P}^{i-1}| + 1$ para todo i . Como se mencionó anteriormente, en el contexto de obtención de mapas de bordes, definir un número muy grande de particiones podría obviamente ser poco informativo, por ende, el parámetro K debe tomar valores relativamente pequeños. Esto contribuye también a reducir significativamente el tiempo computacional necesario para producir la secuencia de particiones.

Observación 4.1.1 *A pesar de que (Felzenszwalb y Huttenlocher, 2004; Soille, 2008; Guimarães et al., 2012; Xu et al., 2017) también utilizan bosque soporte basados en un algoritmo similar al Kruskal, la principal diferencia con sus trabajos es que el algoritmo propuesto ID&L utiliza un bosque soporte que se construye variando el orden de las aristas; es decir, un orden decreciente de disimilitudes para las aristas cuyo peso sea mayor o igual que un umbral y un orden creciente para las aristas restantes. Además, el algoritmo propuesto por (Felzenszwalb y Huttenlocher, 2004) no produce explícitamente una jerarquía de particiones como lo hace el algoritmo ID&L. En contraste con el algoritmo ID&L, los algoritmos presentados en (Ouzounis y Soille, 2011; Najman et al., 2013) realizan sólo una construcción del mínimo árbol soporte.*

4.1.4. Redes gruesas en el algoritmo de partición jerárquica

Las imágenes digitales pueden contener regiones muy homogéneas, las cuales pueden ser tratadas como un solo nodo, con el fin de evitar que se subdividan en el proceso iterativo pudiendo generar ruido en la imagen resultante. En más detalle, los grupos de nodos enlazados por aristas de poca disimilitud, tienen que ser identificados para luego excluirlos del proceso de partición. Para ello, se utilizan las redes gruesas (*coarse nets*) y, a su vez, de esta manera se logra reducir el tiempo computacional. Esto constituye otra gran diferencia entre el algoritmo D&L con el algoritmo propuesto ID&L.

En general, la idea es contraer los píxeles de la red $N = \{G; D\}$, de forma que antes de que se empiece el proceso de partición, cada una de las regiones homogéneas se contraigan en un solo nodo (denominado nodo contraído o reducido). Este tipo de contracción es una técnica de grafo bien conocida (ver por ejemplo Gondran et al., 1984, para más detalles).

Así, sea α_τ un umbral de disimilitud predefinido, los puntos extremos a, b para cualquier arista e_{ab} con una disimilitud $d_{ab} \leq \alpha_\tau$ serán considerados como un único nodo, y la red se reorganiza de manera que las aristas

incidentes a este nuevo nodo son aquellas que incidían en cualquiera de esos puntos extremos. En consecuencia, nunca se separarán en el proceso de partición jerárquica los píxeles incluidos en estos nuevos nodos contraídos. La operación de contracción anterior puede realizarse de la siguiente manera: sea $E(\alpha_\tau) = \{e_{ab} \in E \mid d_{ab} \leq \alpha_\tau\}$ el conjunto de aristas con disimilitudes más bajas a la predefinida con el umbral; cualquier componente conexo al grafo parcial $G(\alpha_\tau) = (V, E(\alpha_\tau))$ se fusiona en un nuevo nodo, de modo que las aristas que incidían en el nodo contraído correspondan a las aristas (en E) que inciden en algunos de los píxeles del componente conexo asociado. Este procedimiento se puede observar en la Figura 4.1, en donde una red de 6 nodos (a, b, c, d, e, f) se contrae en una red de 5 nodos.

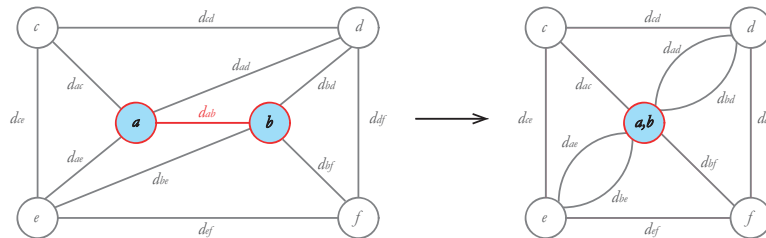


Figura 4.1: Contracción de nodos $d_{ab} \leq \alpha_\tau$

En resumen, la operación de contracción de nodos obtiene una nueva red $\underline{N} = (\underline{V}, \underline{E})$, en donde \underline{V} se refiere al conjunto de componentes conexos de $G(\alpha_\tau)$ y toda arista de \underline{E} es asociada a la arista original de E con una disimilitud mayor a α_τ .

El procedimiento de partición jerárquica descrito anteriormente se aplica a esta red contraída, produciendo particiones no triviales $\{\underline{\mathcal{P}}^1, \dots, \underline{\mathcal{P}}^K\}$. Entonces, el proceso de partición se termina cuando para cualquier t , las regiones $\underline{\mathcal{C}}_t \in \underline{\mathcal{P}}^t$ se expanden para que se recuperen los píxeles originales.

La adición de este paso de contracción generaliza el proceso de partición jerárquica descrito en la sección 2.4.1 y en la subsección 4.1.3, en el sentido de que para los valores $\alpha_\tau < \underline{d}$ se obtienen las mismas particiones como si no se aplicara la operación de contracción. Específicamente, como se ilustra en el siguiente ejemplo, la aplicación anterior de esta operación de contracción de nodos constituye una mejora significativa con respecto al algoritmo básico, tanto en términos de rendimiento como de esfuerzo computacional.

Antes de que se muestre un ejemplo, se debe mencionar que la contracción de nodos en el algoritmo ID&L también se puede realizar considerando solamente los vecinos de cada píxel; ésto sin tener en cuenta los valores de disimilitud. Por ejemplo, cada máscara $L \times L$ puede ser contraída en un píxel siguiendo la misma idea. Por lo tanto, el tratamiento de máscaras como un solo nodo es equivalente a como si la imagen se redujera y a su vez eliminando detalles mínimos o ruido que puedan estar presentes en ellas. Por lo tanto, esta idea puede proporcionar una reducción adicional del ruido contenido en

una imagen. Igualmente, una vez analizadas las imágenes, estas máscaras se vuelven a expandir a su tamaño original.

Ejemplo 4.1.1 (Ejemplo de la partición jerárquica de regiones) Se tiene la imagen en escala de grises mostrada en la Figura 4.2a de una dimensión de 5×6 . Cualquier píxel $(i, j) \in \{1, 2, 3, 4, 5\} \times \{1, 2, 3, 4, 5, 6\}$ tiene un valor de intensidad de gris $f(i, j) \in \{1, 2, 3, 4, 5\}$, como se observa en la Figura 4.2b. Se toma $d((i, j), (i', j')) = |f(i, j) - f(i', j')|$ como las disimilitudes entre los píxeles. La red de la imagen $N = \{G; D\}$ se muestra en la Figura 4.2c.

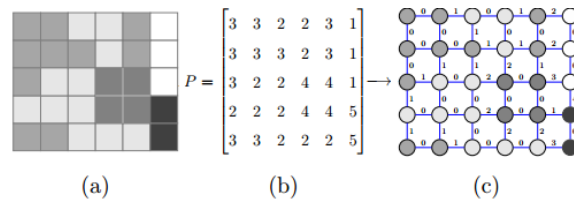


Figura 4.2: (a) Imagen en escala de grises, (b) la matriz asociada y (c) la red N

Para esta red, las disimilitudes máxima y mínima vienen dada por $\bar{d} = 4$ y $\underline{d} = 0$. Para fines ilustrativos, sea $K = 4$ (teniendo $K + 2$ particiones, en donde dos son particiones triviales y $K = 4$ son las no triviales) tal que

$$\alpha_0 = 5 > \bar{d} \geq \alpha_1 = 4 > \alpha_2 > \alpha_3 > \alpha_4 = 1 > \underline{d} = 0 \geq \alpha_5 = 0$$

Excluyendo las particiones \mathcal{P}^0 and \mathcal{P}^5 , las $K = 4$ particiones no triviales obtenidas a través del proceso de partición básica expuesta en la sección 4.1.3 (es decir, sin aplicar la contracción de nodos) se muestra en la Figura 4.3. Observe que el número s_t de regiones C_i^t obtenidas en cada iteración t son $(s_0, s_1, \dots, s_5) = (1, 2, 3, 8, 16, 30)$. Especialmente, al nivel máximo no trivial de detalle, i.e. después de usar el umbral $\alpha_4 = 1$, algunos píxeles adyacentes con la misma intensidad de gris se colocan en diferentes grupos. En realidad, esta situación indeseable comienza a ocurrir con el umbral $\alpha_3 = 2$, ya que el requisito de evitar el ciclo impuesto en la construcción del bosque soporte también evita que se añadan algunas aristas de enlace. Como resultado,

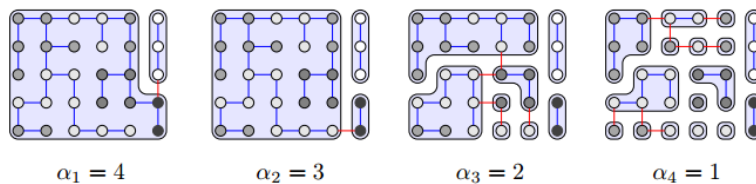


Figura 4.3: Iteraciones del proceso de partición jerárquica de regiones

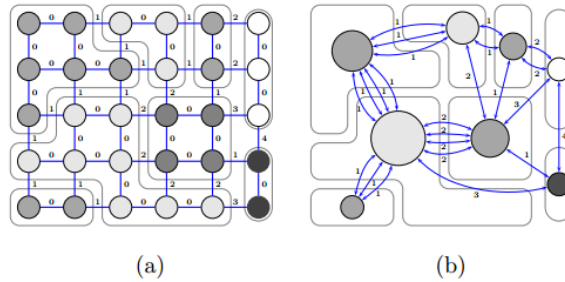


Figura 4.4: (a) Red original y (b) red contraída

muchos píxeles adyacentes similares se separan en las últimas iteraciones, lo que hace un número de regiones innecesariamente alto.

Sin embargo, se puede evitar fácilmente este comportamiento aplicando la operación de contracción en la red inicial. En las Figuras 4.4 y 4.5, se muestra la red \underline{N} obtenida luego de la contracción con $\alpha_\tau = 0$. En más detalle, la Figura 4.4(a) representa a la red original, $G(V, E)$, en donde $|V| = 30$ y $|E| = 49$; y la Figura 4.4(b) se refiere a la red $\underline{G} = (\underline{V}, \underline{E})$ usando $\alpha_\tau = 0$, en donde $|\underline{V}| = 8$ y $|\underline{E}| = 24$.

Específicamente, las particiones no triviales sucesivas del algoritmo jerárquico mejorado con la contracción de nodos, se muestran en la Figura 4.5. Luego, $(s_0, s_1, \dots, s_5) = (1, 2, 3, 5, 8, 30)$; y las regiones obtenidas luego de expandir los nodos contraídos, no separan aquellos píxeles adyacentes que posean la misma intensidad de gris. Además, se puede apreciar que, en este caso el algoritmo tiene que realizar un esfuerzo computacional mucho menor debido a que el número de nodos y aristas considerados se reducen significativamente. Entonces, la contracción de nodos introducida en la subsección 4.1.4, ayuda a mejorar el algoritmo de partición jerárquica ID&L, permitiendo claramente una adaptación mucho mejor al contexto de tratamiento de imágenes y en especial a la tarea de detectar los bordes de las

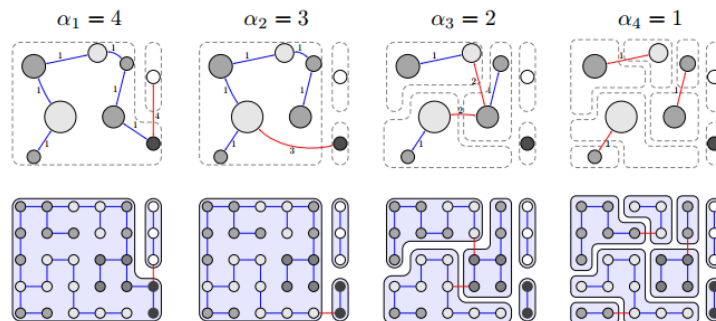


Figura 4.5: Contracción de nodos sobre la partición jerárquica de regiones; las aristas de cortes de color rojo

regiones encontradas.

4.1.5. Transformación de un mapa de regiones a un mapa de bordes

Una vez que se han determinado las regiones en una red, se deben identificar sus bordes o contornos para proceder a su visualización a través de ellos. Las particiones del conjunto de nodos en familias de regiones conexas y ordenadas jerárquicamente obtenidas en las etapas anteriores, pueden ser delimitadas por los bordes de estas regiones. Además, estos bordes pueden ser fácilmente identificados mediante el conjunto de aristas divisivas de los bosques soporte. Las aristas bordes serían entonces el conjunto mínimo de aristas de la red que separan las regiones de una partición de una imagen (Gómez et al., 2015a).

A continuación, se presenta un procedimiento para transformar estas salidas de mapas de regiones en un conjunto de mapa de bordes, similares a las salidas típicas de los algoritmos de detección de bordes. Para ello, primero es necesario clasificar los píxeles de un mapa de regiones jerárquico en dos clases, denominadas *Blanca* y *Negra*, como se muestra a continuación:

Definición 4.1.1 Sea $N = \{G = (V, E), D\}$ la red de una imagen, y sea $\mathcal{P} = \{C_1, \dots, C_t\}$ la partición de dicha red. Además, $C(i, j) \in \mathcal{P}$ denota la región a la que pertenece el píxel (i, j) con $(i, j) \in V$ para todo $i \in \{1, \dots, r\}$; $j \in \{1, \dots, s\}$. Entonces, las clases *Negra* y *Blanca* se definen de la siguiente manera:

- $Negra = \{(i, j) \in V \mid C(i, j) = C(i', j') \text{ para todo } e = \{(i, j), (i', j')\} \in E\}$
- $Blanca = V - Negra$

La definición anterior proporciona un método para visualizar cada mapa de región jerárquico a través de un mapa de bordes, en el cual un píxel está coloreado de blanco si tiene al menos un píxel adyacente en otra región y los píxeles rodeados por píxeles de la misma región están coloreados en negro. De esta manera, es posible representar un mapa de regiones como un mapa de bordes automático (los píxeles de la clase *Blanca* son bordes y los píxeles de la clase *Negra* no son bordes).

