



Segmentación de Imágenes Médicas mediante Grafos Piramidales Irregulares

por

Carlos Abraham Solorio Navarro

Tesis proporcionada como cumplimiento parcial de los requisitos
para el título de la
Ingeniería en Ciencias de la Computación

Asesor: Dr. Carlos Guillén Galván

Agosto 2012

Resumen

La segmentación tiene como meta simplificar o cambiar la representación de una imagen en algo que sea más fácil de analizar y manejar, ya sea para la interpretación humana de los resultados obtenidos o para la búsqueda automatizada de características significativas. Este trabajo de tesis presenta una técnica de segmentación que aprovecha las características de los grafos para representar la información visual de una imagen, esto es, la intensidad de gris o de color en un punto de un plano bidimensional. Inicialmente, los nodos del grafo representan cada uno de los píxeles de la imagen de forma única y las aristas pueden representar toda clase de relaciones entre los píxeles, por ejemplo (en el caso particular de este trabajo) las colindancias o vecindades entre estos. Ésta, así como algunas otras técnicas de segmentación, divide la imagen en regiones que no se traslapan. El concepto intuitivo de regiones no traslapadas en imágenes se relaciona análogamente con un concepto formal en la teoría de grafos, y se le conoce como *Grafo Plano*. La semejanza conceptual entre las características de una imagen y la de los grafos no son las únicas ventajas que se presentan con este enfoque, también puede observarse que una imagen trasladada o rotada, posee un grafo isomorfo al grafo de la imagen original, es decir, ambos grafos poseen la misma estructura y la segmentación que se obtiene de la aplicación de la técnica, en consecuencia, resulta invariante. Gracias a estas importantes propiedades, se formula como hipótesis que la segmentación de imágenes basada en grafos puede proveer resultados preferibles que los que otras técnicas pueden ofrecer. En este proyecto se propone una implementación de los algoritmos de Borůvka aplicados en grafos pirámides irregulares orientados a la segmentación de algunas imágenes médicas. Específicamente, se comparan los resultados con la segmentación realizada por humanos y con la técnica de segmentación inducida por escala invariante, ambas mostradas en [5] (The Berkeley Segmentation Dataset and Benchmark) y con la funcional Mumford y Shah en [20].

Palabras Clave: Técnicas de segmentación, imágenes médicas, grafos piramidales, benchmark de segmentaciones, funcional Mumford y Shah.

Índice

Resumen	i
Agradecimientos	iv
Lista de Figuras	vi
Lista de Tablas	viii
Acrónimos y Notación	x
1 Introducción	1
2 Grafos	6
2.1 Conceptos básicos	6
2.2 Relaciones de contención de grafos y grafos especiales	9
2.3 Conectividad	13
2.4 Árboles	15
2.5 Operaciones en grafos	16
2.6 Álgebra lineal	18
2.7 Grafos planos y duales	20
3 Métodos de segmentación	24
3.1 Tipos de segmentación	25
3.1.1 Umbralización y crecimiento de regiones	26
3.1.2 Clasificadores y agrupadores	28
3.2 Funcional Mumford y Shah	29
4 Grafos Piramidales	31
4.1 Representación Digital de Color	31
4.2 Construcción de Grafos Piramidales	33
4.2.1 Pirámides Regulares	35
4.2.2 Pirámides Irregulares	35
4.2.3 Grafos Imagen Duales	38
4.3 Contracción de Grafo Dual	38
4.3.1 Núcleos de Contracción	39

4.3.2	Núcleos de Contracción Equivalentes	42
4.3.3	Grafo Dual Piramidal	43
4.4	Jerarquía de Particiones	44
4.4.1	Árbol de Expansión Mínima (MST)	44
4.4.2	Construcción de la Jerarquía de Particiones	46
5	Implementación y resultados	52
5.1	Implementación	52
5.2	Resultados	54
5.2.1	Mediciones	54
5.2.2	Segmentación de propósito general	55
5.2.3	DGC vs Benchmarks	58
5.2.4	Segmentación en fotografía ocular	61
6	Conclusiones y trabajo a futuro	70
6.1	Conclusiones	70
6.2	Trabajo a futuro	71
	Apéndice A: Código	72
	Contracción Grafo Dual mediante el algoritmo de Borůvka	72
	Bibliografía	82

Agradecimientos

A todas las personas que me brindaron su apoyo, su confianza y especialmente su paciencia en este largo y duro proceso, lleno de altibajos en las diferentes etapas que lo comprendieron.

A mis padres y mi hermana - Lucía Silvia Navarro Rivera, José Carlos Solorio Cervantes y Cynthia Karla Solorio Navarro - que, gracias a su incondicional respaldo, duro trabajo, fuerte sentido de responsabilidad, disciplina y ética a lo largo de los años, fui bendecido con una sólida educación, un conjunto irremplazable de valores morales y una invaluable formación académica.

A mi sobrino, Luka Sandoval Solorio, que es una de mis más grandes motivaciones para continuar con el trabajo y la pasión del estudio, pues él y la siguiente generación serán los testigos del mundo que la presente generación estamos construyendo.

A todas las personas que han sido modelos a seguir: a las familias Tlalmis Velasco y Flores Pérez, por dar un ejemplo sin igual de unión y cooperación familiar.

A todas las personas que sin tapujos me han brindado su amistad (An, Denisse, Mariel, Joel, Carlos y un gran etcétera, entre ellos cada uno de los integrantes del antiguo equipo de colaboradores del Módulo 2: Dr. Mario Anzures, Jorge, Omar, Aurelio, Martha... Y a todos los que dejo en el tintero), que han estado disponibles para mí por cualquier medio para ofrecerme su apoyo sin importar el día o la hora, me demostraron que las verdaderas amistades son posibles y que una vez alcanzadas, son insustituibles. Todas estas personas marcaron el inicio de una etapa de crecimiento personal que me ayudó de forma inigualable en un momento de mi vida que considero fue especialmente difícil.

A Nayeli que, además de ofrecerme su amistad y una gran cantidad de su tiempo, ha sido un modelo a seguir por su humildad, amabilidad, disposición, un intachable sentido de justicia y voluntad para hacer el bien público. Me ha mostrado un sin fin de perspectivas que ignoraba y que, sin duda, han reforzado y acelerado el proceso de desarrollo y madurez. Aunque, claramente, todavía queda un largo camino por recorrer. Estoy seguro que, sin importar lo que depare el futuro, su influencia habrá dejado una buena huella.

A la Vicerrectoría de Investigación y Estudios de Posgrado (VIEP) por el grado de confianza que tuvieron en este trabajo y que permitió, gracias a su apoyo económico, la realización de cada una de las etapas de esta tesis. A todas las personas que ayudaron a mejorar el trabajo gracias a sus observaciones, críticas y sugerencias, especialmente a los doctores Irene Olaya Ayaquica Martínez, Daniel Alejandro Valdés Amaro y Rafael Lemuz López. Por último, pero no menos importante, a todo el plantel de docentes de la facultad de ciencias de la computación que me enseñaron directa o indirectamente valiosas lecciones sobre esfuerzo, pasión, creatividad y humanidad, haciendo mención especial de los doctores Manuel Isidro Martín Ortiz, Lourdes Sandoval Solís, César Bautista Ramos y Carlos Guillén Galván. Estas destacadas personas y sus grandes logros se han convertido para mí en grandes ejemplos y le han dado forma a mis metas a largo plazo.

A todos, les extiendo mi más profundo y sincero agradecimiento. El entorno de toda una vida es el que forja el carácter de una persona y ustedes han sido elementos fundamentales de ese entorno.

¡GRACIAS!

Índice de figuras

1.1	Ejemplo de segmentación	2
1.2	Detección de bordes	2
2.1	Ejemplo de construcción de un grafo	7
2.2	Simplificación de grafo	7
2.3	Representación de grafos dirigidos y no dirigidos.	7
2.4	Tipo de vértices según su grado.	8
2.5	Grafos inducidos	10
2.6	Diagrama de Hasse	11
2.7	Ejemplo de subgrafo independiente.	11
2.8	Camino, trayectoria y ruta	12
2.9	Cintura y circunferencia de un grafo	12
2.10	Conectividad en grafos	14
2.11	Corte	14
2.12	Conjuntos de corte	15
2.13	Árboles.	15
2.14	Árbol enraizado	16
2.15	Unión e intersección.	17
2.16	Diferencia simétrica.	17
2.17	Eliminación de vértices.	17
2.18	Eliminación de aristas.	18
2.19	Identificación de vértices y contracción de aristas	18
2.20	Grafos planos	22
2.21	Grafos duales	23
3.1	Segmentación con Umbralización y Crecimiento de Regiones	27
3.2	Segmentación con K-medias [4].	28
3.3	Segmentaciones basadas en la funcional de Mumford y Shah.	30
4.1	Modelos o espacios de colores	32
4.2	Pirámide	34
4.3	Construcción de grafos imagen duales	39
4.4	Procedimiento de contracción de grafo dual	42
4.5	Contraste interno y externo	48
4.6	Construcción de pirámide y obtención de segmentación.	51

5.1	Curvas que describen la conducta de la memoria requerida y el tiempo de cómputo de la técnica propuesta.	55
5.2	Segmentación en “ <i>Object11</i> ”	57
5.3	Segmentación en “ <i>Object45</i> ”	57
5.4	Segmentación en “ <i>Rampa</i> ”	58
5.5	Segmentación en “ <i>Lena3</i> ”	59
5.6	Segmentación en “ <i>Monarch</i> ”	60
5.7	Comparación de resultados del objeto 42049 en [5].	62
5.8	Comparación de resultados del objeto 189080 en [5].	63
5.9	Comparación de resultados del objeto 227092 en [5].	64
5.10	Segmentación en fotografía ocular 1 (con coroidopatía serosa central)	66
5.11	Segmentación en fotografía ocular 2 (retina normal)	67
5.12	Segmentación en fotografía ocular 3 (retina normal)	68
5.13	Segmentación en fotografía ocular 4 (retinopatía diabética edematosa)	69

Índice de tablas

2.1	Isomorfismo entre dos grafos	13
2.2	Cortes de un grafo G	15
5.1	Mediciones de segmentación.	54

Lista de algoritmos

4.1	Construcción de una pirámide de grafo dual	44
4.2	Algoritmo de Borůvka	45
4.3	Algoritmo de Borůvka con DGC (<i>Dual Graph Contraction</i>)	46
4.4	Jerarquía de particiones	49

Acrónimos y Notación

CR	C ampo R eceptivo
CP	C onnecting P ath
DGC	D ual G raph C ontraction
MST	M inimum S panning T ree
MIS	M aximal I ndependent S et
MIES	M aximal I ndependent E dge S et
RAG	R egion A djacency G raph
RGB	R ed G reen B lue
CMYK	C yan M agenta Y ellow B lack
HSI	H ue S aturation I ntensity
HSV	H ue S aturation V alue

Capítulo 1

Introducción

La segmentación de imágenes se refiere, de forma muy básica, a la división de esta en sus partes fundamentales [21, 22]. Esta definición tan abierta presenta ambigüedad al tratar de imaginar cuáles podrían ser las partes más importantes de una imagen. Esta decisión varía dependiendo del tipo de imagen que se esté analizando y qué objeto u objetos son los que se buscan recuperar de la misma. Por ejemplo, en la figura 1.1 se busca identificar únicamente la aeronave. En el análisis visual de componentes manufacturados en líneas de producción, la imagen sólo se dividiría hasta poder identificar la presencia o ausencia de desperfectos en las piezas, tales como deformaciones, rupturas o componentes faltantes. Del mismo modo, en el análisis de imágenes médicas (radiografías, resonancias, ultrasonidos, angiografías, etc.), la división se efectuará hasta poder identificar anomalías que podrían llevar al diagnóstico de alguna enfermedad o condición anormal.

Una imagen digital es, básicamente, una cuadrícula formada por una cantidad determinada de filas y columnas, que representa un espacio bidimensional (área). Cada entrada de la cuadrícula, conocida generalmente como **píxel**, contiene uno o varios valores que describen de forma numérica el color en ese punto.

Independientemente de las peculiaridades de los objetos que puedan presentarse en las imágenes, uno desearía idealmente obtener de la simplificación de la imagen; regiones fácilmente distinguibles que constituyan objetos importantes, resaltados



FIGURA 1.1: Ejemplo de segmentación

por algún *borde* bien definido. Se consideran a los *bordes* de los objetos como una característica lo suficientemente general que facilita el proceso para encontrar regiones que pertenezcan o representen algún área de interés particular en la imagen. La figura 1.2 muestra una detección general de todos los bordes de la imagen, sin resaltar alguna región en particular.

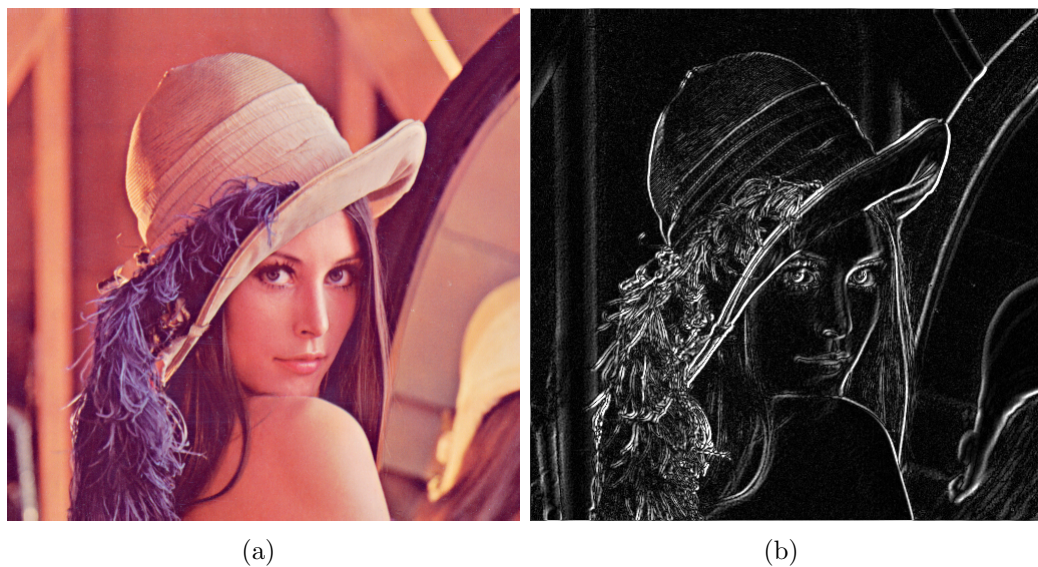


FIGURA 1.2: Detección de bordes

Se tiene como noción básica que el borde de un objeto dentro de una imagen es representado por una diferencia notable de intensidad entre píxeles contiguos. Este enfoque ha sido ampliamente estudiado y las aplicaciones resultado de este análisis son conocidas como “detectores de bordes”, en los que comúnmente se

aplican las primeras y segundas derivadas de una imagen, es decir, la medida del cambio de intensidad entre los *píxeles* de una imagen [22].

En la literatura ([4, 7, 21, 22]), la segmentación ha sido clasificada en las siguientes clases:

- Segmentación Suave: Aquella que permite que las regiones o clases se superpongan (regiones traslapadas).
- Segmentación Dura: La cual obliga a tomar una decisión en cuanto a si el píxel pertenece exclusivamente a una región (regiones no traslapadas).
- Segmentación Discreta: Se da cuando opera típicamente en la misma malla discreta de la imagen (por píxeles).
- Segmentación Continúa: Opera en modelos deformables que proveen precisión a nivel sub-píxel.

De forma más general, la segmentación se divide en dos grandes clases: Segmentación basada en regiones y segmentación basada en bordes [2]. Cada clase tiene sus ventajas y desventajas. Los métodos basados en regiones asignan píxeles a una región de acuerdo a alguna propiedad (como homogeneidad). Su ventaja reside en que trabajan bien en imágenes con ruido, donde los bordes son difíciles de detectar. Sin embargo, pueden generarse regiones sobrantes en forma de hoyos dentro de otras regiones o generar bordes ásperos. Los métodos basados en bordes generan límites de los objetos segmentados. En estos métodos, se puede incluir conocimiento de la forma del objeto para limitar el resultado de la segmentación, pero frecuentemente esto conduce a un alisado excesivo. También se encuentra el problema de que estos métodos son mucho más susceptibles al ruido y generalmente requieren de pre-procesamiento para delimitar inicialmente regiones.

En [2] se propone una adaptación a la funcional de Mumford y Shah para mejorar la calidad de la segmentación de imágenes que contienen objetos fractales. A pesar de que existe una mejora considerable, los resultados no son tan satisfactorios debido a que existen muchas regiones en la imagen que no son segmentadas debido a la

poca sensibilidad del algoritmo a pequeñas variaciones de intensidad. Es por esto que, es importante analizar otras técnicas de segmentación que puedan ayudar a mejorar la calidad de la segmentación.

El objetivo de esta tesis se centra en la aplicación de una técnica de segmentación y comparar los resultados con otras segmentaciones. Específicamente, se tiene como objetivo llevar la imagen a segmentar a su representación en grafo para que, con base en esta, sea posible construir una pirámide irregular o grafo piramidal irregular. Se busca comparar los resultados de la técnica en imágenes generales con los resultados de otro algoritmo de segmentación que sirve como referencia (Berkeley Benchmark [5]) y finalmente aplicar la técnica a imágenes médicas. En éstas, existe una gran variedad de particularidades que pueden presentarse y que son importantes en un diagnóstico. Sin embargo, las dificultades presentes en el problema de segmentar esta clase de imágenes aún no ha sido resuelto por las técnicas actuales, como es el caso de una incorrecta diferenciación entre tipos de materias en el cerebro (en una imagen de resonancia magnética, por ejemplo) o la omisión de vasos sanguíneos anormales en un angiografía (en ambos casos, por intensidades muy similares entre el objeto de interés y los objetos contiguos). Las distintas variaciones existentes en los algoritmos de segmentación utilizados permiten enfrentar casos específicos, pero no son adecuados en todas las posibles circunstancias. Por estas razones, se busca implementar una técnica de segmentación basada en grafos que, tomando en cuenta sus propiedades naturales, se tiene como hipótesis que es más efectiva y general que otras.

En el capítulo 2 se definen formalmente los elementos fundamentales necesarios para la implementación de la técnica de segmentación propuesta: Esto es la definición formal de un grafo, distintas formas de describir su estructura, los tipos de grafos, las operaciones entre grafos y grafos especiales (inducidos, maximales, planos, etc.).

En el capítulo 3 se describen brevemente otros métodos de segmentación y los resultados que ofrecen, tanto en imágenes con las características en las que son adecuados y en las que no. Se hará especial mención a la técnica basada en

la funcional de Mumford y Shah propuesta en [2], pues esta ha sido aplicada específicamente a imágenes médicas a diferencia de las otras técnicas mencionadas.

En el capítulo 4 se detalla la estructura de los distintos modelos de colores utilizados para representar imágenes. Cada modelo afecta los atributos de los nodos en la pirámide y, en consecuencia, las mediciones de similaridad en las regiones de cada nivel del grafo. También se define y describe a detalle la técnica propuesta en este trabajo, basada en los grafos piramidales irregulares, y se explican los algoritmos utilizados para realizar la contracción de los grafos duales, para encontrar las particiones de elementos que conformen las regiones y finalmente para construir la pirámide.

En el capítulo 5 se presentan los resultados obtenidos mediante la segmentación basada en grafos piramidales irregulares enfocada como técnica de segmentación de propósito general y se hace una comparación entre estos y los resultados de referencia que se encuentran en [5]. Por último también se muestran los resultados de la técnica aplicada en imágenes médicas.

En el capítulo 6 se presentan las conclusiones obtenidas del trabajo, las limitaciones y posibilidades de la técnica y el trabajo futuro que puede realizarse para mejorarla.

Por último, el código en *C#* que corresponde a los algoritmos presentados en el capítulo 4 se encuentra anexado en el apéndice A.

Capítulo 2

Grafos

En este capítulo se establece la base de conceptos que serán de utilidad para la implementación de los algoritmos de segmentación de imágenes mediante grafos pirámide irregulares. El desarrollo de la teoría mostrada aquí se apoya fuertemente en [12, 17, 25, 13, 6, 1].

2.1 Conceptos básicos

*Un grafo $G = (V(G), E(G))$ es un par de conjuntos $V(G)$ y $E(G)$ y una relación de incidencia $iG(\cdot)$ que mapea **pares** de elementos (no necesariamente diferentes) de $V(G)$ a elementos de $E(G)$ [17]. En la figura 2.1 se muestra un ejemplo de la construcción de un grafo, donde se aprecian tanto los conjuntos V y E , así como la relación de incidencia.*

Los elementos v_i del conjunto $V(G)$ son llamados vértices (o nodos, o puntos) del grafo G , y los elementos e_j de $E(G)$ son sus aristas (o líneas). Para evitar ambigüedades, decimos que $V(G) \cap E(G) = \emptyset$ o en otras palabras, $V(G)$ y $E(G)$ son disjuntos [6].

Para simplificar la notación utilizada, se omite la relación de incidencia. Así, el grafo se redefine como $G = (V, E)$ sin mencionar explícitamente la relación

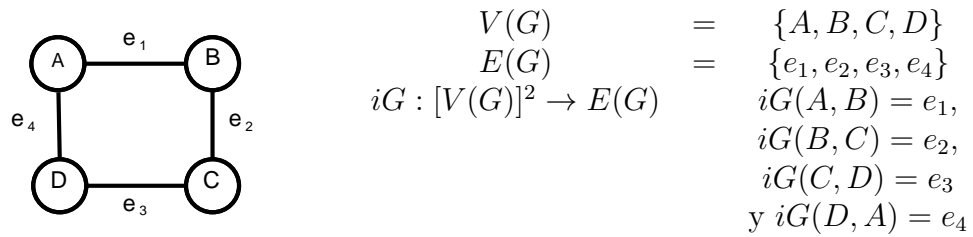


FIGURA 2.1: Ejemplo de construcción de un grafo

$iG(\cdot)$, con $E \subseteq [V]^2$, donde $[V]^n$ es el conjunto de todos los subconjuntos de V con cardinalidad n . El conjunto de vértices $V(G)$ y el conjunto de aristas $E(G)$ pueden escribirse como V y E , respectivamente. Para simplificar la notación aún más, puede denotarse a la arista (a, b) o $\{a, b\}$ como ab . La simplificación de la notación puede verse en la figura 2.2.

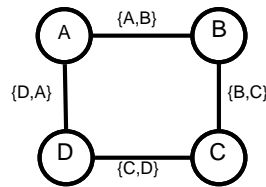


FIGURA 2.2: Grafo simplificado con $V = \{A, B, C, D\}$
 $E = \{\{A, B\}, \{B, C\}, \{C, D\}, \{D, A\}\}$, sin relación de incidencia definida.

Si las aristas son representadas como **pares ordenados** de vértices, es decir, $E \subseteq V^2$, entonces el grafo G es llamado *dirigido u orientado*, de otra manera si los pares no están ordenados, se le llama *no dirigido o no orientado* (véase figura 2.3). Los dos vértices conectados por una arista $e_k = (v_i, v_j)$ se llaman *vértices finales* o simplemente *extremos*. En los grafos dirigidos, el vértice v_i es llamado el *origen* y el vértice v_j es llamado el *objetivo o final*.

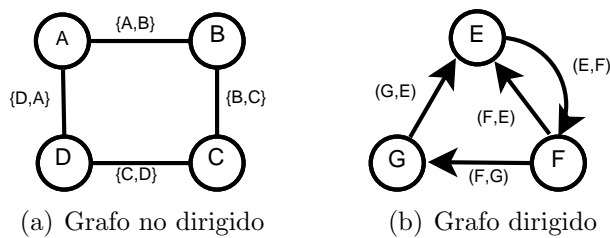


FIGURA 2.3: Representación de grafos dirigidos y no dirigidos.

A pesar de que en un conjunto los elementos repetidos no se consideran diferentes, es posible que el conjunto de aristas pueda tener más de una arista que una a

los mismos vértices. Esto se debe a que, en realidad, el conjunto de aristas sí está compuesto por elementos únicos y es la relación de incidencia la que define la unión entre vértices. La omisión de la definición explícita de la relación es la que podría generar confusión. Las aristas que tienen los mismos vértices extremos son llamadas *aristas paralelas* o *aristas dobles*. Si $e_k = (v_i, v_i)$, es decir, los extremos son los mismos, entonces e_k es llamado un *bucle* (figura 2.4(c)). Un grafo G que contiene bucles y/o aristas paralelas es un *multigrafo*. Un grafo que no tiene bucles ni aristas paralelas se llama *grafo simple*. Al número de vértices en G se le llama *orden* y se escribe como $|V|$; el número de aristas está dado por $|E|$. Un grafo de orden 0 es llamado *grafo vacío*, y de orden 1 es simplemente llamado *grafo trivial*. Un grafo es finito o infinito de acuerdo a su orden.

Dos vértices v_i y v_j son *vecinos* o *adyacentes* si son los vértices extremos de la misma arista $e_k = (v_i, v_j)$. Dos aristas e_i y e_j son *adyacentes* si tienen un vértice extremo en común, o formalmente, si se tiene v_k , $e_i = (v_k, v_l)$ y $e_j = (v_k, v_m)$. Si todos los vértices de G son vecinos, entonces G es *completo*. Un grafo completo en m vértices es escrito como K^m . Una arista es llamada *incidente* en sus vértices extremos. El grado (o valencia), denotado como $deg(v)$, de un vértice v es el número de aristas incidentes en él. Un vértice de grado 0 se llama *aislado*; de grado 1 es llamado *colgante* (figuras 2.4(a) y 2.4(b)). Un bucle en un vértice v contribuye dos veces a su grado.

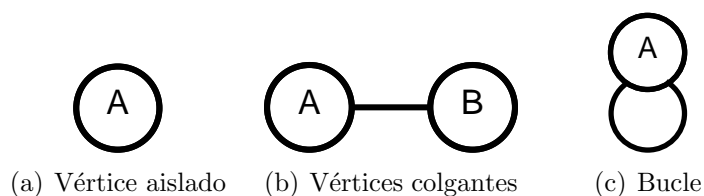


FIGURA 2.4: Tipo de vértices según su grado.

2.2 Relaciones de contención de grafos y grafos especiales

Sean $G = (V, E)$ y $G' = (V', E')$ dos grafos: $G' = (V', E')$ es un subgrafo de G ($G' \subseteq G$) si $V' \subseteq V$ y $E' \subseteq E$, esto es, el grafo G contiene a G' . El grafo G también es llamado un supergrafo de G' ($G \supseteq G'$).

Si $V' \subset V$ o $E' \subset E$, el grafo G' es llamado subgrafo propio de G . Si $G' \subseteq G$ y G' contiene todas las aristas $e = (v_i, v_j) \in E$ tal que $v_i, v_j \in V'$, entonces G' es el *grafo vértice-inducido* de G y V' induce (abarca) G' en G y se escribe como $G' = G[V']$. Dado que $V' \subset V(G)$, entonces $G[V']$ denota el grafo en V' cuyas aristas son las aristas de G con ambos extremos en V' . Si no se indica de otro modo, se refiere a subgrafo vértice-inducido por subgrafo inducido.

De forma similar, si $E' \subseteq E$ y V' se compone de los extremos de las aristas en E' , entonces $G[E']$ es llamado el *subgrafo inducido de G en el conjunto de aristas E'* o simplemente el *subgrafo arista-inducido* de G . Si $G[V']$ no tiene vértices aislados, entonces G' es también un subgrafo arista-inducido. En la figura 2.5 se muestran ejemplos de subgrafos vértice-inducido y arista-inducido. Si $G' \subseteq G$ y V' abarca todo G , esto es, $V' = V$ entonces G' es el subgrafo abarcador de G .

Un subgrafo G' de un grafo G es un subgrafo maximal (o minimal) de G con respecto a alguna propiedad Π si G' tiene la propiedad Π y G' no es un subgrafo propio de cualquier otro subgrafo propio de G con propiedad Π . Los subconjuntos minimales y maximales con respecto a alguna propiedad se definen de forma análoga.