Sin embargo, obsérvese que después de realizar esta transformación los bordes tienen un grosor de dos píxeles, debido a que el procedimiento coloca simultáneamente un píxel borde en el límite de ambas regiones. Esto no cumple con uno de los criterios mencionados en la sección 2.5 acerca de una buena salida de un algoritmo de detección de bordes. Además, esto complica cualquier comparación que se desee hacer con este tipo de salida con la de otras técnicas de detección de bordes, que producen bordes de un píxel de grosor. Luego, una manera de eliminar este problema, es asignar como borde

los píxeles bordes de la región más grande, es decir, la región que posea un mayor número de píxeles es a la que se le asignará el borde. De esta manera, el mapa de bordes final estaría conformado por bordes de sólo un píxel de grosor.

Este proceso de transformación de mapa de regiones a mapas de bordes, constituye el último paso del algoritmo ID&L propuesto en la presente memoria. Por lo tanto, la salida del algoritmo ID&L, estaría conformada por una familia de mapas de bordes automáticos $\mathcal{B}^t, t = 1, \dots, K$ los cuales fueron obtenidos mediante la aplicación de este último paso sobre cada partición de regiones jerárquicas $\mathcal{P}^t, t = 1, \dots, K$.

Observación 4.1.2 Aunque esta transformación produce una familia de mapas binarios de bordes, se puede obtener un único mapa de bordes suavizado o soft mediante la agregación del conjunto de particiones (Gómez et al., 2015a).

4.1.6. Salida del algoritmo Image Divide and Link

La Figura 4.6, muestra la salida generada por el algoritmo ID&L, en donde se pueden observar los mapas de bordes resultantes correspondientes a la imagen del oso perteneciente a la BSDS500. Además, se puede observar la evolución del proceso jerárquico, el cual muestra los distintos niveles de detalles de la imagen obtenidos en cada iteración.

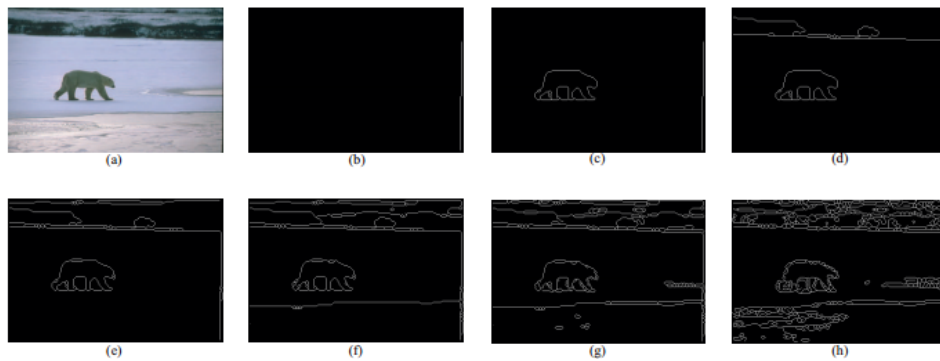


Figura 4.6: Detección de bordes obtenida con el algoritmo ID&L en la imagen del oso. (a) Imagen original. (b)-(h) Mapa de bordes automáticos resultantes del vector de umbrales $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8) = (47.79, 40.55, 33.29, 28.47, 23.63, 18.79, 11.55, 4.30)$ y con $L = 3$

En resumen, este nuevo algoritmo de detección de bordes es divisivo y no supervisado. El algoritmo tiene una complejidad polinomial y a su vez un coste computacional bajo. Otra de sus ventajas es que, puede ser utilizado en grandes redes y, también puede ser utilizado en una variedad de aplicaciones.

Capítulo 5

Evaluación del ID&L

*El tamaño de tu éxito será del tamaño
de tu esfuerzo*

Francisco de Miranda

RESUMEN: En la primera sección de este capítulo, se describe la metodología de evaluación basada en bordes que servirá de base a la metodología de evaluación propuesta (sección 5.1). Posteriormente, en la sección 5.2, se presenta una metodología de evaluación de bordes para algoritmos con salidas jerárquicas.

5.1. Metodología de evaluación a través de bordes

Uno de los métodos más comunes y utilizados para evaluar la efectividad de una salida de un algoritmo de tratamiento de imágenes es mediante la evaluación subjetiva, es decir, en donde un humano compara visualmente los resultados de la imagen resultante. Otro método común alternativo es la evaluación supervisada, en donde una imagen resultante es comparada con una imagen de referencia procesada (Zhang et al., 2008).

En esta sección, se describe el método de evaluación supervisada basado en bordes en el contexto de detectores de bordes (ver por ejemplo Arbeláez et al., 2011). Esta metodología considera una base de datos con distintas imágenes divididas en datos de entrenamiento, de prueba y de validación, en donde cada imagen de la base de datos viene acompañada de un conjunto de mapas de bordes realizada por humanos que sirven de referencia (*ground truth*) para evaluar los mapas de bordes automáticos generados por un algoritmo que detecte bordes (Arbeláez et al., 2011); ver por ejemplo las Figuras 5.1b-5.1f que son los mapas de bordes humanos realizada a la Figura 5.1a.

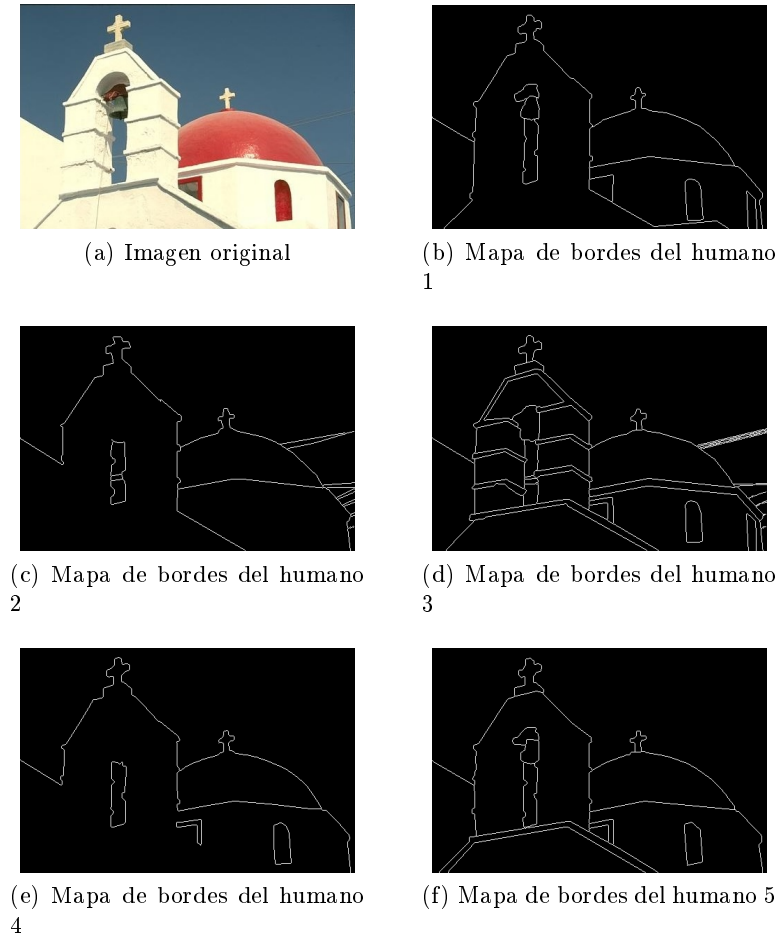


Figura 5.1: Mapa de bordes sobre la imagen de la *iglesia*

La idea general de esta metodología de evaluación consiste en tomar la salida de un algoritmo y evaluarla con el conjunto de referencias humanas (comparación uno-a-muchos), para ello primero se hace una comparación uno-a-uno entre cada una de los mapas de bordes humanos con la salida del algoritmo, y luego se realiza una agregación para obtener el resultado final de la comparación uno-a-muchos. En más detalle, para cada comparación de la salida de un algoritmo con cada referencia humana, se utilizan las medidas de *precision and recall* (P&R). Este sistema, determina a través del *precision*, la fracción de píxeles correctamente clasificados como borde o no-borde, es decir, evalúa la capacidad de predicción del algoritmo. Con el *recall*, se determina la sensibilidad del algoritmo, es decir, esta medida calcula la proporción de píxeles que son bordes y que son clasificados correctamente. Formalmente, dado un mapa de bordes automático E_{cd} como candidato a la evaluación y un mapa de bordes humano E_{gt} como referencia, se calculan

los valores de *precision* y *recall* mediante las siguientes ecuaciones:

$$precision(E_{cd}, E_{gt}) = \frac{tp}{tp + fp} \quad (5.1)$$

$$recall(E_{cd}, E_{gt}) = \frac{tp}{tp + fn} \quad (5.2)$$

En donde, *tp* se refiere a los verdaderos positivos (*true positive*) e indica los píxeles que han sido clasificados correctamente como borde, *tn* son los verdaderos negativos (*true negative*) y se refiere a los píxeles que han sido clasificados correctamente como no-borde, *fp* son los falsos positivos (*false positive*) e indica los píxeles que han sido clasificados incorrectamente como borde, y *fn* corresponde a los falsos negativos (*false negative*) y se refiere a los píxeles que son bordes y que no han sido clasificados como tal. Así, estos contadores se utilizan para comparar los resultados del algoritmo con la imagen de referencia, creando con ellos una matriz de confusión. Cabe destacar que, para el cálculo de estos valores se considera una distancia umbral δ , que especifica el nivel de tolerancia para una pequeña franja de localización de error. Esto, porque hay píxeles que identifica el algoritmo como bordes pero que en los mapas de bordes humanos se encuentran desplazados en unos pocos píxeles y entonces serían clasificados como falsos positivos, como se pueden ver en las imágenes de la Figura 5.2.

Luego, con estas medidas de P&R se calcula su proporción compuesta mediante la media armónica entre ellos dando como resultado una medida *F* que mide la calidad del algoritmo respecto a la imagen de referencia y, que constituye una de las alternativas más aceptada en los últimos años para evaluar cada comparación uno-a-uno (Arbeláez et al., 2011; Estrada y Jepson, 2009).

$$F(E_{cd}, E_{gt}) = \frac{precision(E_{cd}, E_{gt}) \times recall(E_{cd}, E_{gt})}{\beta precision(E_{cd}, E_{gt}) + (1 - \beta) recall(E_{cd}, E_{gt})}, \quad (5.3)$$

con $\beta = 0.5$ para obtener la media armónica.

Una vez obtenidos los valores de *F* para cada comparación uno-a-uno, se procede a agregar dichos valores *F* asociados a la imagen *I*, para así obtener la medida de calidad de la comparación uno-a-muchos entre la salida del algoritmo con el conjunto de referencias humanas de cada imagen *I*. Esta medida de agregación para la imagen *I* se denota como F_I y se obtiene mediante un operador de agregación \mathcal{A} . Este operador \mathcal{A} regula la semántica de cómo agregar los resultados de la comparación uno-a-uno a una medida de calidad para la comparación uno-a-muchos, y usualmente se utiliza el operador de agregación *máximo*, aunque dependiendo del contexto se puede usar un operador distinto como se verá más adelante.

Por lo tanto, sea $E_{gt}^I = \{E_{gt,1}^I, \dots, E_{gt,n}^I\}$ el conjunto de mapas de bordes humanos para la imagen *I*. El resultado F_I de la comparación uno-a-muchos

de un mapa de bordes automático E_{cd} con respecto a E_{gt}^I viene dado por:

$$F_I(E_{cd}, E_{gt}^I) = \mathcal{A}(F(E_{cd}, E_{gt,1}^I), \dots, F(E_{cd}, E_{gt,n}^I)). \quad (5.4)$$

Observe que el valor F_I mide para una determinada configuración paramétrica el rendimiento de un procedimiento de detector de bordes con respecto a la imagen I . Con el fin de obtener una medida global del rendimiento para tales configuraciones paramétricas sobre todas las imágenes de la base de datos, los valores F_I obtenidos en las distintas imágenes del conjunto son por lo general promediadas a través de la media aritmética. Luego, sea F^* dicha medida global de rendimiento. Nótese que para las imágenes del conjunto de datos de entrenamiento, esa medida global F^* se obtiene con cada configuración paramétrica del método evaluado, mientras para los datos de prueba, la medida global F^* se obtiene con la configuración paramétrica óptima encontrada en el conjunto de entrenamiento. Específicamente, dicha configuración paramétrica óptima representa el valor más alto de F^* sobre el conjunto de entrenamiento.

Observación 5.1.1 *De hecho, esta metodología de evaluación es muy similar al enfoque propuesto por (Martin et al., 2004), salvo por dos cuestiones. En primer lugar, la correspondencia de los bordes en los dos mapas de bordes no se lleva a cabo utilizando el algoritmo CSA, sino que en su lugar se usa el algoritmo de validación de (Estrada y Jepson, 2009), que se puede descargar públicamente en (Kermit Image Toolkit, 2015). Esto con el fin de evitar el comportamiento no determinista del algoritmo CSA, así como para permitir que los píxeles bordes sean coincidentes con más de un píxel en la imagen de contrapartida. En segundo lugar, el manejo de las imágenes de referencia humana múltiple es diferente. En (Arbeláez et al., 2011) y (Martin et al., 2004) la comparación uno-a-muchos genera una matriz de confusión combinada, de forma que la información de todas las comparaciones uno-a-uno se funden indeleblemente. En nuestro enfoque, cada comparación uno-a-uno, se lleva a cabo de forma independiente, y los resultados sólo se fusionan después. De esta manera, esta alternativa es más informativa e interpretable, así como más evidente en la medición entre el “parecido humano” con los mapas de bordes automáticos.*

5.2. Nueva metodología de evaluación de bordes

Una vez que se ha desarrollado un algoritmo que detecta bordes basado en una partición jerárquica, el ID&L, es necesario comparar las salidas que se obtienen con dicho algoritmo con la de otras técnicas bien sean jerárquicas o no, y éstas a su vez, evaluarlas con los mapas de bordes realizados por humanos utilizados como referencia. En el mundo del tratamiento de imágenes, existe una metodología muy utilizada y aceptada denominada *evaluación basada*

en bordes explicada en la sección anterior. Esta metodología de evaluación ha sido desarrollada por (Martín et al., 2004), y ha sido muy extendida y por ende, existen más base de datos y trabajos con referencias con este tipo de evaluación (ver por ejemplo Estrada y Jepson, 2009; Liu y Haralick, 2002; López-Molina et al., 2013a; Baddeley, 1992).

De este modo, el objetivo de esta sección es proponer una metodología que permita evaluar un algoritmo jerárquico en el contexto de detectores de bordes mediante una adaptación del método de evaluación basada en bordes de (Martín et al., 2004). Así, se puede comparar la salida del ID&L con técnicas de detección de bordes muy conocidas.

En la sección previa se describió la metodología para evaluar un algoritmo de detección de bordes que produzca una única salida de mapa de bordes para cada configuración paramétrica del método sobre una imagen. Sin embargo, ante un procedimiento jerárquico como el algoritmo ID&L que produce una secuencia de mapa de bordes por cada configuración paramétrica en una imagen, se debe adaptar la metodología de evaluación haciendo comparaciones muchos-a-muchos. Es decir, comparar los mapas de bordes automáticos con los mapas de bordes humanos para que sean comparados por cada imagen y para cada configuración paramétrica. Asimismo, también se debe hacer una adaptación sobre la metodología de evaluación para las secuencias de los mapas de bordes automáticos.

Entonces, para afrontar comparaciones muchos-a-muchos en el contexto de la evaluación de la secuencia jerárquica automática, se sigue el esquema que se propuso en (Rodríguez et al., 2016). En principio, la metodología de evaluación consiste en considerar un conjunto de referencias humanas por cada imagen, en donde cada una de ellas presenta diferentes niveles de detalle según lo que cada humano haya identificado. Además, una ventaja de una salida jerárquica es que produce una serie de particiones o una secuencia consistente de mapas de bordes, identificando jerárquicamente los objetos de una imagen con distintos niveles de detalle. Esta secuencia se puede aproximar entonces, a los distintos niveles de detalle que los humanos pueden reconocer en una imagen. Es decir, un conjunto de referencias humanas puede expresar distintos niveles de detalle por cada imagen, de forma que, existe una partición de la secuencia jerárquica que mejor aproxima el nivel de detalle de cada humano de referencia. En particular, una secuencia jerárquica siempre proporciona un mapa de borde automático que mejor aproxima a un mapa de bordes humano de referencia. Por lo tanto, se propone que han de ser estos mejores mapas de bordes automáticos los que deben proporcionar la base para la evaluación de la secuencia jerárquica como un todo. Así, por medio del esquema propuesto en (Rodríguez et al., 2016), el problema de la comparación muchos-a-muchos se reduce a una comparación uno-a-muchos, en el cual el mapa de bordes automático que más se asemeja a un cierto mapa de bordes humano ese usa para comparar la secuencia con dicha referencia

humana. De forma que, dichas comparaciones uno-a-uno se llevan a cabo como se describió anteriormente y sus resultados se agregan para luego hacer una comparación de uno-a-muchos F_I para cada imagen y, posteriormente en una medida global F^* para cada configuración paramétrica siguiendo los mismos principios.

Formalmente, sea $\mathbf{E}_{gt}^I = \{E_{gt,1}^I, \dots, E_{gt,n}^I\}$ el conjunto de referencia humana para una imagen I . Entonces, para cada mapa de borde humano $E_{gt,i}^I, i = 1, \dots, n$, se realiza una comparación uno-a-uno con cada mapa de borde automático \mathcal{B}_t obtenido para la secuencia $\mathcal{B} = \{\mathcal{B}_1, \dots, \mathcal{B}_K\}$. Esto produce un valor F -de F_{ki} para cada comparación. Luego, se selecciona el mapa de borde automático \mathcal{B}^i de la secuencia con el mayor valor F para cada humano i :

$$\mathcal{B}^i = \mathcal{B}_{k^*}, \text{ where } k^* = \arg \max_{1 \leq k \leq K} F_{ti}. \quad (5.5)$$

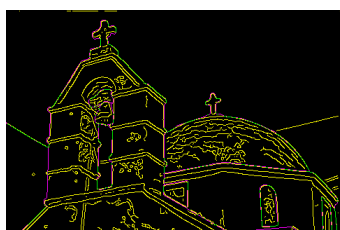
A partir de estos mapas de bordes automáticos que más se aproximan a cada referencia humana, se calcula una medida del rendimiento de la secuencia al capturar los diferentes niveles de detalle proporcionados por el conjunto de referencias humanas a través de un operador de agregación \mathcal{A} acorde a la semántica del resultado de las comparaciones uno-a-muchos. Del mismo modo que lo expresado en la sección anterior, se tiene que:

$$F_I(\mathcal{B}, \mathbf{E}_{gt}^I) = \mathcal{A}(F(\mathcal{B}^1, E_{gt,1}^I), \dots, F(\mathcal{B}^n, E_{gt,n}^I)). \quad (5.6)$$

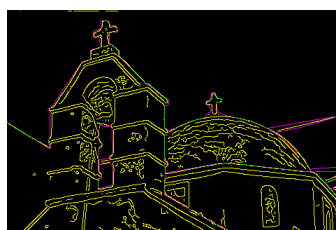
El proceso de evaluación para obtener una medida global F^* para cada configuración paramétrica sobre el conjunto de datos de entrenamiento, es igual al presentado en la sección anterior. Asimismo, para el conjunto de datos de prueba, se calcula solo el valor F^* de la configuración óptima obtenida en el conjunto de datos de entrenamiento.

Como se mencionó anteriormente, los operadores de agregación \mathcal{A} van acorde a la semántica de cómo las comparaciones son agregadas sobre las distintas referencias humanas. Existen distintos operadores de agregación (ver definición 2.6.3), pero los tres operadores descritos en la sección 2.6 y que son de interés en esta memoria se interpretan a continuación:

- $\mathcal{A} = \text{máx}$: uno de los más utilizados, y representa a la aproximación de Turing, y en este caso mide qué tan similar es un mapa de borde automático con respecto a alguna de las referencias humanas a .
- $\mathcal{A} = \text{media}$: se ajusta a la idea de medir cuán bien el mapa de bordes automático se asemeja en promedio al conjunto de referencias humanas.
- $\mathcal{A} = \text{mín}$: es más exigente y se ajusta a la idea de medir lo bien que un mapa de borde automático está simultáneamente cerca de todo el conjunto de referencias humanas.



(a) Imagen comparada con el mapa de bordes del humano 1



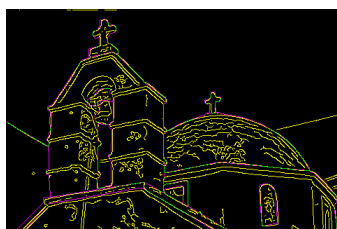
(b) Imagen comparada con el mapa de bordes del humano 2



(c) Imagen comparada con el mapa de bordes del humano 3



(d) Imagen comparada con el mapa de bordes del humano 4



(e) Imagen comparada con el mapa de bordes del humano 5

Figura 5.2: Comparación de la salida del detector Canny con cada referencia humana sobre la imagen de la *iglesia*, sin considerar un nivel de tolerancia para el cálculo de los verdaderos positivos (píxeles verdes), los verdaderos negativos (píxeles negros), los falsos positivos (píxeles amarillos) y los falsos negativos (píxeles morados)

Capítulo 6

Resultados computacionales

Ninguna investigación humana puede ser llamada ciencia real si no puede demostrarse matemáticamente.

Leonardo da Vinci

RESUMEN: En este capítulo, se presentan las experiencias computacionales resultantes de la evaluación del algoritmo ID&L. Específicamente, en la sección 6.1, se presentan los resultados obtenidos con la metodología de evaluación basada en bordes propuesta en el capítulo anterior. Por último, en la sección 6.2 se muestran los resultados obtenidos del algoritmo ID&L en la competición de detección de bordes realizada en el congreso EUSFLAT 2017.

6.1. Resultados computacionales sobre la evaluación del algoritmo ID&L

En esta sección se presentan las experiencias computacionales realizadas para evaluar la salida del algoritmo propuesto de detección de bordes. En más detalle, se muestran los resultados obtenidos al evaluar y comparar el algoritmo ID&L con los muy conocidos y utilizados detectores de bordes de Canny y Sobel, tomando como referencias mapas de bordes humanos.

La comparación se lleva a cabo sobre la base de datos *Berkeley Segmentation Dataset 500* (BSDS500) (Martin et al., 2004; Arbeláez et al., 2011). La BSDS500 contiene un conjunto de 500 (cada una con una resolución de 481×321 píxeles) imágenes naturales divididas en tres grupos, 200 imágenes para entrenamiento, 200 imágenes para pruebas y 100 imágenes para validación, de las cuales son utilizadas las imágenes de entrenamiento y de pruebas para la comparación.

En general, el proceso a seguir para la evaluación del algoritmo ID&L consiste en aplicar la metodología de evaluación a través de bordes descrita en la sección 5.1 a los algoritmos de detección de bordes a utilizar para la comparación, luego se aplica la metodología propuesta de evaluación de bordes 5.2 al algoritmo ID&L. Por último, se realiza un análisis estadístico de los resultados obtenidos en la comparación mediante la prueba de Wilcoxon para establecer diferencias significativas entre los métodos.

Esta sección se organiza de la siguiente manera: la subsección 6.1.1 describe la configuración de los algoritmos a utilizar en las experiencias computacionales. La subsección 6.1.2 muestra los resultados obtenidos con el conjunto de datos de entramiento para luego escoger la mejor configuración paramétrica de cada algoritmo. Por último, en la subsección 6.1.3 se presentan los resultados obtenidos sobre el conjunto de datos de prueba utilizando la configuración paramétrica obtenida en el apartado anterior.

6.1.1. Configuración experimental

Cada imagen de la BSDS500 viene acompañada de entre cuatro y siete mapas de bordes realizados por humanos y, que se utilizan como imágenes de referencia para la evaluación (Martin et al., 2001). La evaluación se lleva a cabo considerando cinco detectores de bordes: Sobel, Sobel-autoadaptado (A-Sobel, por sus siglas en inglés), Canny, Canny-autoadaptado (A-Canny, por sus siglas en inglés) y el Image Divide and Link (ID&L). Los métodos autoadaptados se refieren a las autoconfiguraciones de sus contrapartes (ver Rosin, 2001; Liu et al., 2013, respectivamente). La Tabla 6.1 engloba una descripción de cada uno de los detectores de bordes utilizados en este trabajo en términos de la estructura de desglose de Bezdek (Bezdek et al., 1998) explicada en la sección 2.5 y mostrando la configuración paramétrica utilizada en cada uno de ellos. Cada una de estas técnicas se ejecutará repetidamente sobre el conjunto de datos de entrenamiento de la BSDS500 para determinar su mejor configuración paramétrica.

Método	Sobel	A-Sobel	Canny	A-Canny
1. Acondicionamiento	Filtro Gaussiano con $\sigma_s \in \{0, 1\}$			
2. Extracción de características	Máscara de Sobel		Filtro de Canny con $\sigma_c \in \{2, 4\}$	
3. Mezcla/agregación	Magnitud euclidiana de gradiente y supresión no máxima (Rosenfeld y Thurston, 1971)			
4. Escalado	Umbralización $t \in [0, 1]$	Método de Rosin (Rosin, 2001)	Histéresis $t_1, t_2 \in [0, 1]$	Liu method (Liu et al., 2013)
Lista de parámetros	σ_s, t	σ_s	$\sigma_s, \sigma_c, t_1, t_2$	σ_s, σ_c

Tabla 6.1: Descomposición esquemática de los algoritmos de detección de bordes utilizados en la configuración experimental, de acuerdo con la estructura de desglose de Bezdek (Bezdek et al., 1998)

6.1. Resultados computacionales sobre la evaluación del algoritmo ID&L 79

Se puede observar en la Tabla 6.1, que los parámetros de acondicionamiento son $\sigma_s \in \{0, 1\}$, el motivo de utilizar estos valores es para ver el comportamiento que tienen los algoritmos al frente de a utilizar o no un filtro gaussiano. Asimismo, los valores escogidos para la extracción de características con el filtro de Canny fueron $\sigma_c \in \{2, 4\}$, los cuales son valores muy utilizados y que obtienen buenos resultados en dicho algoritmo. Por otro lado, tanto para la umbralización del detector de Sobel como para la histéresis para el detector de Canny, se consideraron valores dentro del rango $[0, 1]$ tomando un paso de valores de 0.01.