Sea la relación de contención de conjuntos (\subset) la propiedad Π y el conjunto potencia de $\{a, b, c\}$, puede construirse un diagrama que represente las relaciones entre todos los subconjuntos de $\mathcal{P}(a, b, c)$ como lo muestra la figura 2.6. A este diagrama se le conoce como *Diagrama de Hasse* y es útil para visualizar relaciones de orden

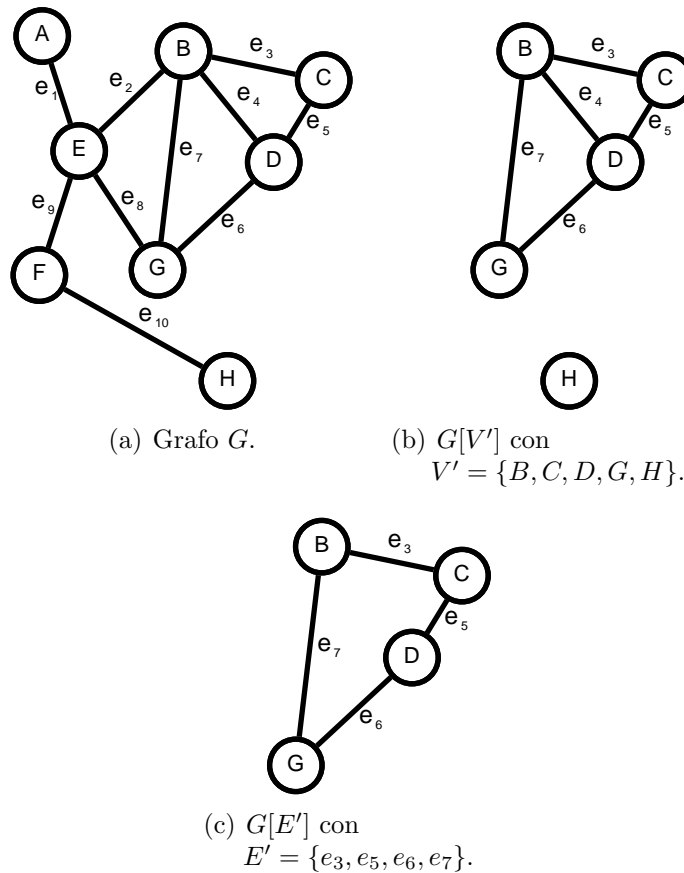


FIGURA 2.5: Grafos inducidos

bajo alguna propiedad, así como para identificar los elementos minimales y maximales (aquellos que no tienen relaciones hacia abajo o arriba respectivamente), entre otros de la misma.

Si cualquier pareja de vértices de un subgrafo $G' \subset G$ no forma una arista de G , entonces llamamos al grafo $G' = (V', \emptyset)$ un *subgrafo independiente de G* (figura 2.7). Formalmente, un subgrafo independiente $G' = (V', \emptyset)$ se define como $V' = \{v_1, v_2 \in V(G) \mid (v_1, v_2) \notin E(G)\}$.

Sean $G = (V, E)$ un grafo con conjuntos $V = \{v_1, v_2, \dots\}$ y $E = \{e_1, e_2, \dots\}$. Un *camino* en un grafo G es una secuencia no vacía, finita y alternada $v_0, e_1, v_1, \dots, v_{k-1}, e_k, v_k$ de vértices y aristas en G tal que $e_i = (v_i, v_{i+1})$ para todo $1 \leq i \leq k$. Este camino es llamado un *camino $v_0 - v_k$* , con v_0 y v_k como sus vértices terminales y todos los demás vértices son vértices internos de este camino. En un camino, aristas y vértices pueden aparecer más de una vez. Si $v_0 = v_k$, el camino es

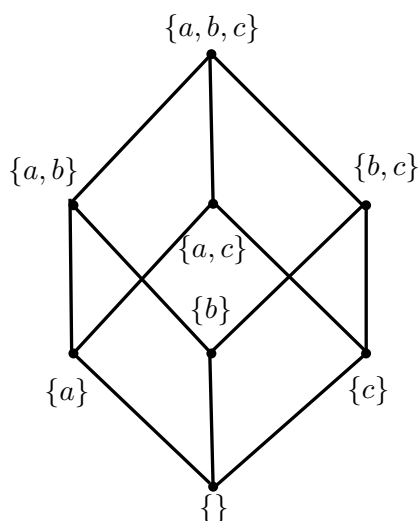


FIGURA 2.6: Representación visual de la relación entre subconjuntos con propiedad II.

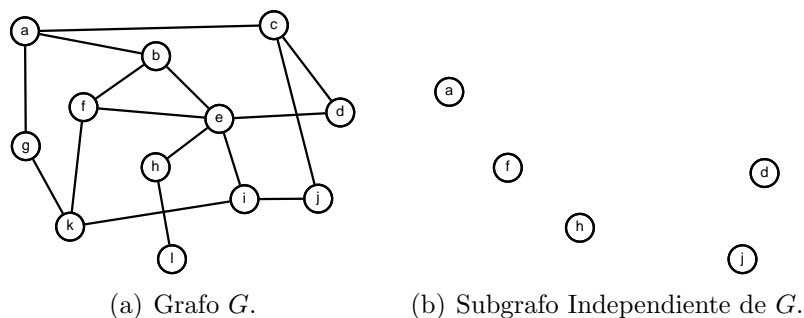


FIGURA 2.7: Ejemplo de subgrafo independiente.

cerrado, de otra manera es *abierto*. Un camino es un *rastro* o *trayectoria* si todas sus aristas son distintas. Una trayectoria es cerrada si sus vértices finales son los mismos, de otra manera, es abierta. Por definición, el camino puede contener el mismo vértice muchas veces. Una *ruta* P es una trayectoria donde todos los vértices son distintos. Estas sutiles diferencias entre caminos, trayectorias y rutas pueden verse claramente en la figura 2.8. Una ruta simple puede escribirse como $P = v_0, v_1, v_2, \dots, v_k$, donde las aristas no están descritas explícitamente ya que en una ruta, todos los vértices son distintos y por tanto, todas las aristas también lo son.

Note que en un multigrafo, una ruta **no** está definida, debido a posibles aristas múltiples entre 2 vértices. Si los vértices v_0 y v_k están ligados por la ruta P ,

cuando hay una función f biyectiva entre $V(G)$ y $V(G')$ ($f : V(G) \rightarrow V(G')$), tal que cualquier par de vértices v y w son adyacentes si y sólo si los vértices $f(v)$ y $f(w)$ también son adyacentes, es decir:

$$\forall e = vw \in E(G) : e' = f(v)f(w) \in E(G')$$

Si dos grafos son isomorfos, se denota como $G \sim G'$. En la tabla 2.1, se muestra la función biyectiva construida que verifica el isomorfismo entre los grafos definidos en la figura.

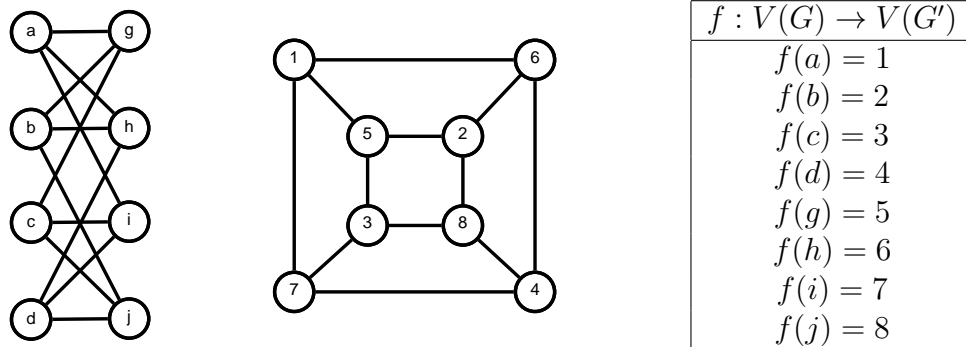


TABLA 2.1: Isomorfismo entre dos grafos

2.3 Conectividad

Determinar las condiciones de conectividad entre vértices y en el mismo grafo es una herramienta sumamente importante. Dos vértices v_i y v_j están conectados en un grafo $G = (V, E)$ si existe una ruta $v_i - v_j$ en G . Un vértice siempre está conectado consigo mismo.

Un grafo no vacío es conexo o conectado si cualesquiera dos vértices están unidos por una ruta en G , es decir, si todos los posibles pares de vértices están conectados (figura 2.10).

Sea $G = (V, E)$ un grafo no conexo, el conjunto de vértices V puede ser particionado en subconjuntos V_1, V_2, \dots, V_p tales que cada subgrafo vértice-inducido $G[V_i]$ sea conexo, y no exista una ruta entre un vértice en el subconjunto V_i y un

vértice en V_j , con $j \neq i$. Un subgrafo conexo maximal de G es llamado un *componente* del grafo G . Un componente de G no es un subgrafo propio de cualquier otro subgrafo propio de G . Un vértice aislado es considerado un componente, ya que, por definición está conectado consigo mismo. Note que un componente siempre es no vacío y que, si un grafo G es conexo entonces tiene sólo un componente, él mismo.

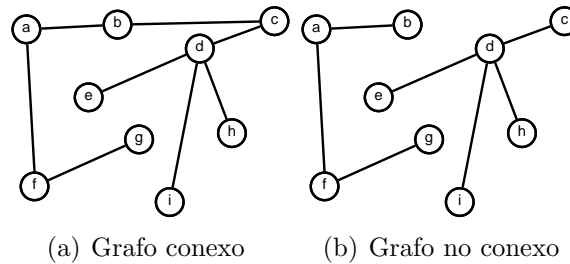


FIGURA 2.10: Conectividad en grafos

Sean $\{V_1, V_2\}$ particiones del conjunto de vértices V de un grafo $G = (V, E)$. El conjunto $\mathcal{K}(V_1, V_2)$ llamado *corte* [12] o *cociclo* [6], se define como el conjunto de todas las aristas teniendo un extremo en una partición de vértices (V_1) y el otro extremo en la segunda partición de vértices (V_2), como se aprecia en la figura 2.11.

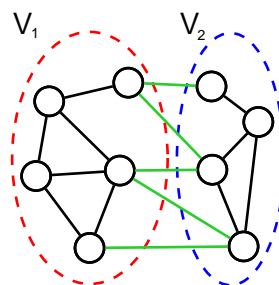
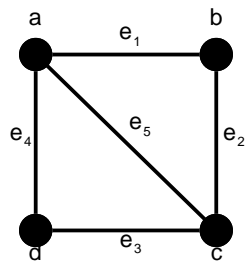


FIGURA 2.11: Las aristas en verde representan $\mathcal{K}(V_1, V_2)$.

Un *conjunto de corte* \mathcal{K}_S de un grafo conexo G es un conjunto minimal *no vacío* de aristas tal que su remoción de G desconecta G , es decir, $G - \mathcal{K}_S$ es desconexo, tal y como se muestra en la tabla 2.2 y la figura 2.12. Si los subgrafos inducidos de G en los conjuntos de vértices V_1 y V_2 son conexos, entonces $\mathcal{K} = \mathcal{K}_S$. Sea un conjunto de vértices $V_1 = \{v\}$, el conjunto de corte se denota por $\mathcal{K}(v)$.



V_1	V_2	$\mathcal{K}(V_1, V_2)$
$\{a\}$	$\{b, c, d\}$	$\{e_1, e_4, e_5\}$
$\{b\}$	$\{a, c, d\}$	$\{e_1, e_2\}$
$\{c\}$	$\{a, b, d\}$	$\{e_2, e_3, e_5\}$
$\{d\}$	$\{a, b, c\}$	$\{e_3, e_4\}$
$\{a, b\}$	$\{c, d\}$	$\{e_2, e_4, e_5\}$
$\{a, c\}$	$\{b, d\}$	$\{e_1, e_2, e_3, e_4\}$
$\{b, c\}$	$\{a, d\}$	$\{e_1, e_3, e_5\}$

TABLA 2.2: Cortes de un grafo G

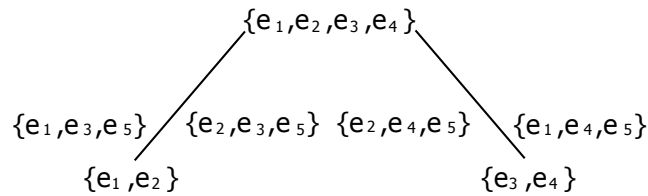
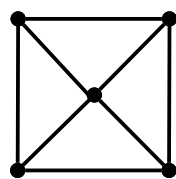


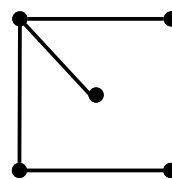
FIGURA 2.12: El Diagrama de Hasse muestra fácilmente los conjuntos de corte de G.

2.4 Árboles

Los árboles son estructuras simples de grafos. Un grafo G es acíclico si no tiene ciclos. Un árbol del grafo G es un subgrafo conexo acíclico de G . Vértices de grado 1 en un árbol son llamados *hojas* y todas las aristas son llamadas *ramas*. Un árbol no trivial tiene, por lo menos, dos hojas y una rama. Por ejemplo, el árbol más simple consiste de dos vértices unidos por una arista. Note que un vértice aislado es por definición un grafo conexo acíclico y, por tanto, es un árbol. Un árbol abarcador o recubridor de G es un árbol de G que contiene todos los vértices de G . Las aristas de árboles recubridores se llaman *ramas* (figura 2.13).



(a) Grafo G.



(b) Árbol abarcador de G.

FIGURA 2.13: Árboles.

Un grafo acíclico con k componentes se llama un k -árbol. Si un k -árbol es un subgrafo abarcador de G , entonces es llamado un k -árbol abarcador de G . Un *bosque*

F de un grafo G es un k -árbol abarcador, donde k es el número de componentes en G . Un *bosque* es simplemente un conjunto de árboles, abarcando todos los vértices de G . Un subgrafo conexo de un árbol T es llamado un subárbol de T . Si T es un árbol entonces hay exactamente una sola ruta entre cualesquiera dos vértices de T . Cuando un árbol posee un nodo o vértice especial de referencia llamado *raíz*, se le llama *árbol enraizado*. La profundidad de un árbol enraizado es la máxima distancia (número de aristas) entre la raíz y todas sus hojas (figura 2.14).

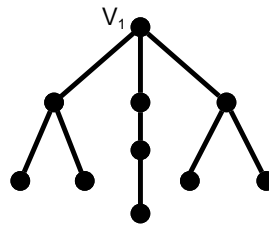


FIGURA 2.14: Árbol con nodo V_1 como raíz y profundidad 3.

2.5 Operaciones en grafos

Sean $G = (V, E)$ y $G' = (V', E')$ dos grafos.