La Tabla 6.2 describe la configuración paramétrica del algoritmo ID&L. Se puede observar que el rango de algunos parámetros depende de la medida de disimilitud d y que este a su vez, toma valores en rangos específicos d sobre cada imagen. Específicamente, en este trabajo la disimilitud entre píxeles adyacentes se calcula con la distancia CIELab siguiendo la Ecuación 2.1 presentada en la sección 2.2, transformándose previamente la imagen RGB al espacio de color CIELab.

Método	Image-Divide-and-Link
1. Acondicionamiento	Filtro Gaussiano con $\sigma_s \in \{0, 1\}$
2. Número de aristas de corte	$K = 6$
3. Valores de umbrales	$\alpha = (\alpha_0, \dots, \alpha_K) \in [\underline{d}, \dots, \bar{d}]$
4. Parámetros de la red gruesa	$\alpha_\tau = 9\% \times \bar{d}; L = 3$
Lista de parámetros	$\sigma_s, K, \alpha, \alpha_\tau, L$

Tabla 6.2: Descomposición esquemática del algoritmo ID&L utilizado en la configuración experimental (Gómez et al., 2015a)

Además, se establece $K = 6$ siguiendo un enfoque de aprendizaje mixto con respecto a la elección de los umbrales $\alpha_1, \dots, \alpha_K$, basadas en una familia de configuraciones de umbrales preseleccionados, a partir de la cual se selecciona una única configuración posterior a un proceso de aprendizaje ejecutado sobre un conjunto de imágenes de entrenamiento. Específicamente, en esta memoria se considera una familia de ocho configuraciones diferentes para los valores de α , en donde cada uno de los distintos $K = 6$ valores intermedios, son escogidos entre los valores extremos \underline{d} y \bar{d} de acuerdo a las diferentes distribuciones. En la Figura 6.1 se muestran las distintas configuraciones consideradas, las cuales están enumeradas con números romanos del I al VIII. La primera distribución (I) considera K umbrales equidistantes uniformemente distribuidos en el rango $[\underline{d}, \bar{d}]$. Luego, se derivan otras cuatro combinaciones partiendo desde el punto medio $w = \frac{(\underline{d} + \bar{d})}{2}$ del intervalo $[\underline{d}, \bar{d}]$, de forma que tres umbrales estén uniformemente distribuidos a cada lado de w y los dos restantes en el otro lado (distribuciones II y III). Las configuraciones IV y V se obtienen de forma similar, salvo que ahora se colocan uniformemente cuatro umbrales a cada lado del punto medio w y el resto

en el otro. Finalmente, las tres distribuciones restantes, VI, VII y VIII, se obtienen al dividir el rango $[\underline{d}, \bar{d}]$ en tres partes iguales, de forma que se sitúa un umbral en el medio de dos de estas partes y los dos umbrales restantes se distribuyen uniformemente en la parte restante.

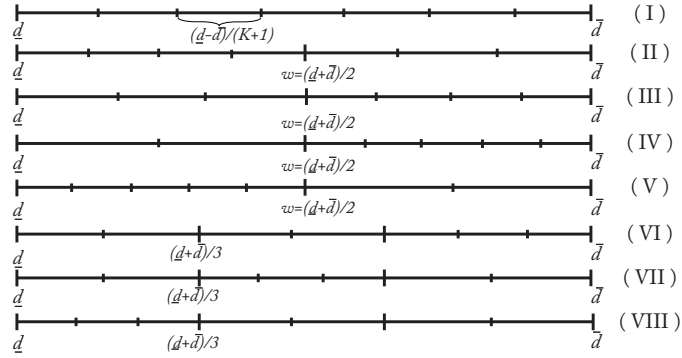


Figura 6.1: (I)-(VIII) Las distintas distribuciones consideradas para los valores α

Asimismo, aunque el algoritmo ID&L posee cuatro parámetros (K , α , α_τ , L), en este capítulo experimental sólo se considera variar el conjunto de umbrales $\alpha = (\alpha_0, \dots, \alpha_K)$ que guía el proceso de la construcción de los mapas de bordes automáticos. Los parámetros K , L y α_τ se dejan constantes en cada ejecución del algoritmo. Sus valores se han elegido de forma que se produzcan particiones suficientes y detalladas. En detalle, se observa que $\alpha_\tau = 0.09 \times \bar{d}$ proporciona un buen equilibrio, el cual permite reducir significativamente el rendimiento computacional del algoritmo mientras que al mismo tiempo no se contrae demasiado la red, por lo que algunos detalles aún logran ser identificados. Se ha escogido un valor de $K = 6$ debido a que proporciona un conjunto de niveles de detalle en la secuencia de mapas de bordes automáticos similares al número de referencias humanas disponibles para las diferentes imágenes de entrenamiento y prueba. Por lo tanto, con el fin de compactar la imagen, se ha elegido $L = 3$, evitando de esta manera demasiados detalles en las particiones. Además, un tamaño de máscara de 3×3 , es generalmente un tamaño estándar en la detección de bordes.

6.1.2. Resultados en el conjunto de entrenamiento

Cada uno de los métodos de detección de bordes y el algoritmo ID&L han sido ejecutados sobre el conjunto de entrenamiento de la BSDS500, siguiendo las configuraciones paramétricas establecidas en las Tablas 6.1 y 6.2, para determinar su configuración paramétrica óptima mediante las metodologías de evaluación presentadas en el capítulo 5. Específicamente, se calculan los valores de *precision* (Ecuación 5.1), *recall* (Ecuación 5.1) y F (Ecuación 5.3)

6.1. Resultados computacionales sobre la evaluación del algoritmo ID&L 81

para cada configuración paramétrica de los algoritmos utilizados para la comparación. La principal ventaja de usar estos valores de *precision* y *recall* (P&R), es que para cada algoritmo permite comparar la bondad de las particiones generadas por ellas y, también los resultados producidos por el mismo algoritmo usando diferentes parámetros de entrada.

Con el propósito de comparar el rendimiento del algoritmo ID&L con respecto a los otros métodos de forma cuantitativa, se han representado gráficamente tres curvas de P&R basadas en los tres operadores de agregación: el máximo, la media (aritmética) y el mínimo ($\mathcal{A} : max, media, min$ respectivamente), explicados en la sección 5.2. Las Figuras 6.2, 6.3, y 6.4 muestran las curvas P&R para los tres operadores de agregación considerados y, proporcionan las mejores configuraciones paramétricas resultantes obtenidas para cada método sobre el conjunto de datos de entrenamiento.

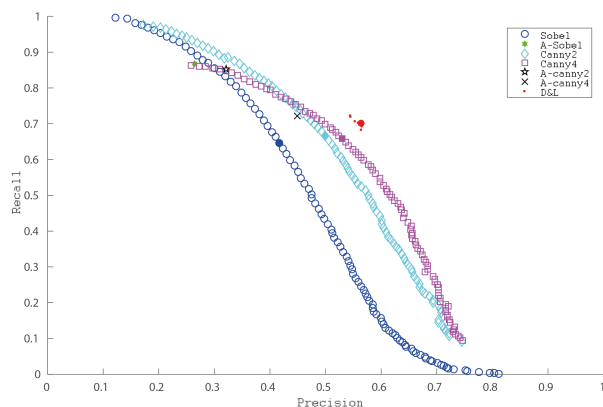


Figura 6.2: Curvas P&R de todos los algoritmos y los valores mayores de P&R (íconos resaltados) obtenidos usando $\mathcal{A} = max$ y $\sigma_s = 0$. Los mejores parámetros resultantes correspondientes a los íconos resaltados son: Sobel ($t = 0.30$), Canny2 ($\sigma_c = 2, t_1 = 0.156, t_2 = 0.39$), Canny4 ($\sigma_c = 4, t_1 = 0.124, t_2 = 0.31$), ID&L ($K = 6, \alpha = VII$ distribución, $\alpha_\tau = 0.09, L = 3$)

En más detalle, en la Tabla 6.3 se incluyen los mejores pares de valores de P&R, específicamente las medidas más altas F^* correspondientes a la mejor configuración paramétrica resultante en cada uno de los métodos. En esta Tabla, se observa que el algoritmo propuesto ID&L obtiene los mejores valores F^* para el máximo, media y mínimo en el conjunto de datos de entrenamiento para los dos casos de acondicionamiento $\sigma_s = 0$ y $\sigma_s = 1$.

6.1.3. Resultados en el conjunto de pruebas

Una vez que se ha determinado la configuración óptima de los parámetros para cada uno de los algoritmos y para cada uno de los operadores de agre-

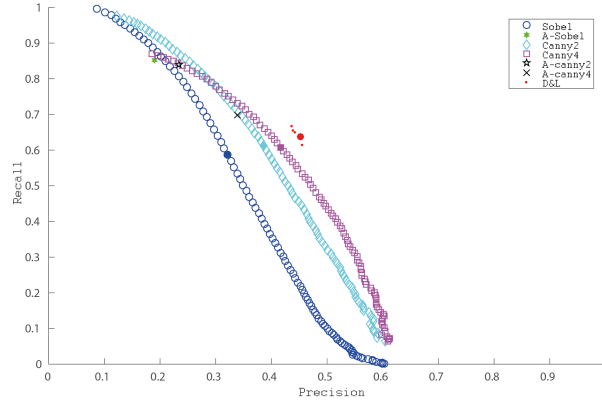


Figura 6.3: Curvas de P&R de todos los algoritmos y los valores mayores de P&R (íconos resaltados) obtenidos usando $\mathcal{A} = \text{media}$ y $\sigma_s = 0$. Los mejores parámetros resultantes correspondientes a los íconos resaltados son: Sobel ($t = 0.31$), Canny2($\sigma_c = 2, t_1 = 0.156, t_2 = 0.39$), Canny4($\sigma_c = 4, t_1 = 0.124, t_2 = 0.31$), ID&L ($K = 6, \alpha = VII$ distribución, $\alpha_\tau = 0.09, L = 3$)

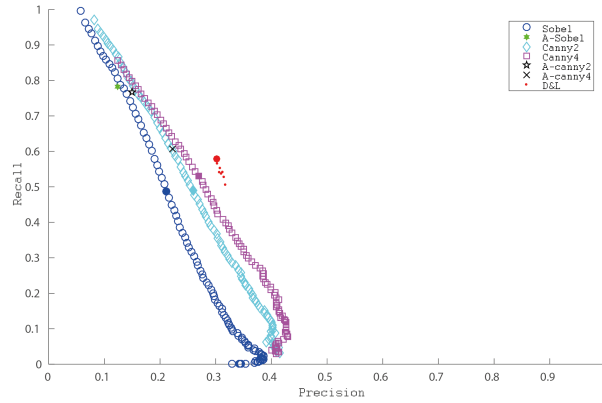


Figura 6.4: Curvas de P&R de todos los algoritmos y los valores mayores de P&R (íconos resaltados) obtenidos usando $\mathcal{A} = \text{min}$ y $\sigma_s = 0$. Los mejores parámetros resultantes correspondientes a los íconos resaltados son: Sobel ($t = 0.31$), Canny2($\sigma_c = 2, t_1 = 0.156, t_2 = 0.39$), Canny4($\sigma_c = 4, t_1 = 0.112, t_2 = 0.28$), ID&L ($K = 6, \alpha = VII$ distribución, $\alpha_\tau = 0.09, L = 3$)

gación, se ejecuta cada algoritmo con sus parámetros óptimos en las 200 imágenes del conjunto de datos de pruebas de BSDS500. Los resultados obtenidos en este conjunto de imágenes de prueba se muestran en la Tabla 6.4.

Con estos resultados, se puede resumir que el algoritmo ID&L obtiene un mayor rendimiento respecto a los otros algoritmos de detectores de bordes considerados. Los resultados de la prueba de Wilcoxon se muestran en las

6.2. Competición de Detección de Bordes en el congreso EUSFLAT 2017 83

Acondicionamiento	Método	<i>max</i>	<i>media</i>	<i>min</i>
$\sigma_s = 0$	Sobel	0.4732	0.3823	0.2588
	A-Sobel	0.3909	0.3017	0.2044
	Canny ($\sigma_c = 2$)	0.5442	0.4466	0.3028
	Canny ($\sigma_c = 4$)	0.5670	0.4726	0.3247
	A-Canny ($\sigma_c = 2$)	0.4504	0.3549	0.2411
	A-Canny ($\sigma_c = 4$)	0.5360	0.4424	0.3085
	ID&L	0.5964	0.5037	0.3554
$\sigma_s = 1$	Sobel	0.5078	0.4138	0.2816
	A-Sobel	0.4334	0.3395	0.2299
	Canny ($\sigma_c = 2$)	0.5520	0.454	0.308
	Canny ($\sigma_c = 4$)	0.5663	0.4729	0.3252
	A-Canny ($\sigma_c = 2$)	0.4711	0.3747	0.2565
	A-Canny ($\sigma_c = 4$)	0.5363	0.4448	0.3100
	ID&L	0.6107	0.5166	0.3656

Tabla 6.3: Resultados globales para cada uno de los algoritmos sobre el conjunto de datos de entrenamiento de la BSDS500

Tablas 6.5, 6.6, 6.7, con las cuales se puede concluir que el algoritmo propuesto ID&L es significativamente mejor que los otros detectores de bordes considerados en este trabajo, sobre los datos de la BSDS500.

Es importante decir que, la base de datos BSDS500 está enfocada a problemas de segmentación de imágenes y no a los problemas de detección de bordes. Sin embargo, esta base de datos ha sido seleccionada en este trabajo con la finalidad de evaluar el algoritmo ID&L que está basado en un enfoque de agrupación jerárquica de grafos y el cual, ha sido la base para obtener una segmentación borrosa de imágenes. Por ende, era importante evaluar el algoritmo ID&L en una base de datos enfocada a la segmentación, pero en trabajos futuros se evaluará el algoritmo en otras bases de datos orientadas a la detección de bordes.

En cuanto al tiempo de ejecución, el CPU no tardó más de un minuto para cada mapa de bordes automático generado por el algoritmo ID&L. Estos tiempos se midieron en un Core(TM)2 Duo de 3.00GHz con 3GB de RAM.

6.2. Competición de Detección de Bordes en el congreso EUSFLAT 2017

En el congreso EUSFLAT 2017 celebrado en Varsovia, Polonia, se celebró una competición de detección de bordes que consistía en realizar esta tarea sobre 30 imágenes correspondientes a la base de datos ED17 utilizando cualquier detector de bordes combinado con métodos de inteligencia computacional¹.

¹EUSFLAT 2017 Competition on Edge Detection, organizado por EUSFLAT Working Group on Soft Computing Methods in Image Processing (<http://irafm.osu.cz/edge2017/main.php>)

Acondicionamiento	Método	<i>max</i>	<i>media</i>	<i>min</i>
$\sigma_s = 0$	Sobel	0.4790	0.3930	0.2836
	A-Sobel	0.3990	0.3148	0.2213
	Canny ($\sigma_c = 2$)	0.5366	0.4488	0.3244
	Canny ($\sigma_c = 4$)	0.5513	0.4705	0.3474
	A-Canny ($\sigma_c = 2$)	0.4609	0.3629	0.2607
	A-Canny ($\sigma_c = 4$)	0.5359	0.4517	0.3316
	ID&L	0.5819	0.4960	0.3761
$\sigma_s = 1$	Sobel	0.5098	0.4212	0.3049
	A-Sobel	0.4455	0.3570	0.2521
	Canny ($\sigma_c = 2$)	0.5435	0.4563	0.3319
	Canny ($\sigma_c = 4$)	0.5514	0.4709	0.3466
	A-Canny ($\sigma_c = 2$)	0.4803	0.3865	0.2754
	A-Canny ($\sigma_c = 4$)	0.5366	0.4531	0.3327
	ID&L	0.5926	0.5067	0.3860

Tabla 6.4: Resultados globales para cada uno de los métodos sobre el conjunto de datos de prueba de la BSDS500

Método	Sobel	A-Sobel	Canny ($\sigma_c = 2$)	Canny ($\sigma_c = 4$)	A-Canny ($\sigma_c = 2$)	A-Canny $\sigma_c = 4$)
Rangos positivos	14	7	46	61	22	55
Rangos negativos	186	193	154	139	178	145
Z	-11.456	-12.185	-8.059	-5.448	-11.279	-6.716
p-value	< .001	< .001	< .001	< .001	< .001	< .001

Tabla 6.5: Prueba de Wilcoxon de las comparaciones del algoritmo ID&L con todos los algoritmos de detectores de bordes considerados para $\mathcal{A} = \text{max}$ con $\sigma_s = 0$

Cada imagen de la base de datos ED17 venía acompañada por una imagen *ground truth* en donde se distinguen los bordes verdaderos, como los límites de los objetos y otras discontinuidades superficiales (bordes geométricos) y las áreas de borde falso que surgen al cambiar la iluminación (sombras, reflejos, etc.). Se excluyen los bordes creados por texturas. En la Figura 6.5 se muestra una imagen original de ejemplo proporcionada por la organización con su respectiva imagen de *ground truth*. Cabe destacar que, una de las complicaciones que presentaban estas imágenes de *ground truth*, era que algunos bordes tenían un grosor de más de un píxel.

El método de evaluación mide la calidad Q del método de detección de bordes respecto a la imagen de referencia, siguiendo el enfoque de (Boaventura y Gonzaga, 2009), que se resume en la siguiente expresión:

$$Q = \sqrt{\left(\frac{TP}{M} - 1\right)^2 + \left(\frac{FN}{M}\right)^2 + \left(\frac{FP}{M}\right)^2 + \left(\frac{F}{M} - 1\right)^2} \quad (6.1)$$

6.2. Competición de Detección de Bordes en el congreso EUSFLAT 2017 85

Método	Sobel	A-Sobel	Canny ($\sigma_c = 2$)	Canny ($\sigma_c = 4$)	A-Canny ($\sigma_c = 2$)	A-Canny $\sigma_c = 4$)
Rangos positivos	12	2	37	71	10	52
Rangos negativos	188	198	163	129	190	148
Z	-11.729	-12.235	-9.082	-4.908	-11.914	-7.158
p-value	< .001	< .001	< .001	< .001	< .001	< .001

Tabla 6.6: Prueba de Wilcoxon de las comparaciones del algoritmo ID&L con todos los algoritmos de detectores de bordes considerados para $\mathcal{A} = \text{media}$ con $\sigma_s = 0$

Método	Sobel	A-Sobel	Canny ($\sigma_c = 2$)	Canny ($\sigma_c = 4$)	A-Canny ($\sigma_c = 2$)	A-Canny $\sigma_c = 4$)
Rangos positivos	11	7	32	62	21	56
Rangos negativos	189	193	168	138	179	144
Z	-11.899	-12.152	-9.864	-5.653	-11.463	-7.248
p-value	< .001	< .001	< .001	< .001	< .001	< .001

Tabla 6.7: Prueba de Wilcoxon de las comparaciones del algoritmo ID&L con todos los algoritmos de detectores de bordes considerados para $\mathcal{A} = \text{min}$ con $\sigma_s = 0$

donde

$$M = \max\{TP + FN, TP + FP\}$$

$$F = \sum_{i=1}^{TP+FP} \frac{1}{1 + \frac{1}{9}d^2(i)}$$

y TP se refiere a los verdaderos-positivos, FP a los falsos positivos, TN a los verdaderos negativos, FN a los falsos negativos y $d(i)$ a la distancia entre los píxeles del i -ésimo borde del resultado propuesto y el *ground truth*.

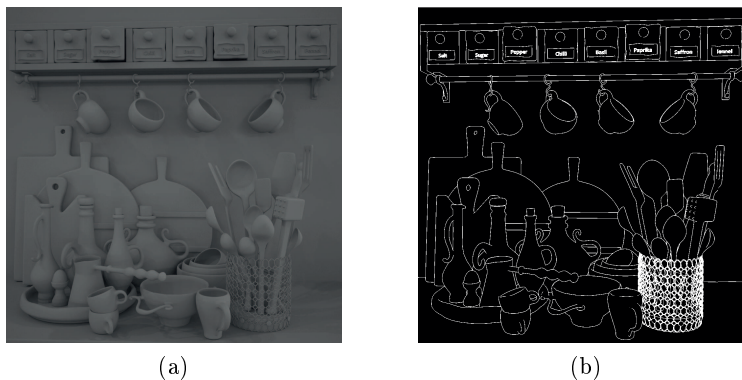


Figura 6.5: (a) Imagen original y (b) *ground truth*.

6.2.1. Resultados del ID&L

Como se ha visto en el capítulo 4, una ventaja del algoritmo ID&L es que proporciona una secuencia de particiones. De esta forma, el usuario tiene la opción de elegir la partición de la secuencia que sea más informativa para su interés, contrariamente a otros algoritmos que proporcionan una imagen de partición única creada a partir de cualquier restricción y que la salida puede no ser informativa. Esta ventaja del ID&L para elegir la partición necesaria, permite que el algoritmo se compare con otros algoritmos no jerárquicos. Particularmente, si las regiones proporcionadas por el algoritmo ID&L están representadas por un mapa de bordes binario en donde los bordes corresponden a las regiones más grandes, entonces el algoritmo puede ser comparado con otros algoritmos de detección de bordes.

El algoritmo ID&L fue aplicado sobre las 30 imágenes de la base de datos ED17. El procedimiento a seguir fue el siguiente:

Preprocesamiento: primero se convierte la imagen original RGB en una imagen de intensidad de escala de grises y luego, se ajustan los valores de intensidad de la imagen ajustando el contraste de modo que, el 1 % de los píxeles esté saturado en intensidades bajas y altas (ver Figura 6.6).

Procesamiento: una vez realizado el preprocesamiento, se procede a ejecutar el algoritmo ID&L. La medida de disimilitud considerada fue la distancia Euclídea (Ecuación 2.1) y un conjunto de umbrales de tamaño $K = 9$.

Postprocesamiento: por último, se elige la partición más representativa ($t = 5$) de la secuencia obtenida para cada imagen como resultado de la mejor partición obtenida en las imágenes de los ejemplos (ver Figura 6.7).

En resumen, el algoritmo compitió con otros 7 otros algoritmos y obtuvo la segunda mejor puntuación (Tabla 6.8).

6.2. Competición de Detección de Bordes en el congreso EUSFLAT 2017 87



Figura 6.6: Imagen resultante al ajustar los valores de intensidad de la imagen de ejemplo 6.5a antes de aplicar el ID&L

Método	Q
1. Fuzzy image gradient approximation with thresholding	0.809264
2. Image Divide and Link (ID&L)	1.07878
3. Image contours detection with deep features and SVM	1.08313
4. Fuzzy image gradient approximation with thresholding and non-maxima suppression	1.08648
5. Edges tracked from interest points	1.09399
6. A New Edge Detector Based on SMOTE and Logistic Regression	1.10983
7. Edge Detection Based on the Fusion of Multiscale Anisotropic Edge Strength Measurements	1.11018
8. Edge detection based on ordered directionally monotone functions	1.43447

Tabla 6.8: Resultados de la competición de Edge Detection, EUSFLAT 2017

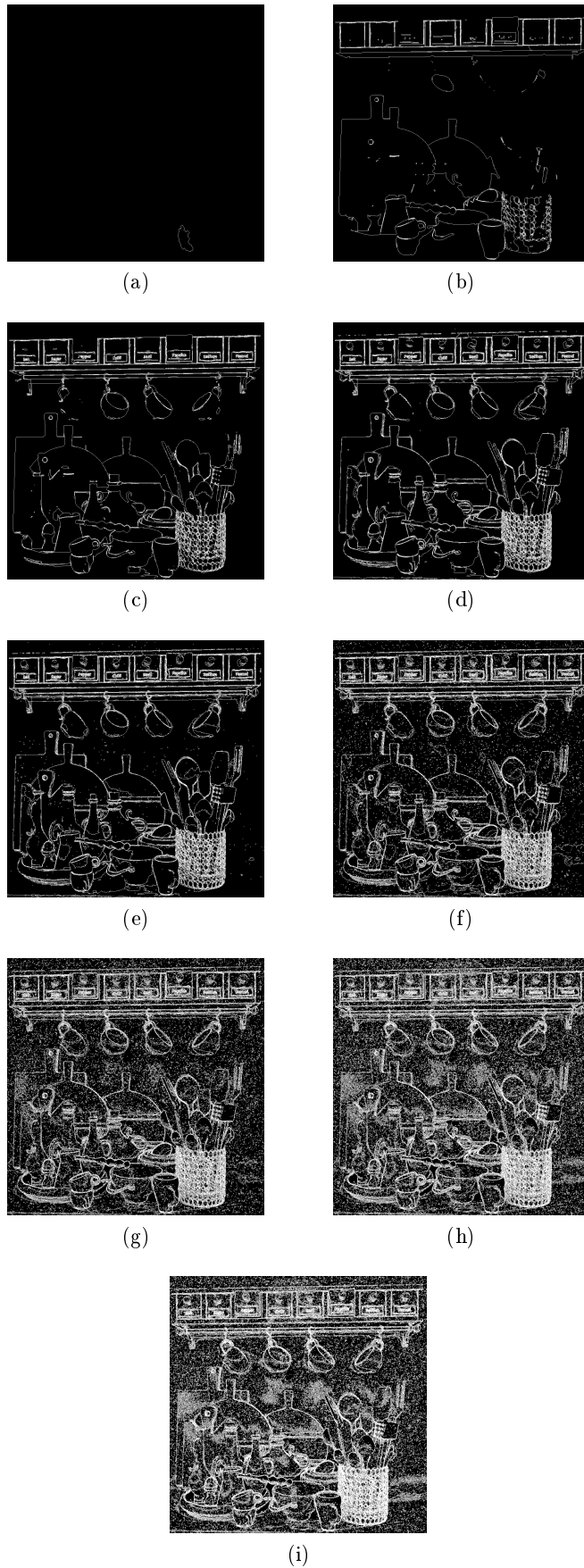


Figura 6.7: Salida del ID&L

Capítulo 7

Conclusiones y trabajo futuro

*Las ciencias y la literatura llevan en sí la
recompensa de los trabajos y vigilias que
se les consagran.*

Andrés Bello

RESUMEN: En el primer apartado de este capítulo, se mencionan las conclusiones obtenidas al realizar el estudio presentado en esta memoria (sección 7.1). Asimismo, en la sección 7.2, se exhiben las contribuciones realizadas durante el desarrollo de esta memoria. Finalmente, en la sección 7.3 se presentan las futuras investigaciones que se pueden realizar como continuación de este trabajo.

7.1. Conclusiones

La segmentación de imágenes es una tarea esencial e importante en el tratamiento de imágenes de bajo nivel, además, puede ser aplicada en muchas áreas. Asimismo, la segmentación jerárquica de imágenes puede ser vista como una extensión de la segmentación de imágenes, con la que se obtiene una secuencia jerárquica de particiones. De este modo, la segmentación jerárquica provee más libertad al usuario debido a que puede escoger la partición de la secuencia jerárquica que más se adapte a sus objetivos. El planteamiento formal de segmentación jerárquica en esta memoria se realizó a través de la red de una imagen. Para ello, primero se definió la segmentación de imágenes a través de las aristas de un grafo, siendo una definición equivalente al concepto de segmentación de imágenes a través de los nodos de un grafo que ya existía en la literatura.