- *Unión e intersección.* La *unión* de G y G' es el grafo $G'' = G \cup G' = (V \cup V', E \cup E')$, esto es, el conjunto de vértices de G'' es la unión de V y V' , y el conjunto de aristas de G'' es la unión de E y E' , respectivamente (figura 2.15(b)). La *intersección* de G y G' es el grafo $G'' = G \cap G' = (V \cap V', E \cap E')$, esto es, el conjunto de vértices de G'' tiene sólo los vértices presentes en V y V' , y el conjunto de aristas de G'' tiene sólo las aristas presentes en E y E' , respectivamente (figura 2.15(c)).
- *Diferencia Simétrica.* La diferencia simétrica entre dos grafos G y G' , escrito como $G \oplus G'$, es el grafo G'' inducido por el conjunto de aristas $E \boxplus E' = (E \setminus E') \cup (E' \setminus E)$, esto es, el grafo no tiene vértices aislados y contiene aristas presentes en G o en G' pero no en ambos (figura 2.16).
- *Remoción o eliminación de vértices.* Sea $v_i \in G$, entonces $G - v_i$ es el subgrafo inducido de G en el conjunto de vértices $V - v_i$; esto es, $G - v_i$ es el

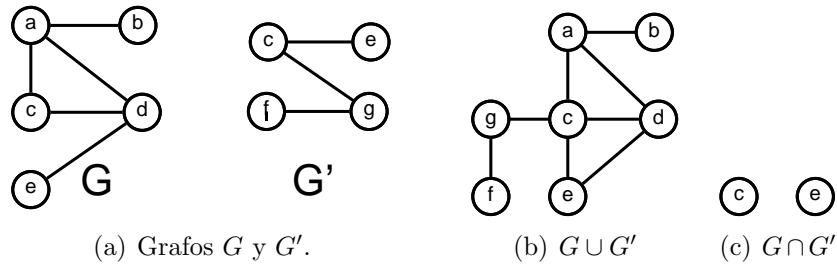


FIGURA 2.15: Unión e intersección.

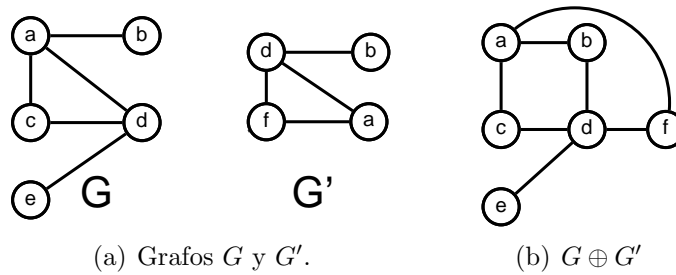


FIGURA 2.16: Diferencia simétrica.

grafo obtenido después de remover el vértice v_i y todas las aristas $e_j = (v_i, v_j)$ incidentes en v_i . La eliminación de un conjunto de vértices de un grafo se hace como la sucesión de la eliminación de un solo vértice (figura 2.17).

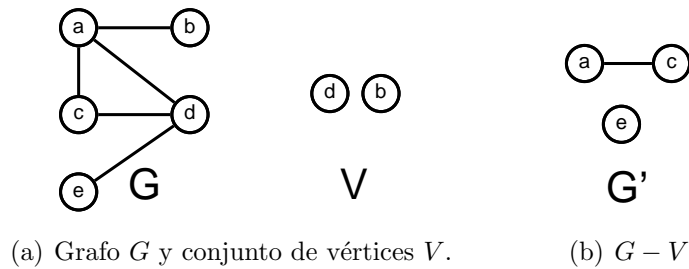


FIGURA 2.17: Eliminación de vértices.

- *Remoción o eliminación de aristas.* Sea $e \in G$, entonces $G - e$ es el subgrafo obtenido después de remover la arista e de E . Los vértices finales de la arista $e_j = (v_i, v_j)$ no son eliminados. La remoción de un conjunto de aristas de un grafo se hace como la sucesión de la eliminación de una sola arista (figura 2.18).

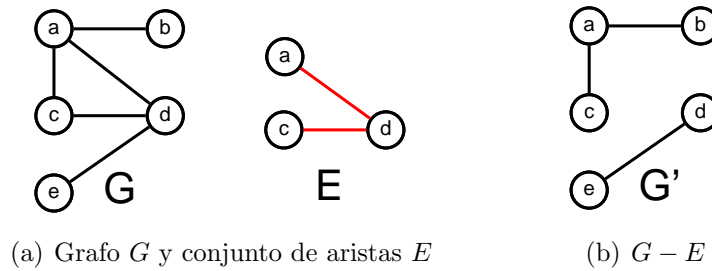


FIGURA 2.18: Eliminación de aristas.

- *Identificación de vértices.* Sean v_i y v_j dos vértices distintos de un grafo G unidos por la arista $e_j = (v_i, v_j)$. Dos vértices v_i y v_j son identificados si son reemplazados por un nuevo vértice v^* tal que todos los vértices incidentes en v_i y v_j ahora son incidentes en el nuevo vértice v^* (figura 2.19(b)).
- *Contracción de aristas.* Sea $e_j = (v_i, v_j) \in G$ una arista con extremos distintos $v_i \neq v_j$ a ser contraída. La operación de contracción de aristas denota la eliminación de la arista e e identificar sus vértices v_i y v_j por un nuevo vértice v^* . Si un grafo G' resulta de G después de contraer una secuencia de aristas, entonces se dice que G puede ser contraído a un grafo G' (figura 2.19(c)).

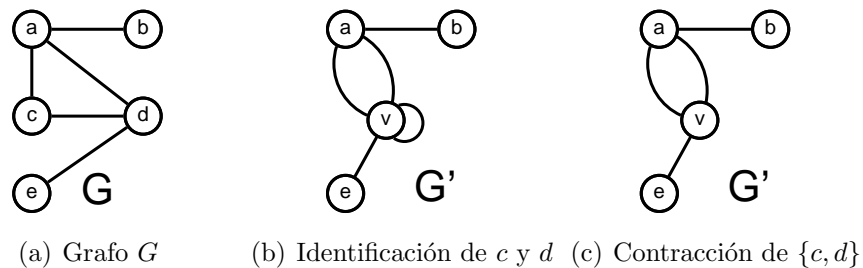


FIGURA 2.19: Identificación de vértices y contracción de aristas

2.6 Álgebra lineal

Es posible la aplicación de los conceptos de espacio vectorial para describir un grafo [6]. Esto nos proporciona ventajas como flexibilidad al aplicar operaciones

entre grafos. \mathbb{V} es un espacio vectorial sobre el campo \mathbb{K} con las operaciones $+$: $\mathbb{V} \times \mathbb{V} \rightarrow \mathbb{V}$ y \cdot_{λ} : $\mathbb{K} \times \mathbb{V} \rightarrow \mathbb{V}$, si se cumplen las siguientes propiedades:

- $\forall v \in \mathbb{V}, \exists 0^* \in \mathbb{V} : v + 0^* = 0^* + v = v$
- $\forall v \in \mathbb{V}, \exists v^* \in \mathbb{V} : v + v^* = 0^*$ y denotamos $v^* = -v$.
- $\forall v, w \in \mathbb{V} : v + w = w + v$
- $\forall u, v, w \in \mathbb{V} : u + (v + w) = (u + v) + w$
- $\forall v, w \in \mathbb{V}, \forall \lambda \in \mathbb{K} : \lambda(v + w) = \lambda v + \lambda w$
- $\forall v \in \mathbb{V}, \forall \lambda, \gamma \in \mathbb{K} : (\lambda + \gamma)v = \lambda v + \gamma v$
- $\forall v \in \mathbb{V}, \forall \lambda, \gamma \in \mathbb{K} : (\lambda\gamma)v = \lambda(\gamma v)$
- $\forall v \in \mathbb{V}, 1 \bullet v, 1 \in \mathbb{K}$

Los elementos de \mathbb{V} se llaman *vectores* y los elementos de \mathbb{K} se llaman *escalares*. Sea un conjunto de vectores $\{v_1, v_2, \dots, v_n\}$, se dice que el conjunto es *linealmente independiente* si la ecuación $\lambda_1 v_1 + \lambda_2 v_2 + \dots + \lambda_n v_n = 0$ tiene como solución **única** la trivial, es decir, $\lambda_1 = \lambda_2 = \dots = \lambda_n = 0$. Si existe una solución distinta a la trivial, se dice que el conjunto $\{v_1, v_2, \dots, v_n\}$ es *linealmente dependiente*.

Un conjunto de vectores $\{v_1, v_2, \dots, v_n\}$ de un espacio vectorial \mathbb{V} sobre un campo \mathbb{K} se llama *base* de \mathbb{V} si es linealmente independiente y genera todo el espacio \mathbb{V} , es decir, $\forall v \in \mathbb{V} : \exists \lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{K}, v = \lambda_1 v_1 + \lambda_2 v_2 + \dots + \lambda_n v_n$. Si $\{v_1, v_2, \dots, v_n\}$ es una base de \mathbb{V} sobre \mathbb{K} , decimos que \mathbb{V} es de dimensión finita y la dimensión de \mathbb{K} es n , denotado como $\dim \mathbb{K} = n$.

Sea (\mathbb{V}, \mathbb{K}) un espacio vectorial, $\mathbb{W} \subseteq \mathbb{V}$ es un subespacio vectorial de \mathbb{V} si para cualquier pareja v, w de vectores en \mathbb{W} y escalares λ, γ en \mathbb{K} , se cumple que $\lambda v + \gamma w \in \mathbb{W}$. A su vez, el espacio generado por un conjunto de vectores $\{v_1, v_2, \dots, v_n\}$ de un espacio vectorial \mathbb{V} sobre un campo \mathbb{K} se define como $\{\lambda_1 v_1 + \lambda_2 v_2 + \dots + \lambda_m v_m : \lambda_i \in \mathbb{K}\}$ y se denota como $gen(\{v_1, v_2, \dots, v_n\})$.

Dado que el conjunto de aristas describen la estructura fundamental de un grafo, usar los principios del álgebra lineal resulta más ventajoso. Sea $\mathcal{P}(E)$ el conjunto potencia de $E(G)$, podemos dotarlo de una estructura de espacio vectorial sobre el campo $\mathbb{F} = \{0, 1\}$, con la suma $+$ definida como $0 + 1 = 1 + 0 = 1$, $1 + 1 = 0$, el producto \bullet definido como $0 \bullet 1 = 1 \bullet 0 = 0$, $1 \bullet 1 = 1$ y definiendo $\mathbb{V} = \mathcal{P}(E)$, $\mathbb{K} = \mathbb{F}$ con $\{e_1, e_2, \dots, e_k\} \in \mathbb{V}$ siendo vectores y \mathbb{F} el conjunto de escalares.

Las operaciones en \mathbb{V} son:

- $+$: $\mathcal{P}(E) \times \mathcal{P}(E) \rightarrow \mathcal{P}(E)$, donde $\mathcal{V} + \mathcal{W} = \mathcal{V} \oplus \mathcal{W} = (E \setminus E') \cup (E' \setminus E)$ y $\mathcal{V}, \mathcal{W} \in \mathcal{P}(E)$.
- \cdot_λ : $\mathbb{F} \times \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ donde
 - $1 \cdot_\lambda \mathcal{V} = \mathcal{V}$
 - $0 \cdot_\lambda \mathcal{V} = \emptyset$ donde $\mathcal{V} \in \mathcal{P}(E)$.

De esta manera, es posible realizar una suma de vectores de $\mathcal{P}(E)$ rápidamente, sin tener que recurrir a operaciones de conjuntos.

Sea $E = \{e_1, e_2, e_3, e_4\}$, entonces $\{e_1, e_2, e_3\} + \{e_2, e_3, e_4\} = e_1 + (e_2 + e_2) + (e_3 + e_3) + e_4 = e_1 + e_4 = \{e_1, e_4\}$.

Sea $\mathcal{C}(G)$ el espacio generado por el conjunto de ciclos de G , se cumple que $\lambda v + \gamma w \in \mathcal{C}(G) : \forall \lambda, \gamma \in \mathbb{F}$ y $\forall v, w \in \mathcal{C}(G)$. Así, podemos decir que $\mathcal{C}(G)$ es un subespacio de $\mathcal{P}(E)$.

2.7 Grafos planos y duales

Los *grafos planos* particionan a \mathbb{R}^2 en regiones llamadas caras. Los grafos planos junto a los conceptos de ciclos y cortes definen formalmente a un *grafo Dual*.

Para describir adecuadamente la estructura de los grafos planos, es necesario definir previamente ciertos conceptos topológicos. Un *segmento de recta* en el plano Euclidiano es un subconjunto de \mathbb{R}^2 con la forma $\{p + \lambda(q - p) \mid 0 \leq \lambda \leq 1\}$

para puntos diferentes p, q de \mathbb{R}^2 . El *interior* de un segmento de recta es el conjunto $\{p + \lambda(q - p) \mid 0 < \lambda < 1\}$. Un *arco* es la unión finita de varios segmentos de recta.

Un grafo \tilde{G} con conjunto finito de vértices V y aristas E es llamado un *grafo plano* si puede ser dibujado en un plano en \mathbb{R}^2 tal que:

- $V \subset \mathbb{R}^2$
- Toda arista es un arco entre dos vértices.
- Ningún par de aristas se cruza, es decir, el interior de una arista no contiene ningún vértice ni punto de cualquier otra arista.

Las regiones conectadas de $\mathbb{R}^2 \setminus \tilde{G}$ son *caras* f de \tilde{G} . Dado que \tilde{G} tiene borde, una de sus caras es *no acotada*, es decir, tiene área infinita y se le llama *cara externa*. Las caras restantes encierran áreas finitas y son llamadas *caras internas*. Las aristas y vértices incidentes a una cara se llaman *fronteras* de esa cara.

Una *incrustación planar* de un grafo G es un isomorfismo entre G y un grafo plano \tilde{G} . A G se le llama *grafo planar*. De forma similar a \tilde{G} , G se dibuja de forma que sus aristas sólo se crucen en sus vértices. Puede decirse que un grafo G es *planar* si puede ser *incrustado* en el plano. Nótese que el concepto de incrustación puede extenderse a cualquier superficie, es decir, un grafo G es **incrustable** en una superficie S si puede dibujarse en S de manera que sus aristas sólo se intersequen en sus vértices. Un grafo planar y su incrustación se ilustran en la figura 2.20.