A partir de estas nuevas definiciones de segmentación de imagen basada en las aristas de una red y de segmentación jerárquica de imágenes, se

procedió a presentar formalmente el concepto de segmentación borrosa de imágenes basada en aristas. Cabe destacar que, el concepto de segmentación borrosa basado en aristas es más fácil formalizarlo que el homólogo basado en nodos, aunque actualmente se está trabajando en ello. Este concepto de segmentación borrosa basado en aristas tampoco estaba formalmente definido en la literatura revisada. Por lo que formalizar este nuevo concepto abre un nuevo campo en las técnicas de tratamiento de imágenes. Una de las ventajas de este concepto propuesto, es que al visualizar su salida no produce una imagen binaria de fondo y objeto de las imágenes como suelen hacer los algoritmos de segmentación más conocidos, sino que produce una secuencia de bordes con gradación de intensidad de acuerdo a la disimilitud de cada objeto con los de su alrededor. Estas gradaciones en los bordes aportan más información en comparación con las salidas convencionales binarias. Igualmente, estas gradaciones de color en los bordes se asemejan más cuando un humano segmenta una imagen, en donde puede resaltar más aquellos objetos que ve con más claridad en la imagen. Esta visualización de segmentación borrosa se basó en una metodología propuesta en esta memoria que consistía en agregar la salida de un algoritmo de segmentación jerárquica de redes. Sin embargo, por problemas de representación y evaluación de una salida de segmentación jerárquica en la parte experimental, nació la necesidad de transformar un algoritmo de segmentación jerárquica en redes (Zarrazola, 2013) a uno de detección de bordes. De este modo, también se desarrolló en esta memoria un algoritmo denominado Image Divide and Link (ID&L) que detectara bordes en las regiones de una partición jerárquica de redes.

El algoritmo de detección de bordes ID&L tiene varias ventajas, una de ellas es que, al modelar una imagen como una red, el algoritmo es capaz de trabajar con distintos tipos de imágenes. En esta memoria las imágenes en color RGB que se utilizaron de entrada para el algoritmo ID&L, fueron transformadas al espacio de color CIELab el cual posee una información similar a cómo el humano percibe los colores. Adicionalmente, las imágenes a color poseen mayor información respecto a las imágenes en escala de grises con las que trabajan otros algoritmos de tratamiento de imágenes, proporcionando así otra ventaja comparado con dichos algoritmos.

Por otro lado, en esta memoria se propuso y se ejecutó computacionalmente una nueva metodología de evaluación de bordes, esto con el propósito de poder comparar las salidas del algoritmo ID&L con otros algoritmos jerárquicos o no jerárquicos de detección de bordes sobre la BSDS500. El resultado computacional obtenido fue que las salidas generadas con el algoritmo ID&L fueron significativamente más parecidas en semejanza, promedio y simultaneidad a los mapas de bordes humanos que las salidas obtenidas con los muy conocidos detectores de bordes de Canny y Sobel.

Igualmente, el algoritmo ID&L ha obtenido un buen rendimiento al quedar en segundo lugar respecto a otros ocho nuevos algoritmos de detección

de bordes en la Competición de Detección de Bordes del congreso EUSFLAT 2017. Esta competición utilizó otra base de datos de imágenes (ED17) con otro tipo de coloración y, al haber tenido un buen resultado el algoritmo ID&L, se certifica en cierto modo la robustez del mismo.

Además, en esta memoria también se presentó una nueva clasificación de algunos problemas de tratamiento de imágenes basada en la salida que cada uno de ellos consigue. Esto con la finalidad de diferenciar dichos problemas conceptualmente debido a que son frecuentemente confundidos en la literatura.

7.2. Contribuciones relacionadas con este trabajo

En esta sección se presentan las contribuciones realizadas producto del desarrollo de esta memoria. Específicamente, en la subsección 7.2.1 se listan los artículos publicados en revistas de impacto. Seguidamente, en la subsección 7.2.2 se muestran los artículos publicados en otras revistas. Por último, en la subsección 7.2.3 se presentan las contribuciones hechas en congresos.

7.2.1. Artículos en revistas de impacto

- D. Gómez, J. Yáñez, C. Guada, J.T. Rodríguez, J. Montero, E. Zarrazola. Fuzzy image segmentation based upon hierarchical clustering. *Knowledge-Based Systems* (2015), Vol. 87, pp. 26-37.
- C. Guada, D. Gómez, J.T. Rodríguez, J. Yáñez, J. Montero. Classifying image analysis techniques from their output. *International Journal of Computational Intelligence Systems* (2016). Vol. 9, Sup. 1, pp. 43-68.
- C. Guada, E. Zarrazola, J. Yáñez, J.T. Rodríguez, D. Gómez, J. Montero. A novel edge detection algorithm based on a hierarchical graph-partition approach. *Journal of Intelligent and Fuzzy Systems* (2018), Vol. 34, No. 3, pp. 1875-1892.

7.2.2. Artículos en otras revistas

- C. Guada, D. Gómez, J.T. Rodríguez, J. Yáñez, J. Montero. A fuzzy edge-based image segmentation approach. In *Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology. Advances in Intelligent Systems Research* (89). Atlantis Press, pp. 1216-1222. ISBN 978-94-62520-77-6.
- C. Guada, D. Gómez, J.T. Rodríguez, J. Yáñez, J. Montero. Fuzzy image segmentation base on the hierarchical Divide and Link clustering algorithm. In *Intelligent Systems and Knowledge Engineering (ISKE)*,

2015 10th International Conference on. IEEE, pp. 12-17. ISBN 978-1-4673-9322-5.

- C. Guada, y D. Gomez, J.T. Rodríguez, J. Yáñez, J. Montero. Segmentación Borrosa de Imágenes basada en un Algoritmo de Segmentación Jerárquica. En actas de la XVI Conferencia de la Asociación Española para la Inteligencia Artificial. CAEPIA-15, Albacete, pp. 519-528. ISBN 978-84-608-4099-2.
- J.T. Rodríguez, C. Guada, D. Gomez, J. Yáñez, J. Montero. A methodology for hierarchical image segmentation evaluation. In Information Processing and Management of Uncertainty in Knowledge-Based Systems (2016). Comunicación en Computer and Information Science, I (610). Springer, pp. 635-647. ISBN 978-3-319-40596-4.
- C. Guada, D. Gómez, J.T. Rodríguez, J. Yáñez, J. Montero. Graph approach in image segmentation. In Proceedings of the Conference of the European Society for Fuzzy Logic and Technology 2017. Advances in Fuzzy Logic and Technology 2017, Vol. 642. Springer International Publishing (2018).
- P. Flores-Vidal, D. Gómez, P. Olaso, C. Guada A new edge detection approach based on fuzzy segments clustering. In Proceedings of the Conference of the European Society for Fuzzy Logic and Technology 2017. Advances in Fuzzy Logic and Technology 2017, Vol. 642. Springer International Publishing (2018).

7.2.3. Contribuciones a congresos

- C. Guada, J. Montero D. Gómez, J.T. Rodríguez. Aprendizaje automático para el tratamiento de imágenes. SEMATICA-14. Barcelona (España), noviembre 20-21, 2014.
- C. Guada, D Gómez, J.T. Rodríguez, J. Yáñez, J. Montero. Segmentación borrosa de imágenes basada en un algoritmo de segmentación jerárquica. Simposio LFSC dentro de CAEPIA-15. Albacete,(España), noviembre, 9-12 2015.
- C. Guada, D. Gómez, J. Montero, J.T. Rodríguez, J. Yáñez. Segmentación borrosa de imágenes basadas en un algoritmo de segmentación jerárquica. FuzzyMAD-15. Madrid (España), diciembre 15, 2015.
- C. Guada, J.T. Rodríguez, D. Gómez, J. Yáñez, J. Montero. Una metodología para evaluar algoritmos de segmentación jerárquica. ESTYLF-16. San Sebastián (España), mayo 25-27, 2016.

- C. Guada, J.T. Rodríguez, D. Gomez, J. Yáñez, J. Montero. Evaluación de bordes en segmentación jerárquica de imágenes. LODISCO'16 (CAEPIA-16-CEDI-16). Salamanca (España), septiembre 14-18, 2016.
- C. Guada, A. Zarrazola, J. Yáñez, J.T. Rodríguez, D. Gómez, J. Montero. Algoritmo de clustering jerárquico basado en grafos para la detección de bordes. FuzzyMAD-16. Madrid, diciembre 16, 2016.
- C. Guada, D. Gómez, J.T. Rodríguez, J. Yáñez, J. Montero. On the fuzzy image segmentation concept. Joint IFSA-EUSFLAT-15. Gijón (España), junio 30-Julio 3, 2015.
- C. Guada, D. Gómez, J.T. Rodríguez, J. Yáñez, J. Montero. Fuzzy image segmentation base on the hierarchical Divide and Link clustering algorithm. Intelligent Systems and Knowledge Engineering (ISKE 15). Taipei (Taiwan), noviembre 24-27, 2015.
- J.T. Rodríguez, C. Guada, D. Gómez, J. Yáñez, J. Montero. A methodology for hierarchical image segmentation evaluation. Int. Conf. Information Proc. & Manag. of Uncertainty in KBS, IPMU-16. Eindhoven (Holanda), junio 20-24, 2016.
- C. Guada. Computational Intelligence in Image Segmentation. Doctoral Consortium of Soft Computing and Decision Making en el Congreso de Fuzzy Logic on Uncertainty Modelling in Knowledge Engineering and Decision Making, FLINS-16. Roubaix (Francia), agosto 24-26, 2016.
- C. Guada, D. Gómez, J.T. Rodríguez, J. Yáñez, J. Montero. A Hierarchical Graph-based Algorithm for Image Segmentation. ISCAMI. Malenovice (República Checa), junio 8-11, 2017.
- C. Guada, D. Gómez, J.T. Rodríguez, J. Yáñez, J. Montero. A Graph Approach in Image Segmentation. EUSFLAT-2017. Varsovia (Polonia), septiembre 11-15, 2017.
- C. Guada, D. Gómez, J.T. Rodríguez, J. Montero. Fuzzy image segmentation concept and its visualization. FSTA-2018. Liptovský Ján (Eslovaquia), enero 28-febrero 2, 2018.
- C. Franco, C. Guada, J. T. Rodríguez, D. Gómez, J. Montero, J. Nielsen, J. Rasmussen. Automatic Detection of Thistle-Weeds in Cereal Crops from Aerial RGB Images. IPMU-2018. Cádiz (España), junio 11-15, 2018.

7.3. Futuras líneas de investigación

El trabajo presentado en esta memoria nació como continuación de trabajos anteriores relacionados con la segmentación jerárquica en redes, sin embargo, ha dejado varias investigaciones abiertas. En general, el tratamiento de imágenes y específicamente la segmentación borrosa de imágenes, es un tema amplio y novedoso el cual requiere aún de mucha más investigación y extensión. Además, al haber planteado un nuevo algoritmo de detección de bordes basado en particiones jerárquicas, se puede extender mucho más su utilidad, como por ejemplo a imágenes más complejas como son las hiperespectrales. Asimismo, las salidas del algoritmo pueden ser mejoradas con un pre y post procesamiento. A continuación, se listan algunas de las posibles futuras líneas de investigación:

- Formalizar el concepto de segmentación borrosa de imágenes basado en los nodos de una red.
- A pesar de que el algoritmo ID&L es un algoritmo no supervisado, es posible automatizar aún más la selección de los umbrales para la realización de las particiones acorde a las características de la imagen en estudio. Es decir, en este trabajo no se propone cómo elegir automáticamente los umbrales, sino que se ha escogido la mejor distribución de umbrales de entre ocho propuestas sobre un conjunto de datos de entrenamiento. Una posible salida a la selección de la mejor distribución de umbrales sin contar con un conjunto de datos de entrenamiento es haciendo uso de las *funciones de disimilitud restringida* (Bustince et al., 2006a). Por lo que esta selección de umbrales es un problema abierto que se tratará en un futuro.
- Las imágenes correspondientes a las salidas del algoritmo ID&L se pueden mejorar al aplicar un preprocesamiento incluido internamente en el algoritmo, es decir, se puede emplear un filtro durante el cálculo de la disimilitud entre los píxeles adyacentes de forma que se comporte como un preprocesamiento paralelo a la ejecución del algoritmo.
- Asimismo, se puede aplicar un postprocesamiento para mejorar las salidas que genere el algoritmo ID&L con otro que haga una clasificación sobre los bordes malos o buenos con el fin de facilitar una imagen más limpia como salida final del algoritmo.
- La comparación del algoritmo ID&L con los algoritmos de detección de bordes utilizados en este trabajo puede ser realizada usando otra base de datos que esté más enfocada a los algoritmos de detección de bordes como la de la Universidad del Sur de Florida (?), la cual omite bordes interiores de los objetos.

-
- Además, se puede aplicar el algoritmo ID&L a otro tipo de imágenes digitales no utilizado anteriormente, por ejemplo las imágenes hiperespectrales, debido a la ventaja que posee de asociar una imagen a un grafo y, así se podría conseguir una aplicación interesante.
 - Se puede definir el nuevo concepto de detección jerárquica de bordes.
 - El método de evaluación de bordes jerárquico propuesto en esta memoria fue planteado como un inicio para la evaluación de la segmentación borrosa de imágenes. Por lo que se puede extender el método de evaluación jerárquica presentado anteriormente, a la evaluación de la segmentación borrosa de imágenes mediante los bordes de los objetos.