Sea $G = (V, E)$ un grafo planar y que pueda tener, en general, aristas paralelas y bucles, construimos un nuevo grafo $\bar{G} = (\bar{V}, \bar{E})$, denotamos a cada una de sus caras por un nuevo vértice $\bar{v} \in \bar{V}$ y los colocamos dentro de cada una de las caras de G (ahora la noción de los vértices en cada cara y el término cara se usará sin distinción entre ellos). Si un par de caras (\bar{v}_i, \bar{v}_j) inciden en la misma arista $e_i \in E$, éstas se conectan por una arista \bar{e}_i , de manera que e_i y \bar{e}_i se crucen. Así, para cada arista $e_i \in G$, hay una arista \bar{e}_i del grafo \bar{G} , que recibe el nombre de *Grafo Dual de G* . Formalmente, un grafo $\bar{G} = (\bar{V}, \bar{E})$ es un dual de $G = (V, E)$ si existe

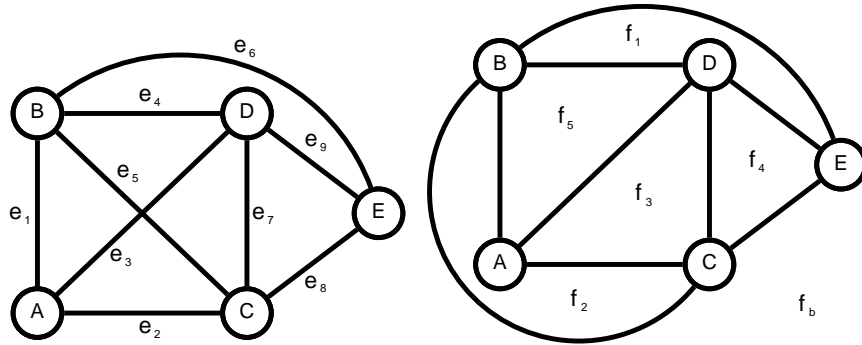


FIGURA 2.20: Grafo Planar G (izquierda) y su incrustación planar, el grafo plano \tilde{G} (derecha).

una biyección entre las aristas de G y \bar{G} tal que un conjunto de aristas en \bar{G} es un *ciclo* si y sólo si el correspondiente conjunto de aristas en G es un *conjunto de corte*. El grafo G es llamado el *grafo primal* y a \bar{G} se le llama el *grafo dual*. Si un grafo plano G' es el dual de G , entonces G es el dual de G' [1, 6].

Formalmente puede definirse la dualidad de grafos mediante los conceptos de ciclo y cortes. Un grafo $\bar{G} = (\bar{V}, \bar{E})$ es dual de $G = (V, E)$ si existe una biyección entre las aristas de G y \bar{G} tal que para cada conjunto de aristas en \bar{G} que sea un ciclo (simple, sin ciclos internos), su correspondiente en G sea un corte.

Los grafos duales poseen dos propiedades importantes que son esenciales para la simplificación que se necesita al construir la pirámide imagen irregular: Un grafo y su dual son también duales después de remover una arista e en el grafo primal y contraer la arista correspondiente \bar{e} en el grafo dual \bar{G} . Además, un grafo es plano si y sólo si tiene un dual [1]. En la figura 2.21 se observa la construcción de un grafo dual a partir de su primal.

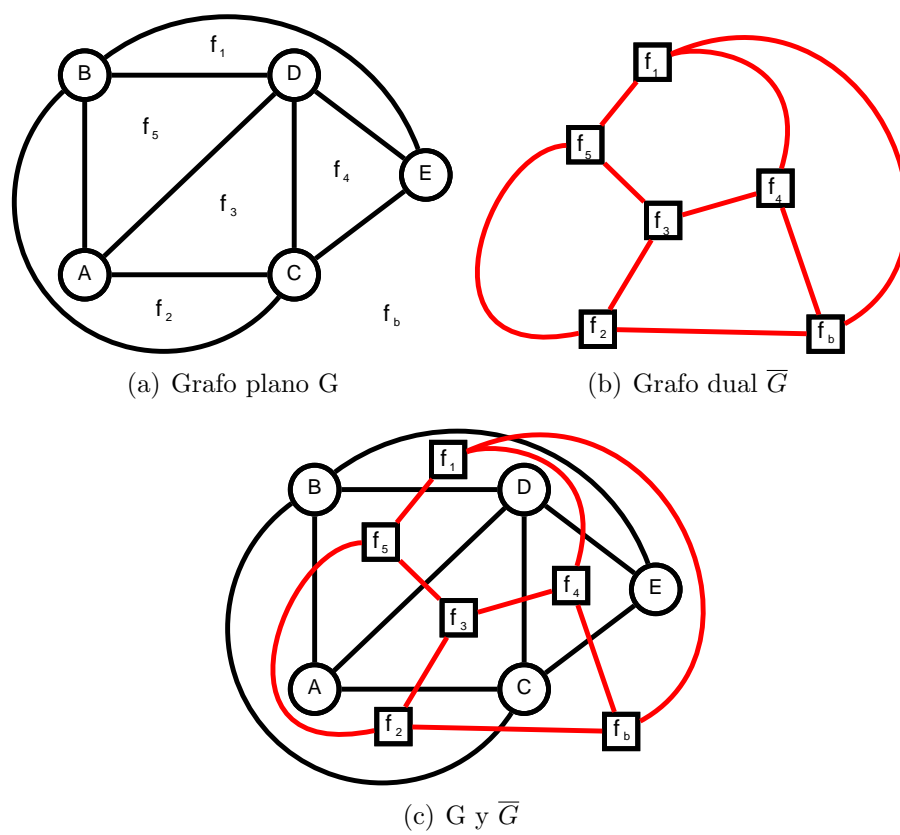


FIGURA 2.21: Grafos duales

Capítulo 3

Métodos de segmentación

La segmentación es muy importante para los problemas de visión por computadora y reconocimiento de patrones, entre otros. Existen una gran variedad de técnicas ampliamente estudiadas, debido al uso de las imágenes digitales en virtualmente todos los campos de la ciencia y a su vez, porque los requerimientos de la segmentación son muy diferentes en cada posible campo de aplicación [22, 21, 1].

En la medicina, el diagnóstico auxiliado por imágenes ha sido una herramienta invaluable que ha proporcionado grandes conocimientos sobre anatomía. Las distintas modalidades para obtener estas imágenes han ido aumentando tanto en tipos (tomografías, mamografías, ultrasonidos, resonancias magnéticas, escáneos por emisión de positrones, etc.), como en el tamaño y número de éstas. Esto, en consecuencia, ha aumentado el uso de las computadoras para facilitar el procesamiento e incluso auxiliar en el análisis de las imágenes. Actualmente, es típico que se utilicen algoritmos para una gran cantidad de tareas en el área de la medicina, como son: Cuantificación de volumen de tejidos, estudio y delineamiento de estructuras anatómicas, localización y diagnóstico de patologías, planeación de tratamientos e incluso asistencia en cirugías.

Los valores numéricos que contienen las imágenes de naturaleza médica - visualmente interpretadas como color - representan distintos fenómenos como absorción de radiación en las radiografías, presión acústica en los ultrasonidos o la amplitud

de una señal de radiofrecuencia en una resonancia magnética. Se busca idealmente agrupar estos valores, mediante alguna técnica de segmentación, y posteriormente etiquetar cada región con un nombre significativo. Por ejemplo, haciendo referencia a una mamografía, se busca localizar y separar el tejido saludable del canceroso y etiquetarlos según corresponda [4, 7]. No obstante, existen dificultades que van más allá de la naturaleza de las imágenes. Entre ellas están la existencia de ruido o la presencia de artefactos que alteran la imagen y disminuyen el rendimiento y la efectividad de los métodos de segmentación que dan por hecho que la intensidad de cada clase de tejido es constante. Es decir, dan por hecho que los diferentes tejidos que representan la misma etiqueta (tejido sano, tejido canceroso, etc.) siempre van a poseer el mismo valor (o rango) numérico. Estos artefactos aparecen debido a las limitaciones físicas de los escáneres usados. Por ejemplo, en el caso de las imágenes de resonancia magnética los artefactos que se presentan toman la forma de sombras que cambian la intensidad de los tejidos [7].

Esta sección se auxilia fuertemente de los estudios realizados en [4, 7], especializados en la comparación de los diferentes métodos de segmentación utilizados en el campo médico. También se toman como fuentes otras recopilaciones de métodos de segmentación aplicados tanto a la medicina ([2]), como a otros ámbitos ([3]).

3.1 Tipos de segmentación

Las técnicas para realizar la segmentación tienen grandes variaciones en la eficacia y precisión de sus resultados dependiendo del tipo de imagen médica en las que se apliquen. Una técnica puede tener buenos resultados al localizar y etiquetar distintos tipos de tejido cerebral pero podría ser incapaz de reconocer tejido dañado en el hígado u otro tipo de tejidos.

En la literatura existen varias clasificaciones vigentes de la segmentación [4, 7, 21, 2]. Una de estas es la división entre la segmentación suave y dura. La segmentación suave hace referencia a la división de la imagen en regiones o clases

que pueden traslaparse. Esto suele suceder cuando diferentes tipos de tejido contribuyen al valor numérico de un mismo píxel. Este tipo de segmentación retiene más información de la imagen original, al permitir cierta incertidumbre en la localización de los bordes o fronteras entre objetos. Se utilizan funciones características o funciones que midan o cuantifiquen la calidad de un dato para ser miembro de alguna clase con el fin de relacionar un píxel a una región (ya sea simplemente para determinar si un píxel está o no en esa región o para especificar cuantitativamente su contribución en la misma).

Por otro lado, la segmentación dura no permite estos traslapes y fuerzan a decidir a qué región pertenece un píxel. Es posible convertir una técnica de segmentación suave a una dura simplemente eligiendo la región a la que un píxel hace mayor contribución.

En general, la técnicas de segmentación pueden ser representadas como un problema de optimización, donde la segmentación deseada minimiza una función de energía o costo [4, 7, 21].

Los distintos métodos de segmentación pueden dividirse en varias categorías, dependiendo de la clase de técnicas que se utilicen en su implementación y se detallan a continuación.

3.1.1 Umbralización y crecimiento de regiones

Los enfoques con umbralización realizan la segmentación haciendo un particionamiento de las intensidades de una imagen. Para esto, se contabiliza la frecuencia de las intensidades en los píxeles de la imagen (histograma) y se encuentra una característica (un valor de intensidad), con base en esta medición, que segmente la imagen, generalmente se utiliza el contraste entre las intensidades. Cada valor encontrado que represente la frontera entre regiones se le llamaba *umbral* y los píxeles se agrupan en regiones comparando su intensidad con algún umbral. Aunque esta técnica es sencilla, es susceptible a ruido y por esta razón, rara vez se usa sola y es más bien aplicada como un paso inicial en un procesamiento de imágenes más

robusto. En la figura 3.1(b) puede observarse el resultado de segmentar la figura 3.1(a) mediante un algoritmo de umbralización.

El crecimiento de regiones, como su nombre lo indica, consiste en la elección de un punto inicial llamado semilla (de forma manual por un operador) y en el crecimiento de esa región extrayendo los píxeles conectados a esta por algún criterio previamente definido, como un valor similar o igual de intensidad. Al igual que la umbralización, el crecimiento de regiones es susceptible al ruido pues puede provocar la desconexión de píxeles que en condiciones normales no ocurriría. El resultado de esta técnica se muestra en la figura 3.1(c).

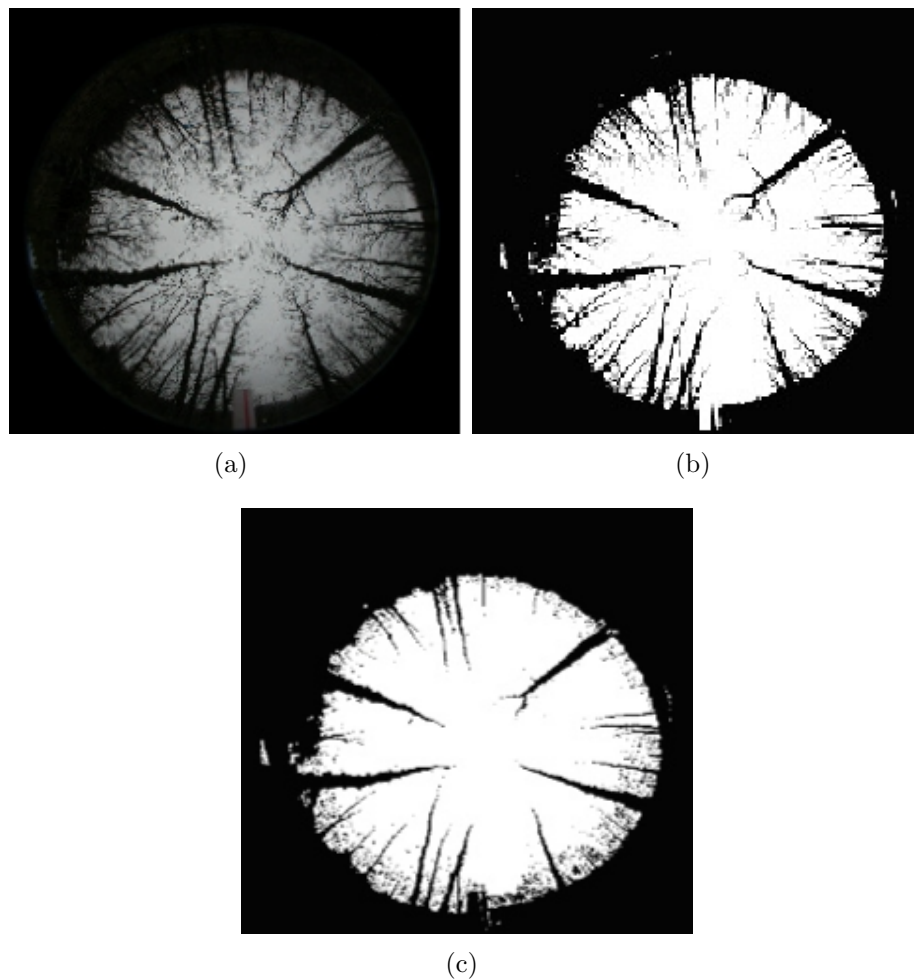


FIGURA 3.1: Imagen original (a), segmentada por umbralización (b) y por crecimiento de regiones (c) [3].

3.1.2 Clasificadores y agrupadores

Los clasificadores son técnicas de reconocimiento de patrones que buscan dividir un espacio de características derivado de la imagen usando datos con etiquetas ya conocidas. Estas etiquetas son datos que ya han sido proporcionados por un operador; que haya segmentado previamente la imagen de forma manual, para ser usadas como referencia en futuras segmentaciones, por lo que a los clasificadores se les llaman “métodos supervisados”. Existe una gran variedad de métodos clasificadores, por ejemplo, en los que los datos suministrados son las intensidades representativas de una clase y se agrupan los píxeles con la intensidad más cercana a esta.

Los algoritmos de agrupación o agrupadores funcionan de la misma manera que los métodos clasificadores, pero sin la necesidad de suministrar datos de entrenamiento. Por esto, en contraste, se les llaman “métodos no supervisados”. Para compensar la carencia de datos de entrenamiento, parte del procesamiento se centra en caracterizar las propiedades de cada clase, es decir, se entrenan a sí mismos. Uno de los algoritmos de agrupación más comunes es el K-medias, en el que se agrupan datos, calculando de manera iterativa las intensidades medias de cada clase y agrupando los píxeles con la intensidad más cercana. Los resultados de dicho algoritmo se observan en la figura 3.2 con los resultados vistos en [4, 7].