Bibliografía

*Comienza tu obra; comenzar es haber
hecho la mitad; comienza de nuevo, y la
obra quedará terminada.*

Jacinto Benavente

- ADAMS, R. y BISCHOF, L. Seeded Region Growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16(6), páginas 641–647, 1994.
- AHONEN, T., HADID, A. y PIETIKÄINEN, M. Face recognition with local binary patterns. En *European conference on computer vision*, páginas 469–481. Springer, 2004.
- AL-KOFAHI, Y., LASSOUED, W., LEE, W. y ROYSAM, B. Improved automatic detection and segmentation of cell nuclei in histopathology images. *IEEE Transactions on Biomedical Engineering*, vol. 57(4), páginas 841–852, 2010.
- ALBERS, J. *Interaction of color*. Yale University Press, 2013.
- ARBELÁEZ, P., MAIRE, M., FOWLKES, C. y MALIK, J. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, vol. 33(5), páginas 898–916, 2011.
- BADDELEY, A. J. An error metric for binary images. *Robust computer vision*, vol. 5978, 1992.
- BARRENECHEA, E., BUSTINCE, H., DE BAETS, B. y LÓPEZ-MOLINA, C. Construction of interval-valued fuzzy relations with application to the generation of fuzzy edge images. *IEEE Transactions on fuzzy systems*, vol. 19(5), páginas 819–830, 2011.
- BASAVAPRASAD, B. y RAVINDRA, H. A survey on traditional and Graph Theoretical Techniques for Image Segmentation. *IJCA Proceedings on National Conference on Recent Advances in Information Technology*, vol. NCRAIT, páginas 38–46, 2014.

- BELIAKOV, G. Definition of general aggregation operators through similarity relations. *Fuzzy sets and Systems*, vol. 114(3), páginas 437–453, 2000.
- BELIAKOV, G., PRADERA, A. y CALVO, T. *Aggregation functions: A guide for practitioners*, vol. 221. Springer, 2007.
- BEUCHER, S. y MEYER, F. The morphological approach to segmentation: the watershed transformation. *Optical Engineering-New York-Marcel Dekker Incorporated-*, vol. 34, páginas 433–433, 1992.
- BEZDEK, J. *Pattern Recognition with Fuzzy Objective Function Algorithms*. USA, 1981.
- BEZDEK, J., EHRLICH, R. y FULL, W. FCM: The fuzzy c-means clustering algorithm. *Computers and Geosciences*, vol. 10(2-3), páginas 191–203, 1984.
- BEZDEK, J. C., CHANDRASEKHAR, R. y ATTIKOUZEL, Y. A geometric approach to edge detection. *IEEE Transactions on Fuzzy Systems*, vol. 6(1), páginas 52–75, 1998.
- BIBILONI, P., GONZÁLEZ-HIDALGO, M. y MASSANET, S. A survey on curvilinear object segmentation in multiple applications. *Pattern Recognition*, vol. 60, páginas 949–970, 2016.
- BLOCH, I. Fuzzy connectivity and mathematical morphology. *Pattern Recognition Letters*, vol. 14(6), páginas 483–488, 1993.
- BOAVENTURA, I. y GONZAGA, A. Edge detection in digital images using fuzzy numbers. vol. 2, páginas 1–12, 2009.
- BONDY, J. A., MURTY, U. S. R. ET AL. *Graph theory with applications*, vol. 290. Citeseer, 1976.
- BOYKOV, Y. y FUNKA-LEA, G. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, vol. 70(2), páginas 109–131, 2006.
- BOYKOV, Y. y KOLMOGOROV, V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26(9), páginas 1124–1137, 2004.
- BOYKOV, Y. y VEKSLER, O. *Graph Cuts in Vision and Graphics: Theories and Applications*, capítulo 5, páginas 79–96. Handbook of Mathematical Models in Computer Vision, 2005.
- BRICE, C. y FENNEMA, C. Scene analysis using regions. *Artificial Intelligence*, vol. 1(3-4), páginas 205–226, 1970.

- BRUCE, J., BALCH, T. y VELOSO, M. Fast and inexpensive color image segmentation for interactive robots. En *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 3, páginas 2061–2066. IEEE, 2000.
- BUSTINCE, H., BARRENECHEA, E. y PAGOLA, M. Restricted equivalence functions. *Fuzzy Sets and Systems*, vol. 157(17), páginas 2333–2346, 2006a.
- BUSTINCE, H., BARRENECHEA, E., PAGOLA, M. y FERNÁNDEZ, J. Interval-valued fuzzy sets constructed from matrices: Application to edge detection. *Fuzzy Sets and Systems*, vol. 160(13), páginas 1819–1840, 2009.
- BUSTINCE, H., BARRENECHEA, E., PAGOLA, M., FERNÁNDEZ, J., XU, Z., BEDREGAL, B., MONTERO, J., HAGRAS, H., HERRERA, F. y DE BAETS, B. A historical account of types of fuzzy sets and their relationships. *IEEE Transactions on Fuzzy Systems*, vol. 24(1), páginas 179–194, 2016.
- BUSTINCE, H., MOHEDANO, V., BARRENECHEA, E. y PAGOLA, M. Definition and construction of fuzzy di-subsethood measures. *Information Sciences*, vol. 176(21), páginas 3190–3231, 2006b.
- BUSTINCE, H., MONTERO, J., BARRENECHEA, E. y PAGOLA, M. Semiautoduality in a restricted family of aggregation operators. *Fuzzy Sets and Systems*, vol. 158(12), páginas 1360–1377, 2007.
- BUSTINCE, H., PAGOLA, M., BARRENECHEA, E., FERNÁNDEZ, J., MELO-PINTO, P., COUTO, P., TIZHOOSH, H. R. y MONTERO, J. Ignorance functions. an application to the calculation of the threshold in prostate ultrasound images. *Fuzzy sets and Systems*, vol. 161(1), páginas 20–36, 2010.
- CALVO, T., KOLESÁROVÁ, A., KOMORNÍKOVÁ, M. y MESIAR, R. Aggregation operators: Basic concepts, issues and properties. *Aggregation operators: New trends and applications. Studies in fuzziness and soft computing*, vol. 97, páginas 3–106, 2002a.
- CALVO, T., MAYOR, G. y MESIAR, R. *Aggregation operators: New trends and applications*. Physica-Verlag, 2002b.
- CAMPOMANES-ÁLVAREZ, C., IBÁÑEZ, O. y CORDÓN, O. Design of criteria to assess craniofacial correspondence in forensic identification based on computer vision and fuzzy integrals. *Applied Soft Computing*, vol. 46, páginas 596–612, 2016.
- CANNY, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8(6), páginas 679–698, 1986.

- CANNY, J. F. Finding edges and lines in images. 1983.
- CASILLAS, J., CORDÓN, O., TRIGUERO, F. H. y MAGDALENA, L. *Interpretability issues in fuzzy modeling*, vol. 128. Springer, 2013.
- CHAMORRO-MARTÍNEZ, J., SÁNCHEZ, D., PRADOS-SUÁREZ, B., GALÁN-PERALES, E. y VILA, M. A. A hierarchical approach to fuzzy segmentation of colour images. En *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on*, vol. 2, páginas 966–971. IEEE, 2003.
- CHAUDHRY, S. y CHANDRA, R. Face detection and recognition in an unconstrained environment for mobile visual assistive system. *Applied Soft Computing*, vol. 53, páginas 168–180, 2017.
- CHENG, H.-D. y SUN, Y. A hierarchical approach to color image segmentation using homogeneity. *IEEE Transactions on image processing*, vol. 9(12), páginas 2071–2082, 2000.
- CHICLANA, F., HERRERA, F., HERRERA-VIEDMA, E. y MARTINEZ, L. A note on the reciprocity in the aggregation of fuzzy preference relations using owa operators. *Fuzzy Sets and Systems*, vol. 137(1), páginas 71–83, 2003.
- CLAUSET, A., MOORE, C. y NEWMAN, M. E. Hierarchical structure and the prediction of missing links in networks. *Nature*, vol. 453(7191), página 98, 2008.
- CONNOLLY, C. y FLEISS, T. A study of efficiency and accuracy in the transformation from rgb to cielab color space. *IEEE Transactions on Image Processing*, vol. 6(7), páginas 1046–1048, 1997.
- DUBOIS, D., OSTASIEWICZ, W. y PRADE, H. Fuzzy sets: history and basic notions. En *Fundamentals of fuzzy sets*, páginas 21–124. Springer, 2000.
- DUBOIS, D. y PRADE, H. *Fundamentals of fuzzy sets*, vol. 7. Springer Science & Business Media, 2012.
- DUDA, R. y HART, P. *Pattern Classification and Scene Analysis*. EEUU, 1973.
- DUNN, J. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, vol. 3(3), páginas 32–57, 1973.
- ESPINILLA, M., MONTERO, J. y RODRÍGUEZ, J. T. Computational intelligence in decision making. *International Journal of Computational Intelligence Systems*, vol. 7(sup1), páginas 1–5, 2014.

- ESTRADA, F. J. y JEPSON, A. D. Benchmarking image segmentation algorithms. *International Journal of Computer Vision*, vol. 85(2), páginas 167–181, 2009.
- FELZENSZWALB, P. F. y HUTTENLOCHER, D. P. Efficient graph-based image segmentation. *International journal of computer vision*, vol. 59(2), páginas 167–181, 2004.
- FLAVELL, S., WINTER, S. y WILSON, D. Matching region adjacency graphs. *Microprocessing and Microprogramming*, vol. 31(1-5), páginas 31–33, 1991.
- FORTUNATO, S. Community detection in graphs. *Physics reports*, vol. 486(3-5), páginas 75–174, 2010.
- FRANCO, C., GUADA, C., RODRÍGUEZ, J. T., GÓMEZ, D., MONTERO, J., RASMUSSEN, J. y NIELSEN, J. Automatic detection of thistle-weeds in cereal crops from aerial rgb images. En *17th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, página in press. 2018.
- FU, K.-S. y MUI, J. A survey on image segmentation. *Pattern Recognition*, vol. 13(1), páginas 3–16, 1981.
- GIRVAN, M. y NEWMAN, M. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99(12), páginas 7821–7826, 2002a.
- GIRVAN, M. y NEWMAN, M. E. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, vol. 99(12), páginas 7821–7826, 2002b.
- GLASBEY, C. A. An analysis of histogram-based thresholding algorithms. *CVGIP: Graphical models and image processing*, vol. 55(6), páginas 532–537, 1993.
- GÓMEZ, D., FIGUEIRA, J. R. y EUSÉBIO, A. Modeling centrality measures in social network analysis using bi-criteria network flow optimization problems. *European Journal of Operational Research*, vol. 226(2), páginas 354–365, 2013.
- GÓMEZ, D. y MONTERO, J. Fuzzy sets in remote sensing classification. *Soft Computing*, vol. 12, páginas 243–249, 2008.
- GÓMEZ, D., MONTERO, J. y BIGING, G. Improvements to remote sensing using fuzzy classification, graphs and accuracy statistics. *Pure and Applied Geophysics*, vol. 165(8), páginas 1555–1575, 2008.
- GÓMEZ, D., MONTERO, J. y PODOIMANI, C. A graph coloring algorithm approach for image segmentation. *Omega*, vol. 35, páginas 173–183, 2007.

- GÓMEZ, D., YÁÑEZ, J., GUADA, C., RODRÍGUEZ, J. T., MONTERO, J. y ZARRAZOLA, E. Fuzzy image segmentation based upon hierarchical clustering. *Knowledge-Based Systems*, vol. 87, páginas 26–37, 2015a.
- GÓMEZ, D., ZARRAZOLA, E., YÁÑEZ, J. y MONTERO, J. A divide-and-link algorithm for hierarchical clustering in networks. *Information Sciences*, vol. 316, páginas 308–328, 2015b.
- GÓMEZ, D., ZARRAZOLA, E., YÁÑEZ, J., RODRÍGUEZ, J. y MONTERO, J. A new concept of fuzzy image segmentation. En *WSPC - Proceedings Trim Size*. 2014.
- GONDRAN, M., MINOUX, M. y VAJDA, S. *Graphs and algorithms*. John Wiley & Sons, Inc., 1984.
- GONZÁLEZ, R., WOODS, R. y EDDINS, S. Digital Image Processing using MATLAB. *Chapman and Hall Computing*, 2009.
- GRADY, L. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28(11), páginas 1768–1783, 2006.
- GRAU, V., MEWES, A., ALCANIZ, M., KIKINIS, R. y WARFIELD, S. K. Improved watershed transform for medical image segmentation using prior information. *IEEE Transactions on medical imaging*, vol. 23(4), páginas 447–458, 2004.
- GREIG, D., PORTEOUS, B. y SEHEULT, A. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 51(2), páginas 271–279, 1989.
- GUADA, C., GÓMEZ, D., RODRÍGUEZ, J., YÁÑEZ, J. y MONTERO, J. Classifying image analysis techniques from their output. *International Journal of Computational Intelligence Systems*, vol. 9(sup1), páginas 43–68, 2016.
- GUADA, C., ZARRAZOLA, E., YÁÑEZ, J., RODRÍGUEZ, J. T., GÓMEZ, D. y MONTERO, J. A novel edge detection algorithm based on a hierarchical graph-partition approach. *Journal of Intelligent and Fuzzy Systems*, vol. 34(3), páginas 1875–1892, 2018.
- GUIMARÃES, S. J. F., COUSTY, J., KENMOCHI, Y. y NAJMAN, L. An efficient hierarchical graph based image segmentation. *arXiv preprint arXiv:1206.2807*, 2012.
- HOROWITZ, S. y PAVLIDIS, T. Picture segmentation by a directed split and merge procedure. *Proc. of 2nd international joint conference on pattern recognition*, páginas 424–433, 1974.