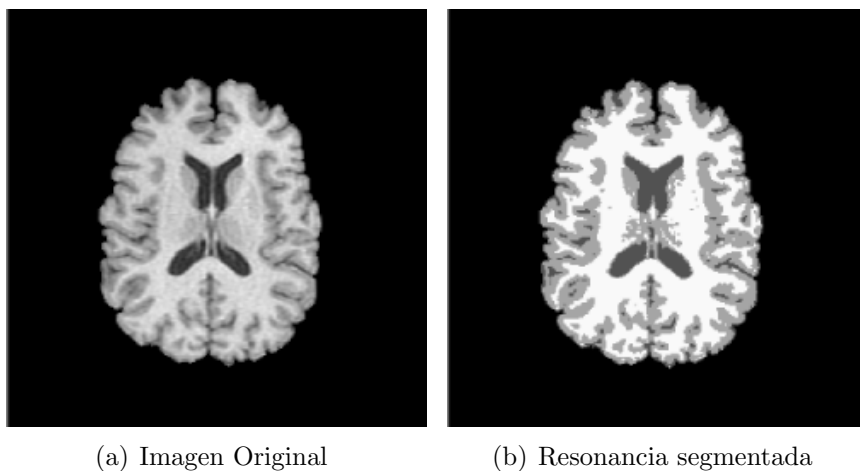


FIGURA 3.2: Segmentación con K-medias [4].

3.2 Funcional Mumford y Shah

Como se mencionó anteriormente, muchos de los métodos de segmentación consisten en la minimización de una función de energía o costo. Entre éstas se encuentra una técnica de gran importancia y popularidad, cuyos resultados sirven de referencia a este proyecto (por ser segmentaciones en imágenes médicas), llamada la funcional de Mumford y Shah. Esta técnica consiste en aproximar la imagen original desde un conjunto finito de imágenes simples (regiones) separados por bordes lisos (discontinuidades). [20]

La minimización sobre curvas en un plano es un proceso pesado desde el punto de vista numérico por eso, se ha realizado un gran esfuerzo de investigación para encontrar las formas numéricas más eficientes para minimizar la funcional de Mumford-Shah. Entre estas, se han empleado ecuaciones diferenciales parciales, aproximación elíptica, evolución de curvas, etc.

Debido a la frecuente aparición de patrones fractales que se dan en la naturaleza (como notó y postuló Mandelbrot en [14]), en [2], se propone una modificación de la funcional Mumford-Shah mediante la consideración del análisis fractal a ésta. Las imágenes médicas no son la excepción pues la estructura cerebral, entre otros órganos, posee estos mismos patrones y los resultados obtenidos de la segmentación de la funcional modificada fueron una mejora de los obtenidos con la funcional convencional. En la figura 3.3 se comparan los resultados de la segmentación con la funcional Mumford-Shah estándar y aplicando análisis fractal sobre una imagen de un ojo obtenida mediante una retinografía.

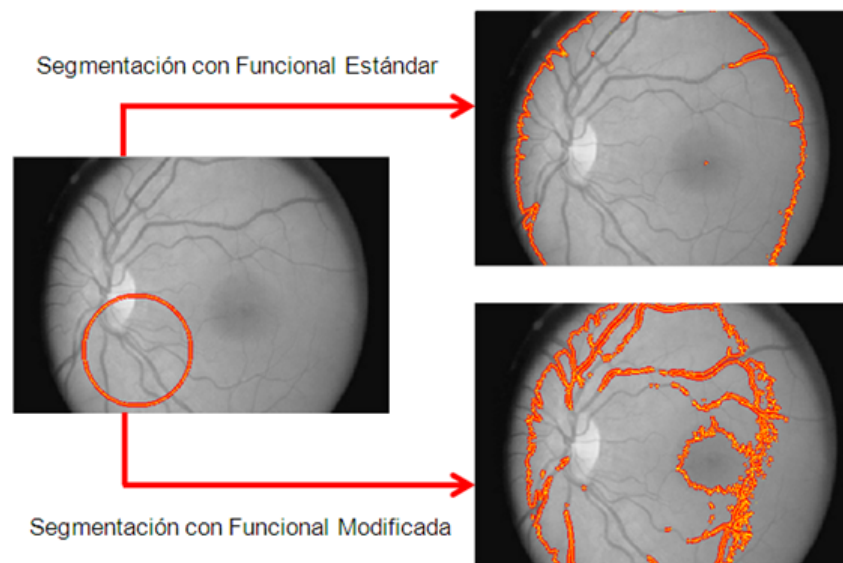


FIGURA 3.3: Segmentación usando la funcional de Mumford y Shah estándar y con análisis fractal [2].

Capítulo 4

Grafos Piramidales

Las imágenes son caracterizadas por una cantidad masiva de datos con alta redundancia. Es decir, para describir la estructura de una imagen es necesario suministrar desde miles hasta millones de números que representen colores y estos valores tienden a repetirse con frecuencia. La información relevante se encuentra agrupada en el espacio (en una sola imagen) o tiempo (en una secuencia de imágenes). Ésto se traduce en la búsqueda de métodos y reglas de organización para hacer frente a la inmensa complejidad computacional presente en el tratamiento de los datos acumulados y para unir la brecha entre los datos en bruto y la descripción simbólica de los elementos de la imagen. El desarrollo de esta capítulo se apoya fuertemente de [1, 12, 17].

4.1 Representación Digital de Color

El propósito de un modelo de color (también llamado espacio o sistema de color) es el de facilitar la especificación de colores de alguna forma generalmente aceptada. En esencia, un modelo o espacio de colores es una especificación de un sistema coordinado y un subespacio dentro de ese sistema, donde cada color se representa por un solo punto. En la figura 4.1 se puede apreciar la distribución y, al mismo tiempo, la semejanza entre los colores en cada uno de los distintos modelos [22].

La mayoría de los modelos de color en uso están orientados hacia la facilidad de implementación en el hardware. Los más comunes son el modelo *RGB* usado para los monitores y muchos tipos de cámaras, el modelo *CMYK* usado para la impresión a color. Los espacios de color que más se asemejan a la interpretación humana de la visión (pero aún es ampliamente discutido) son el *HSI* y *HSV*, pues separan la información del color (tonalidad) y la intensidad de éste [22].

El proceso de selección de modelos de color se vuelve importante cuando se quieren encontrar numéricamente píxeles de intensidades similares. Dado que los modelos de color son un sistema coordenado, la similaridad entre colores está dada como la distancia entre dos puntos y esta medida provista por algunos modelos no concuerda en muchos los casos con lo que el ojo humano percibe como semejanza. Debido a este problema, la selección de un modelo de color es, en general, un problema difícil [12, 25].

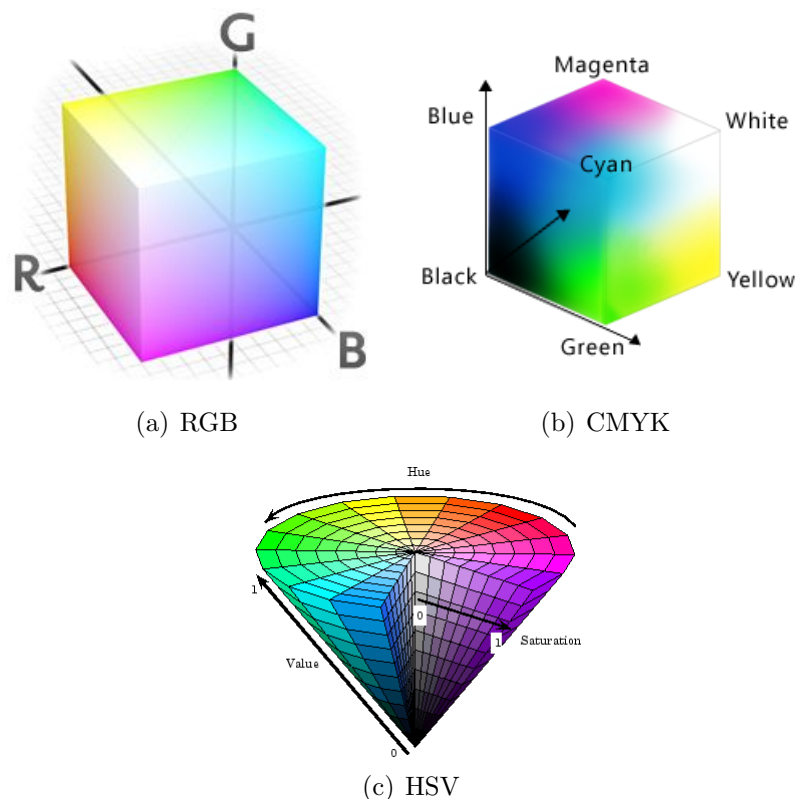


FIGURA 4.1: Modelos o espacios de colores

4.2 Construcción de Grafos Piramidales

En el tratamiento de problemas de visión, se comienza con el análisis del elemento más pequeño de una imagen; el píxel, y se continúa con zonas más grandes (como un conjunto de píxeles contiguos). A este enfoque se le llama procesamiento de información *local* o *distribuida* porque es posible realizar el trabajo asignando segmentos de la imagen a subprocesos independientes, sin embargo, esto no es eficiente para problemas de visión de nivel alto o intermedio (de mayor complejidad), porque estos procesos necesitan información que esté basada en los datos de toda la imagen. A este tipo de información se le llama *global* o *centralizada*. Por lo tanto, se requiere una estructura de datos que permita la transformación de información local (basadas en fragmentos o subimágenes) a información global (basada en toda la imagen) y ser capaz de manejar información tanto distribuida como centralizada. La pirámide es tal estructura y se le conoce como *arquitectura jerárquica*. Ésta permite que procesos que manejan información local puedan hacer uso de información global. Se dice que la pirámide cede posibilidades de paralelización pero obtiene las ventajas del manejo jerárquico de una imagen, es decir a distintas resoluciones [12, 17].

Una pirámide puede poseer la descripción del contenido de una imagen a múltiples niveles de resolución. Se tiene como el nivel base (nivel 0, inferior o el *fondo* de la pirámide) a la imagen de entrada original en alta resolución. La cantidad de información entre niveles sucesivos (superiores) se va reduciendo por un *factor de reducción* $\lambda > 1.0$. Es decir, un factor de reducción $\lambda = 2$ reduce la cantidad de información de un nivel a otro a la mitad. La relación entre una celda (conjunto de píxeles) en un nivel reducido (superior) con un conjunto de celdas en el nivel directamente por debajo se le conoce como *Ventana de reducción*. De esta manera, los procesos locales independientes (y paralelos) propagan la información en todas direcciones dentro de la pirámide: Hacia arriba (a niveles superiores reducidos), hacia abajo (a niveles anteriores a mayor resolución) y lateralmente (en un mismo nivel en la pirámide). El contenido de una celda de baja resolución, es decir, de un nivel superior se computa mediante una función *reducción*, cuyo argumento de

entrada es la descripción de las celdas en la *ventana de reducción*. En ocasiones surge la necesidad de extrapolar u obtener la información de las celdas de una resolución mayor (nivel inferior) a partir de las celdas de una resolución menor (o nivel superior). Esta función es llamada *función de refinamiento o expansión* [12, 17].

La información de los niveles superiores debe estar relacionada con los datos de entrada originales del nivel base de la pirámide, es decir, la descripción de la celda de un nivel k con $k > 0$ debe estar basada en la información de los píxeles correspondientes de la imagen original (nivel 0). A esta relación se le identifica como el *campo receptivo (CR)* de una celda piramidal dada. El campo receptivo $CR(c_i)$ une todas las celdas (píxeles) del nivel base de las cuales la celda c_i es su ancestro [17, 1].

La estructura vertical, las relaciones entre ventanas de reducción y las imágenes en diferentes resoluciones pueden observarse en la figura 4.2.

Basándose en la manera en que las celdas en niveles subsecuentes se unen, se dice que existen dos tipos de pirámides: Regulares e irregulares.

Estos conceptos están fuertemente relacionados con la habilidad de la pirámide para representar patrones de figuras regulares e irregulares que recubren el plano de la imagen. A estos patrones se les conoce como “*teselados*”.

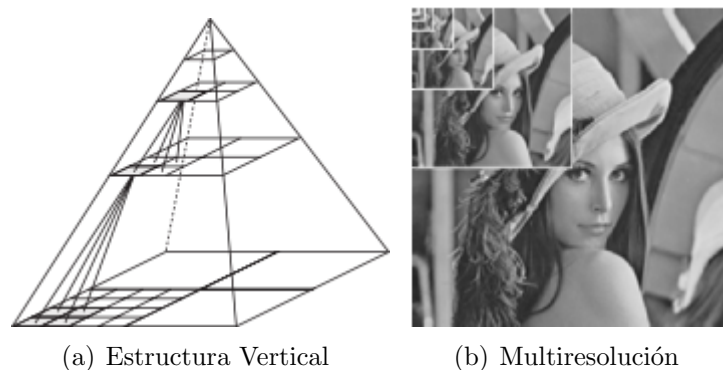


FIGURA 4.2: Pirámide

4.2.1 Pirámides Regulares

Una pirámide regular, como su nombre indica, posee el mismo factor de reducción entre todos sus niveles y, de igual manera, las ventanas de reducción se construyen de la misma forma para todas las celdas. A esto se le llama *factor de reducción constante* y *ventana de reducción constante* que definen completamente la estructura de la pirámide regular. La proporción decreciente de celdas de nivel a nivel es determinada por el factor de reducción. El número de niveles h está limitado por el factor de reducción $\lambda > 1.0$ donde $h \leq \log(\text{tamaño_imagen}) / \log(\lambda)$. La *principal ventaja computacional* de pirámides regulares es debido a su *complejidad logarítmica*.

Usualmente, las pirámides regulares son usadas en planos con rejillas regulares que poseen patrones que recubren toda la imagen, por tanto, la ventana de reducción es normalmente un cuadrado de $n \times n$, esto significa que las celdas $n \times n$ están asociadas a una celda en un nivel mayor directamente por encima. Las pirámides regulares se describen usando la notación $n \times n/\lambda$. De esta manera, por la rígida estructura vertical, la pirámide imagen regular es una estructura ideal para una rápida agrupación y un acceso veloz a los objetos de la imagen a través de la imagen de entrada. Sin embargo, la representación de una pirámide regular de una imagen desplazada, rotada y/o escalada no es única y además no preserva la conectividad. Esto hace que las pirámides imagen regulares no sean adecuadas para un algoritmo de segmentación de propósito general.