- KERRE, E. E. *Introduction to the basic principles of fuzzy set theory and some of its applications*. Communication and Cognition, 1993.
- KLIR, G. y YUAN, B. *Fuzzy sets and fuzzy logic*, vol. 4. Prentice hall New Jersey, 1995.
- KRUSKAL, J. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, vol. 7, páginas 48–50, 1956.
- LEZORAY, O. y GRADY, L. *Graph Approach in Image Segmentation*, páginas 1–24. CRC Press, Boca Raton, 2012.
- LHERMITTE, S., VERBESSELT, J., JONCKHEERE, I., NACKAERTS, K., VAN AARDT, J. A., VERSTRAETEN, W. W. y COPPIN, P. Hierarchical image segmentation based on similarity of ndvi time series. *Remote Sensing of Environment*, vol. 112(2), páginas 506–521, 2008.
- LIANG, L. y LOONEY, C. Competitive fuzzy edge detection. *Applied Soft Computing*, vol. 3, páginas 123–137, 2003.
- LIU, G. y HARALICK, R. M. Optimal matching problem in detection and recognition performance evaluation. *Pattern Recognition*, vol. 35(10), páginas 2125–2139, 2002.
- LIU, X., YU, Y., LIU, B. y LI, Z. Bowstring-based dual-threshold computation method for adaptive canny edge detector. En *28th International Conference of Image and Vision Computing New Zealand*, páginas 13–18. IEEE, 2013.
- LLADOS, J., MARTI, E. y VILLANUEVA, J. Symbol Recognition by Error - Tolerant Subgraph Matching between Region Adjacency Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23(10), páginas 1137–1143, 2001.
- LÓPEZ-MOLINA, C. *The breakdown structure of edge detection: Analysis of individual components and revisit of the overall structure*. Tesis Doctoral, Universidad Pública de Navarra, 2012.
- LÓPEZ-MOLINA, C., BUSTINCE, H., FERNÁNDEZ, J., COUTO, P. y DE BAETS, B. A gravitational approach to edge detection based on triangular norms. *Pattern Recognition*, vol. 43(11), páginas 3730–3741, 2010.
- LÓPEZ-MOLINA, C., DE BAETS, B. y BUSTINCE, H. Quantitative error measures for edge detection. *Pattern Recognition*, vol. 46(4), páginas 1125–1139, 2013a.

- LÓPEZ-MOLINA, C., DE BAETS, B., BUSTINCE, H., SANZ, J. y BARRENECHEA, E. Multiscale edge detection based on gaussian smoothing and edge tracking. *Knowledge-Based Systems*, vol. 44, páginas 101–111, 2013b.
- LU, J. y WENG, Q. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, vol. 28(5), páginas 823–870, 2007.
- MA, Y., ZHAN, K. y WANG, Z. Applications of Pulse-Coupled Neural Networks. 2010.
- MADRID, N. y HURTIK, P. Lane departure warning for mobile devices based on a fuzzy representation of images. *Fuzzy Sets and Systems*, vol. 291, páginas 144–159, 2016.
- MANGAN, A. y WHITAKER, R. Partitioning 3D Surface Meshes Using Watershed Segmentation. *IEEE Transactions on Visualization and Computer Graphics*, vol. 5(4), páginas 308–321, 1999.
- MARR, D. y HILDRETH, E. Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 207(1167), páginas 187–217, 1980.
- MARTIN, D., FOWLKES, C., TAL, D. y MALIK, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. En *8th International Conference in Computer Vision*, vol. 2, páginas 416–423. 2001.
- MARTIN, D. R., FOWLKES, C. C. y MALIK, J. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26(5), páginas 530–549, 2004.
- MAYOR, G. y TORRENS, J. On a class of binary operations: non-strict archimedean aggregation functions. En *Multiple-Valued Logic, 1988., Proceedings of the Eighteenth International Symposium on*, páginas 54–59. IEEE, 1988.
- MEDINA-CARNICER, R., MADRID-CUEVAS, F. J., CARMONA-POYATO, A. y MUÑOZ-SALINAS, R. On candidates selection for hysteresis thresholds in edge detection. *Pattern Recognition*, vol. 42(7), páginas 1284–1296, 2009.
- MONTERO, J. Rational aggregation rules. *Fuzzy Sets and Systems*, vol. 62(3), páginas 267–276, 1994.
- NACHTEGAEL, M., KERRE, E. E., PHILIPS, W. y VAN DER WEKEN, D. *Soft computing in image processing: recent advances*, vol. 210. Springer, 2007.

- NAJMAN, L., COUSTY, J. y PERRET, B. Playing with kruskal: algorithms for morphological trees in edge-weighted graphs. En *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, vol. 7883, páginas 135–146. Springer, 2013.
- NAPALKOVA, L., ROZENBLIT, J. W., HWANG, G., HAMILTON, A. J. y SUANTAK, L. An optimal motion planning method for computer-assisted surgical training. *Applied Soft Computing*, vol. 24, páginas 889–899, 2014.
- NEWMAN, M. Fast algorithm for detecting community structure in networks. *Physical Review E*, vol. 69(6), 2004a.
- NEWMAN, M. E. Fast algorithm for detecting community structure in networks. *Physical review E*, vol. 69(6), página 066133, 2004b.
- NOVÁK, V., PERFILIEVA, I. y MOCKOR, J. *Mathematical principles of fuzzy logic*, vol. 517. Springer Science & Business Media, 2012.
- OUZOUNIS, G. K. y SOILLE, P. Pattern spectra from partition pyramids and hierarchies. En *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, páginas 108–119. Springer, 2011.
- PAL, N. R. y PAL, S. K. A review on image segmentation techniques. *Pattern recognition*, vol. 26(9), páginas 1277–1294, 1993.
- PAL, S. y KING, R. On edge detection of x-ray images using fuzzy sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5(1), páginas 69–77, 1983.
- PAVLIDIS, T. *A graphics and image processing*. Verlag, 1982.
- PENG, B., ZHANG, L. y ZHANG, D. A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, vol. 46(3), páginas 1020–1038, 2013.
- PHILLIPS, D. *Image Processing in C*. 2000.
- PRATT, W. *Digital image processing*. 2001.
- PREWITT, J. M. Object enhancement and extraction. *Picture processing and Psychopictorics*, vol. 10(1), páginas 15–19, 1970.
- RAMANATH, R., KUEHNI, R. G., SNYDER, W. E. y HINKS, D. Spectral spaces and color spaces. *Color Research & Application*, vol. 29(1), páginas 29–37, 2004.
- RASHEDI, E. y NEZAMABADI-POUR, H. A stochastic gravitational approach to feature based color image segmentation. *Engineering Applications of Artificial Intelligence*, vol. 26(4), páginas 1322–1332, 2013.

- REEVES, A. P., AKEY, M. L. y MITCHELL, O. R. A moment based two-dimensional edge operator. En *Proc. CVPR*, vol. 312, página 317. 1983.
- RICHARDS, J. y JIA, X. Remote Sensing Digital Image Analysis. An Introduction. 1999.
- ROBERTS, L. Machine perception of three-dimensional solids. En *Optical and Electro-optical Information Processing*, páginas 159–197. MIT, 1965.
- RODRÍGUEZ, J. T., GUADA, C., GÓMEZ, D., YÁÑEZ, J. y MONTERO, J. A methodology for hierarchical image segmentation evaluation. En *Information Processing and Management of Uncertainty in Knowledge-Based Systems* (editado por Springer), vol. 610, páginas 635–647. Communications in Computer and Information Science, 2016.
- ROERDINK, J. B. y MEIJSTER, A. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta informaticae*, vol. 41(1, 2), páginas 187–228, 2000.
- ROSENFELD, A. Fuzzy digital topology. *Information and Control*, vol. 40(1), páginas 76–87, 1979.
- ROSENFELD, A. y THURSTON, M. Edge and curve detection for visual scene analysis. *IEEE Transactions on Computers*, vol. 100(5), páginas 562–569, 1971.
- ROSIN, P. L. Unimodal thresholding. *Pattern recognition*, vol. 34(11), páginas 2083–2096, 2001.
- RUSSO, F. A user-friendly research tool for image processing with fuzzy rules. *Proceedings of the First IEEE International Conference on Fuzzy Systems*, páginas 561–568, 1992.
- RUSSO, F. Edge detection in noisy images using fuzzy reasoning. *Instrumentation and Measurement Technology Conference*, vol. 1, páginas 369–372, 1998.
- SAHA, P. K., UDUPA, J. K. y ODHNER, D. Scale-based fuzzy connected image segmentation: theory, algorithms, and validation. *Computer Vision and Image Understanding*, vol. 77(2), páginas 145–174, 2000.
- SAHOO, P. K., SOLTANI, S. y WONG, A. K. A survey of thresholding techniques. *Computer vision, graphics, and image processing*, vol. 41(2), páginas 233–260, 1988.
- SCHROETER, P. y BIGÜN, J. Hierarchical image segmentation by multi-dimensional clustering and orientation-adaptive boundary refinement. *Pattern Recognition*, vol. 28(5), páginas 695–709, 1995.

- SEISING, R. *The Fuzzification of Systems: The Genesis of Fuzzy Set Theory and its Initial Applications-Developments up to the 1970s*. Springer Publishing Company, Incorporated, 2007.
- SENTHILKUMARAN, N. y RAJESH, R. Edge detection techniques for image segmentation—a survey of soft computing approaches. *International journal of recent trends in engineering*, vol. 1(2), páginas 250–254, 2009a.
- SENTHILKUMARAN, N. y RAJESH, R. Edge detection techniques for image segmentation. a survey of soft computing approaches. *International journal of recent trends in engineering*, vol. 1(2), páginas 250–254, 2009b.
- SEZGIN, M. y SANKUR, B. Survey over image thresholding techniques and quantitative performance evaluation. *Electronic Imaging*, vol. 1(13), páginas 146–165, 2004.
- SHI, J. y MALIK, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22(8), páginas 888–905, 2000.
- SINHA, S. Graph cut algorithms in vision, graphics and machine learning. an integrative paper. *UNC Chapel Hill*, 2004.
- SOBEL, I. History and Definition of the Sobel Operator. 2014.
- SOILLE, P. Constrained connectivity for hierarchical image partitioning and simplification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30(7), páginas 1132–1145, 2008.
- SOLOMON, C. y BRECKON, T. *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons, 2011.
- SONKA, M., HLAVAC, V. y BOYLE, R. *Image Processing, Analysis and Machine Vision*. Gran Bretaña, UK, 1993.
- SONKA, M., HLAVAC, V. y BOYLE, R. *Image Processing, Analysis, and Machine Vision. International Student Edition*. 2008.
- SOTELO, M. A., RODRIGUEZ, F. J., MAGDALENA, L., BERGASA, L. M. y BOQUETE, L. A color vision-based lane tracking system for autonomous driving on unmarked roads. *Autonomous Robots*, vol. 16(1), páginas 95–116, 2004.
- SUN, P. G., GAO, L. y YANG, Y. Maximizing modularity intensity for community partition and evolution. *Information Sciences*, vol. 236, páginas 83–92, 2013.

- TATIRAJU, S. y MEHTA, A. Image Segmentation using k-means clustering, EM and Normalized Cuts. Informe técnico, Department of Eecs, University of California, 2008.
- TELLAECHE, A., PAJARES, G., BURGOS-ARTIZZU, X. P. y RIBEIRO, A. A computer vision approach for weeds identification through support vector machines. *Applied Soft Computing*, vol. 11(1), páginas 908–915, 2011.
- TERANO, T., ASAI, K. y SUGENO, M. *Fuzzy systems theory and its applications*. Academic Press Professional, Inc., 1992.
- TETALI, P. Random walks and the effective resistance of networks. *Journal of Theoretical Probability*, vol. 4(1), páginas 101–109, 1991.
- KERMIT IMAGE TOOLKIT, T. The Kermit Image Toolkit (KIT). www.kermitimagetoolkit.com, 2015.
- TRÉMEAU, A. y COLANTONI, P. Regions Adjacency Graph Applied to Color Image Segmentation. *IEEE Transactions on Image Processing*, vol. 9(4), páginas 735–744, 2000.
- UDUPA, J. K. y SAMARASEKERA, S. Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation. *Graphical models and image processing*, vol. 58(3), páginas 246–261, 1996.
- VINCENT, L. y BEUCHER, S. *The morphological approach to Segmentation: An Introduction*. School of Mines, Paris, 1989.
- WACKER, A. y LANDGREBE, D. Minimum distance classification in remote sensing. 1972.
- WASSERMAN, S. y FAUST, K. *Social network analysis: Methods and applications*, vol. 8. Cambridge university press, 1994.
- WECHSLER, H. y KIDODE, M. A random walk procedure for texture discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1(3), páginas 272–280, 1979.
- WESZKA, J., NAGEL, R. y ROSENFELD, A. *A technique for facilitating threshold selection for object extraction from digital pictures*, vol. TR-243. University of Maryland Computer Science Center, 1973.
- XU, Y., CARLINET, E., GÉRAUD, T. y NAJMAN, L. Hierarchical segmentation using tree-based shape space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39(3), páginas 457–469, 2017.
- ZADEH, L. Fuzzy sets. *Information and Control*, vol. 8, páginas 338–353, 1965.

-
- ZARRAZOLA, E. *Segmentación Jerárquica en Redes. Aplicaciones*. Tesis Doctoral, Universidad Complutense de Madrid, España, 2013.
- ZHANG, H., FRITTS, J. y GOLDMAN, S. Image Segmentation Evaluation: A Survey of Unsupervised Methods. *Computer Vision and Image Understanding*, vol. 110, páginas 260–280, 2008.
- ZIOU, D., TABBONE, S. ET AL. Edge detection techniques-an overview. *Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii*, vol. 8, páginas 537–559, 1998.

*En la ciencia el reconocimiento se concede
al hombre quien convence al mundo,
no a aquel a quien se le ocurre la idea*

Willian Osler