4.2.2 Pirámides Irregulares

A diferencia de las pirámides regulares, las pirámides irregulares carecen de un factor de reducción constante que haga que las ventanas de reducción se construyan de igual forma para todas las celdas de un nivel y, por lo tanto, es necesario indicar de forma explícita las relaciones de vecindad entre los píxeles para evitar la aparición de ventanas de reducción traslapadas que produzcan aún más redundancia en los datos de la que, por naturaleza de las imágenes, ya existe. Describir

estas vecindades (tanto verticales como horizontales) se hace usualmente mediante formalismos de grafos. Una de las razones principales para usar pirámides irregulares es la de lograr invarianza con los desplazamientos. Otra justificación es que la irregularidad de la estructura permite las modificaciones necesarias para adaptarse a la naturaleza también irregular de la organización de los sensores de visión [12, 25, 13, 22, 7]. Esto puede verse en las resoluciones disponibles de la mayoría de instrumentos de captura de imágenes, tanto médicos como convencionales, que poseen rejillas de sensores (CCD o CMOS) de cantidad variable ya sea en la proporción entre sensores verticales y horizontales, o en la cantidad total de estos.

Las pirámides deben poseer las siguientes características, sin importar si son regulares o irregulares [12]:

1. Las operaciones sobre los datos de la pirámide deben ser locales. Es decir, el resultado puede ser calculado sin variación independientemente del orden en que sean aplicadas, para permitir la paralelización.
2. Reducción o decimación exponencial del número de celdas en la construcción hacia arriba (*Bottom-up*) de la pirámide.

Tanto la estructura general de la pirámide regular como el proceso de reducción para obtener dicha estructura se determina por la ventana y factor de reducción de la pirámide. Si se elimina esta restricción de regularidad, se tiene que definir un procedimiento para derivar la estructura del nivel reducido G_{k+1} a partir de G_k . Dicho de otra forma, se debe definir un método para contraer las celdas de un nivel de la pirámide.

Para definir este método, primero es necesario tener una representación que permita la manipulación computacional de la imagen. Cada uno de los niveles de la pirámide corresponderá a una de estas representaciones.

Los grafos son una estructura de datos que permite describir de manera ideal una imagen digital, donde cada punto (píxel) puede ser relacionado con un vértice.

Por la flexibilidad que tienen los grafos, estos se vuelven un elemento esencial en la construcción de las pirámides. La ganancia que se obtiene con esta flexibilidad se paga con un acceso a los datos menos eficiente y un incremento en los requerimientos de memoria.

Para poder expresar la conectividad entre puntos de la imagen, la representación de ésta debe ser ampliada con una relación de vecindad. La vecindad de los puntos de muestreo puede ser representada explícitamente mediante un grafo (por sus aristas). El concepto *puntos de muestreo* se refiere a la toma de suficientes puntos discretos en un espacio bidimensional continuo que representen una imagen.

El grafo puede consistir de vértices correspondientes a los píxeles o puntos de muestreo, y de las aristas que conectan a los vértices vecinos. Aunque esta estructura de datos consume más memoria, tiene varias ventajas [12]:

- Los puntos de muestreo no necesitan ser acomodados en una rejilla regular.
- Cada nodo y/o arista puede acompañarse por uno o más atributos que lo describan.
- Las aristas pueden ser determinadas automáticamente.

En las pirámides irregulares, cada nivel representa una partición del conjunto de píxeles en celdas, es decir, subconjuntos *conexos* de píxeles. Usualmente en el nivel base (nivel 0) de una pirámide imagen irregular, una celda representa un solo píxel y la vecindad de cada celda está definida por la cuádruple conectividad de los píxeles, excepto en los píxeles correspondientes a las orillas, puesto que por su posición no tendrían vecinos superiores, inferiores o laterales. Una celda en el nivel $k + 1$ (padre) es una unión de celdas vecinas del nivel k (hijos). Las vecindades del nivel $k + 1$ se derivan de las vecindades del nivel k . Dos celdas c_1 y c_2 son vecinas si existen píxeles $p_1 \in c_1$ y $p_2 \in c_2$ tales que p_1 y p_2 son vecinos.

Un grafo dual es una representación útil que permite la simplificación necesaria para construir cada uno de los niveles en la pirámide imagen irregular y ha sido definido en el capítulo 2.

4.2.3 Grafos Imagen Duales

Una imagen puede transformarse en un grafo de manera que cada píxel se asocie a un vértice y que las vecindades horizontales y verticales entre píxeles se representen mediante aristas. Hay que notar que no se imponen restricciones en cuanto a las dimensiones de la imagen para la creación de la rejilla de muestreo y, por consiguiente, cualquier imagen puede ser transformada a un grafo. Ya que generalmente los grafos con aristas que representan las vecindades diagonales no son planos, estas no se incluyen en la construcción de la pirámide. La intensidad, el color o cualquier otra característica asociada al píxel puede ser considerada como un atributo del vértice (y/o la arista). Al grafo que representa los píxeles de la imagen y sus vecindades, se le llama *grafo primal* o *grafo de vecindades* y se denota por $G = (V, E)$. Los píxeles representan regiones y el grafo G representa un grafo con *caras* como vértices. El grafo dual \bar{G} representa los bordes de estas caras y se llama simplemente *grafo dual*. Formalmente, el par de grafos (G, \bar{G}) , donde $G = (V, E)$ y $\bar{G} = (\bar{V}, \bar{E})$ se llaman grafos imagen duales si ambos grafos (G, \bar{G}) son finitos, planos, conexos y duales uno del otro. Dado que la imagen es finita y conexa, el grafo también lo es. La representación visual de la construcción de los grafos imagen duales, desde una imagen muy sencilla (5×5 píxeles), se muestra en la figura 4.3.

4.3 Contracción de Grafo Dual

Las pirámides irregulares se construyen de forma que un nivel $k + 1$ resulte de la contracción (dual) del nivel anterior k . La contracción del grafo dual consiste de 2 pasos:

- Contracción de una arista primal y remoción de su dual.
- Contracción de una arista dual y remoción de su primal.

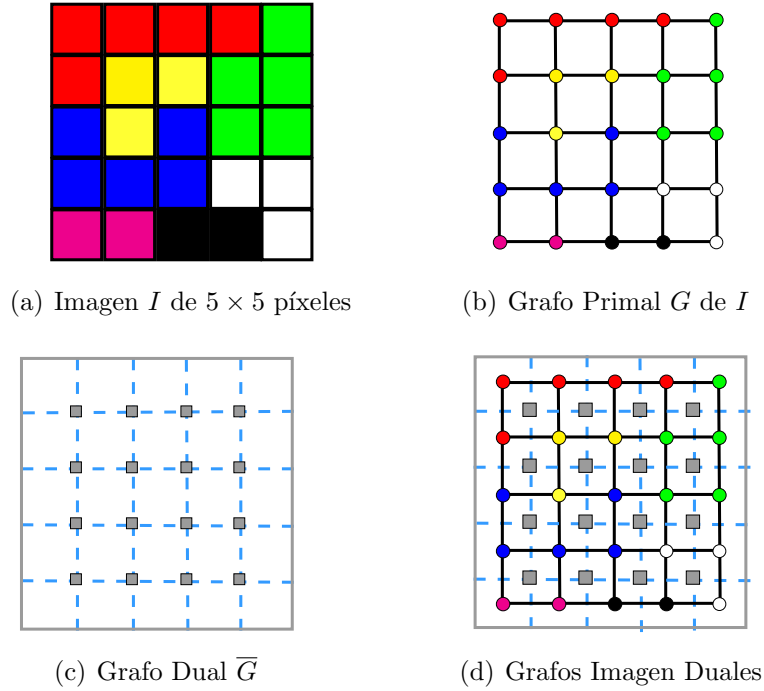


FIGURA 4.3: Construcción de grafos imagen duales

La base de la pirámide consiste del par de grafos imagen duales $(G_0, \overline{G_0})$. Para proceder con la contracción de grafo dual, debe definirse un conjunto de núcleos de contracción (parámetros de decimación).

4.3.1 Núcleos de Contracción

Sea S el conjunto de vértices sobrevivientes de la contracción de un nivel k a $k + 1$, y N el conjunto de aristas primales no sobrevivientes. El conjunto de componentes conexos $CC(s)$ (Cap. 2), con $s \in S$, forman un conjunto de estructuras de árboles enraizados $T(s)$ que, si se contraen, se reducen a la raíz s . La cantidad de estos árboles es $|S|$. La unión de árboles $T(s)$ contiene las aristas primales no sobrevivientes. Cuando se omitan subíndices en la notación $CC(s)$, esta se referirá a los componentes conexos correspondientes a la contracción de un nivel k a $k + 1$. $T(s)$ es un *árbol abarcador* del componente conexo $CC(s)$ o, de forma equivalente, (V, N) es un bosque abarcador del grafo $G = (V, E)$. Para decimar el

grafo $G = (V, E)$, se debe seleccionar el conjunto de vértices sobrevivientes $S \subset V$ y el conjunto de aristas primales no sobrevivientes $N \subset E$, de tal manera que [12]:

- (V, N) sea un bosque abarcador de G .
- Los vértices sobrevivientes $s \in S \subset V$ sean las raíces de los árboles en (V, N) .

El conjunto de núcleos de contracción $\langle S_k, N_{k,k+1} \rangle$ consiste de un subconjunto de vértices sobrevivientes $S_k = V_{k+1} \subset V_k$ y un subconjunto de aristas primales no sobrevivientes $N_{k,k+1} \subset E_k$ (donde el subíndice $k,k+1$ se refiere a *la contracción del nivel k al $k+1$*). Los vértices sobrevivientes son aquellos que no se modificarán por la contracción, es decir, después de la contracción estos vértices pasarán a ser el conjunto V_{k+1} del grafo G_{k+1} . Cada vértice no sobreviviente $v \in V_k \setminus S_k$ debe emparejarse a un vértice sobreviviente de forma única, usando aristas primales no sobrevivientes [12].

Recopilando, un conjunto de árboles enraizados disjuntos con longitud de ruta dos, *pasando por la raíz*, se llama *conjunto de núcleos de contracción*. Análogamente, los árboles $T(v)$ del bosque (V, N) con raíces $v \in V$ son *núcleos de contracción*.

La contracción de una arista primal no sobreviviente consiste en la identificación de sus extremos y la remoción de la arista primal contraída y su dual correspondiente.

Es importante recalcar que $\langle S_k, N_{k,k+1} \rangle$ determina la estructura de la pirámide irregular. La relación entre los dos pares de grafos duales (G_k, \overline{G}_k) y $(G_{k+1}, \overline{G}_{k+1})$ se establece por la contracción de grafo dual como:

$$(G_{k+1}, \overline{G}_{k+1}) = C [(G_k, \overline{G}_k), \langle S_k, N_{k,k+1} \rangle] \quad (4.1)$$

La contracción de las aristas duales y eliminación de sus primales tiene el papel de limpiar el grafo primal, simplificando la mayoría de las aristas múltiples y bucles, pero no aquellas que rodean partes sobrevivientes del grafo, ya que son necesarias para preservar la estructura. La contracción grafo dual reduce el número

de vértices y aristas de un par de grafos duales mientras preserva las relaciones topológicas entre las partes sobrevivientes [12].

Después de aplicar el algoritmo de contracción grafo dual al par de grafos duales, uno tiene que establecer una ruta que conecte dos vértices sobrevivientes en el nuevo grafo resultante.

Sea $G = (V, E)$ un grafo con parámetros de decimación (S, N) . Una ruta en G se llama una ruta de conexión entre dos vértices sobrevivientes $s, s' \in S$ si consiste de tres subconjuntos de aristas:

- * La primera parte es una rama, posiblemente vacía del núcleo de contracción $T(s)$.
- * La segunda parte es una arista $e \in E \setminus N$ que conecte a los dos núcleos de contracción $T(s)$ y $T(s')$.
- * La tercera parte es una rama, posiblemente vacía del núcleo de contracción $T(s')$.

La ruta de conexión se denota por $CP(s, s')$. La arista e (la parte media) se llama el *punte* de la ruta de conexión $CP(s, s')$. Cada arista $e' = (v, v') \in E_{k+1}$ tiene una ruta de conexión correspondiente $CP_k(s, s')$, donde $s, s' \in S \subset V_k$ son vértices sobrevivientes en el grafo $G_k = (V_k, E_k)$. Esto significa que dos vértices sobrevivientes s y s' , con $s \neq s'$ que puedan ser conectados por una ruta $CP_k(s, s')$ en G_k están conectados por una sola arista en E_{k+1} . Si el grafo G_k es conexo, después de la contracción grafo dual la conectividad se preserva en el grafo G_{k+1} [12].

El procedimiento detallado de la contracción se muestra de forma esquematizada en la figura 4.4.

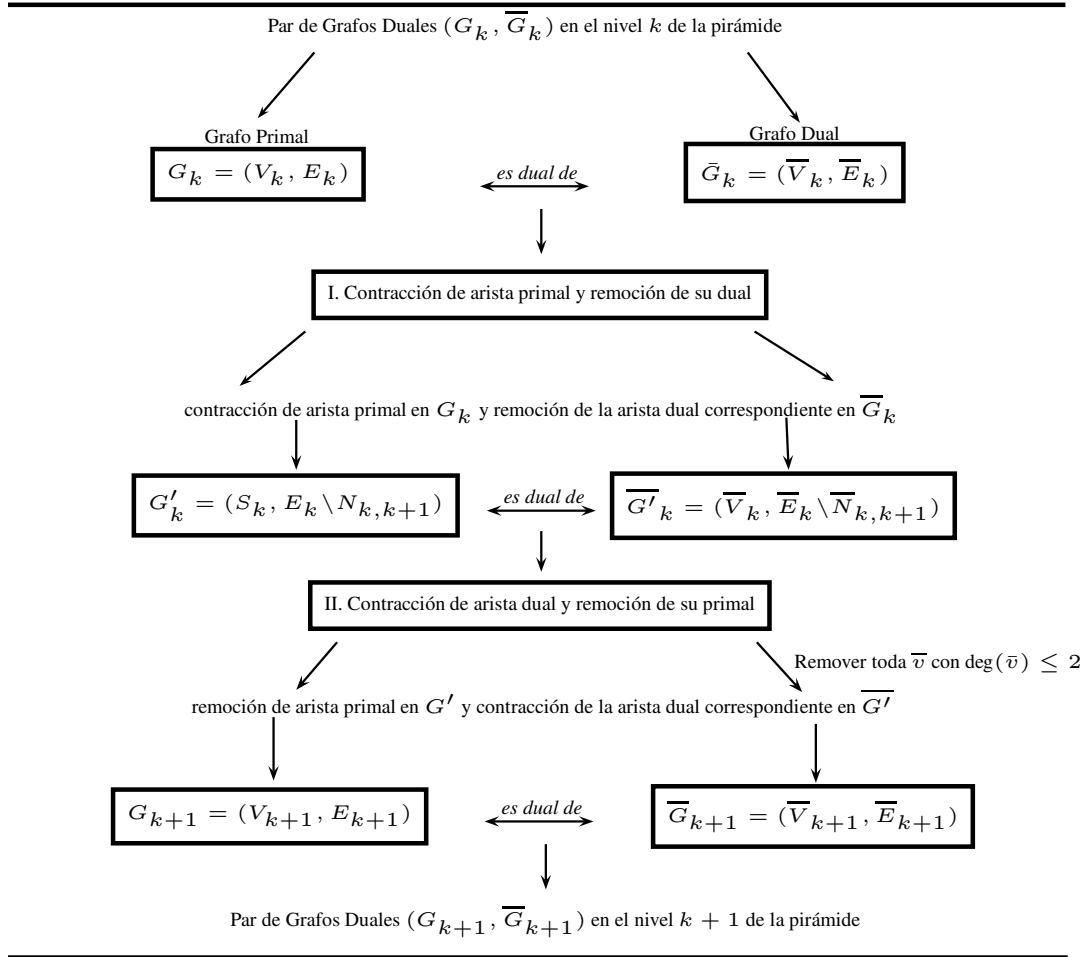


FIGURA 4.4: Procedimiento de contracción de grafo dual

4.3.2 Núcleos de Contracción Equivalentes

Es posible calcular cualquier nivel de una pirámide directamente desde el nivel base, combinando dos o más contracciones grafo dual. Inicialmente, puede construirse un núcleo de contracción con base en dos núcleos de contracción de niveles contiguos. A esto se le llama *Núcleo de Contracción Equivalente*.

La estructura de G_{k+1} está determinada por G_k y los parámetros de decimación $\langle S_k, N_{k,k+1} \rangle$. Los vértices sobrevivientes de G_k a G_{k+1} son $S_{k+1} = V_{k+1}$. Las aristas de los núcleos de contracción deseados deben estar formados por las aristas $N_{k,k+1} \subset E_k$. Una arista $e_{k+1} = (v_{k+1}, v'_{k+1}) \in N_{k+1,k+2}$ corresponde a una ruta de conexión $CP_k(v_{k+1}, v'_{k+1})$ en G_k . Por definición, $CP_k(v_{k+1}, v'_{k+1})$ consiste de una rama de $T_k(v_{k+1})$, una rama de $T_k(v'_{k+1})$ y una arista sobreviviente $e_k \in E_k$ conectando los dos núcleos de contracción $T_k(v_{k+1})$ y $T_k(v'_{k+1})$ [12].

La función puente $bridge : E_{k+1} \mapsto E_k$ asigna a cada arista $e_{k+1} = (v_{k+1}, w_{k+1}) \in E_{k+1}$ uno de los puentes $e_k \in E_k$ de las rutas de conexión $CP_k(v_{k+1}, w_{k+1})$: $bridge(e_{k+1}) = e_k$.

Conectar dos estructuras de árboles disjuntas por una sola arista resulta en una nueva estructura de árbol. Con la función recién definida, puede construirse $N_{k,k+2}$ conectando todos los núcleos de contracción T_k mediante los puentes, formalmente como:

$$N_{k,k+2} = N_{k,k+1} \cup \bigcup_{e_{k+1} \in N_{k+1,k+2}} bridge(e_{k+1})$$

Este proceso puede repetirse sucesivamente para cualquier par de niveles k y k' , tal que $k < k'$. Así, es posible computar cualquier nivel de la pirámide directamente desde la base.

4.3.3 Grafo Dual Piramidal

Un grafo piramidal es una pirámide donde cada nivel es un grafo $G(V, E)$ que consiste de vértices V y aristas E que relacionan dos vértices. Para representar correctamente la estructura del grafo en el plano de la imagen, se almacena adicionalmente el grafo dual $\overline{G}(\overline{V}, \overline{E})$ por cada nivel. Los niveles son representados como pares (G_k, \overline{G}_k) de grafos duales planos G_k y \overline{G}_k .

La secuencia (G_k, \overline{G}_k) , donde $0 \leq k \leq h$, se llama *Grafo Dual Piramidal*, donde 0 es el índice del nivel base y h es el índice del nivel de la cima, también llamado la altura de la pirámide. Adicionalmente, a los grafos se les asignan atributos de forma $G(V, E, attr_v, attr_e)$ donde $attr_v : V \mapsto \mathbb{R}^+$ y $attr_e : E \mapsto \mathbb{R}^+$. Esto significa que el contenido de un grafo se almacena en atributos adjuntos a cada vértice y arista[12]. En general, un grafo piramidal puede construirse de forma *bottom-up* (del nivel base al nivel más alto), como lo muestra el algoritmo 4.1.

Algoritmo 4.1 Construcción de una pirámide de grafo dual

Entrada: Grafos $(G_0, \overline{G_0})$

- 1: **Mientras** sea posible más abstracción **Hacer**
- 2: Determinar núcleos de contracción $N_{k,k+1}$.
- 3: Realizar contracción y simplificación de los grafos duales.
- 4: Aplicar funciones de reducción para calcular el contenido del atributo del nuevo nivel reducido.
- 5: $k \leftarrow k + 1$
- 6: **Fin Mientras**

Salida: Grafo Pirámide $(G_k, \overline{G_k})$, $0 \leq k \leq h$

4.4 Jerarquía de Particiones

En las pirámides irregulares, cada nivel representa una *partición* del conjunto de píxeles en celdas (subconjuntos conexos de píxeles). A la estructura formada por niveles $k+1$ reducidos con base en un nivel k , relacionados para permitir el acceso a los datos ya sea de la base hacia la cúspide o de la cúspide hacia la base, se le llama *jerarquía*. Esta jerarquía de particiones (subconjuntos de píxeles recursivamente construidos de los subconjuntos del nivel anterior) es una herramienta importante para el particionamiento de imágenes ya que el criterio de homogeneidad está naturalmente representado por las distintas regiones con fuertes semejanzas internas en las que se busca dividir la imagen.

4.4.1 Árbol de Expansión Mínima (MST)

La segmentación tiene como fin encontrar agrupaciones naturales de píxeles (regiones), para esto es necesario encontrar una forma eficaz de particionar los píxeles en regiones y de medir de alguna manera la semejanza entre ellos. Existen varios métodos estocásticos (relativos al azar) que eligen un conjunto de píxeles, en los que posteriormente se realiza un análisis para particionar a los mismos en regiones. Entre estos métodos se encuentran los algoritmos MIS y MIES[18]. Estos algoritmos buscan un conjunto independiente maximal de los vértices de un nivel de la pirámide para luego seleccionar los elementos que desaparecerán y que sobrevivirán en el siguiente nivel. Otro enfoque para seleccionar los vértices en cada iteración es

mediante uno de los problemas más estudiados en las ciencias de la computación: Los árboles recubridores (abarcadores o de expansión) mínimos (MST). Es posible combinar los métodos estocásticos previamente mencionados y construir árboles recubridores mínimos (MST) a partir de los nodos resultantes.

El problema se define de la siguiente forma. Sea $G = (V, E)$ un grafo conexo no dirigido con un conjunto finito de vértices V y un conjunto finito de aristas E . Cada arista $e \in E$ es incidente con un par de vértices $v_i, v_j \in V$ tales que $v_i \neq v_j$. Además, cada arista $e \in E$ está asociada a un único peso $w(e) = w(v_i, v_j)$ del universo ordenado (se asume que todos los pesos son distintos pero los empates pueden romperse arbitrariamente). Así, se busca construir un árbol que contenga todos los vértices de G y cuyo peso total de sus aristas (o ramas) sea el mínimo posible. Nótese que los pesos de las aristas paralelas también se asumen diferentes, esto es $e_1 = (v_i, v_j)$, $e_2 = (v_i, v_j)$, $e_1 \neq e_2$.

Algoritmo 4.2 Algoritmo de Borůvka

Entrada: Grafo $G(V, E)$

- 1: $MST \leftarrow$ Lista de aristas vacía.
- 2: Cada uno de los vértices $v \in V$ hacen una lista de árboles L .
- 3: **Mientras** haya más de un árbol en L **Hacer**
- 4: Para cada árbol $T \in L$, encontrar la arista e con el mínimo peso que conecta T a $G \setminus T$ y agregar arista e a MST .
- 5: Usando la arista e , unir pares de árboles en L .
- 6: Limpiar el grafo de bucles de ser necesario.
- 7: **Fin Mientras**

Salida: Árbol recubridor mínimo (Subgrafo inducido de aristas en MST)

Aunque hay varios algoritmos para la construcción del árbol recubridor mínimo, se elige el algoritmo de Borůvka debido a que deja el camino abierto para aplicar paralelización. Nótese que mientras más grande se vuelva el árbol T procesado dentro de la iteración del paso 4 en el algoritmo 4.2, el proceso de encontrar la arista con el mínimo peso se vuelve más tardado.

El algoritmo general de la contracción del grafo dual (4.1) crea la relación vertical entre niveles de la pirámide al crear *supervértices*, es decir, crea la relación padre-hijo entre los vértices de un nivel $k + 1$ (padre) y del nivel k (hijo), mientras que el algoritmo de Borůvka crea la relación entre vértices del mismo nivel de

la pirámide (relación horizontal). El algoritmo 4.2 sólo describe la construcción del MST, mientras que el algoritmo 4.3 incluye los pasos para contraer aristas no sobrevivientes, eliminar aristas paralelas y bucles. El proceso de medición de características similares entre píxeles que encuentra los bordes de cada región será discutido en la siguiente sección.

Algoritmo 4.3 Algoritmo de Borůvka con DGC (*Dual Graph Contraction*)

Entrada: Grafo con atributos $G_0(V, E)$

1: $MST \leftarrow$ Lista de aristas vacía.

2: $k \leftarrow 0$.

3: **Repetir**

4: Para cada vértice $v \in G_k$, encontrar la arista $e \in G_k$ con el mínimo peso incidente en el vértice v y marcar las aristas e que serán contraídas.

5: 4. Determinar CC_i^k como los componentes conexos de las aristas e marcadas.

6: 5. Contraer los componentes conexos CC_i^k en un solo vértice y eliminar las aristas paralelas (excepto aquella con el mínimo peso) y los bucles. Crear el grafo $G_{k+1} = C[G_k, CC_i^k]$.

7: $k \leftarrow k + 1$

8: **Hasta que** todos los componentes conexos de G estén contraídos en un solo vértice.

Salida: Un grafo pirámide con una cúspide.

4.4.2 Construcción de la Jerarquía de Particiones

Los criterios de semejanza dentro de las regiones de la imagen o *criterio de homogeneidad* surgen naturalmente dentro de la jerarquía de particiones. En [24] se propone un predicado booleano P para representar un criterio de homogeneidad.

Sea V un conjunto, un predicado booleano P sobre sus partes $\Phi(V)$ verifica la propiedad de consistencia: $\forall(x, y) \in \Phi(V) : x \subset y \Rightarrow (P(y) \Rightarrow P(x))$. En el análisis de imágenes, esto equivale a decir que las subregiones de una región homogénea también son homogéneas. De esta manera, si Pyr es una jerarquía y P es un criterio de homogeneidad consistente en V , entonces el conjunto de elementos maximales de Pyr que satisfacen P definen una única partición de V . Como resultado, el uso combinado de la jerarquías y los criterios de homogeneidad permiten definir una partición de forma natural.

El objetivo es encontrar particiones formadas de componentes conexas $P_k = \{CC(u_1), \dots, CC(u_n)\}$ en el k -ésimo nivel de la pirámide tal que estos elementos satisfagan ciertas propiedades. Es importante especificar que P es una partición de $V \in G$. Formalmente se define como $\forall i \neq j, CC_i \cap CC_j = \emptyset$ y que $\bigcup_{i=1 \dots n} CC_i = V$. Para verificar que los elementos de los componentes conexas cumplan propiedades específicas, se usa una función B de comparación a pares definida como:

$$B(CC(u_i), CC(u_j)) = \begin{cases} \text{Verdadero} & \text{Si existe frontera entre } CC(u_i) \text{ y } CC(u_j) \\ \text{Falso} & \text{De otro modo.} \end{cases} \quad (4.2)$$

La definición específica de la función B depende de la aplicación. En este proyecto, la función booleana mide la diferencia sobre la frontera de dos componentes en relación a la medida de las diferencias internas de los componentes. Esta definición trata de encapsular la noción de contraste: una zona contrastada es una región que contiene dos componentes conexas cuyas diferencias internas (contraste interno) son menores que las diferencias dentro de su contexto (contraste externo).

Para aclarar la notación usada en los algoritmos, se tiene que cada vértice $u \in G_k$ es un representante de un componente conexo $CC(u)$ de la partición P_k . El núcleo de contracción equivalente $N_{0,k}(u)$ de un vértice $u \in G_k$, es un conjunto de aristas en el *nivel base* que se contraen. Es decir, al aplicar $N_{0,k}(u)$ en el nivel 0 (base), se contrae el subgrafo $G' \subseteq G$ y se convierte en el vértice u . El *contraste interno* de $CC(u) \in P_k$ es la *disimilaridad más grande* dentro del componente $CC(u)$, es decir, el atributo de la arista más grande de $N_{0,k}(u)$ del vértice $u \in G_k$. Formalmente se denota:

$$Int(CC(u)) = \max\{attr_e(e), e \in N_{0,k}(u)\}. \quad (4.3)$$

Sean $u_i, u_j \in V_k, u_i \neq u_j$ los vértices extremos de una arista $e \in E_k$. El contraste externo entre dos componentes $CC(u_i), CC(u_j) \in P_k$ es la disimilaridad más pequeña entre el componente $CC(u_i)$ y $CC(u_j)$. Es decir, la arista con el atributo más pequeño que conecte $N_{0,k}(u_i)$ y $N_{0,k}(u_j)$ de los vértices $u_i, u_j \in G_k$.

Formalmente, se denota:

$$Ext(CC(u_i), CC(u_j)) = \min \{attr_e(e), e = (u_i, u_j) : u_i \in N_{0,k}(u_i) \wedge u_j \in N_{0,k}(u_j)\} \quad (4.4)$$

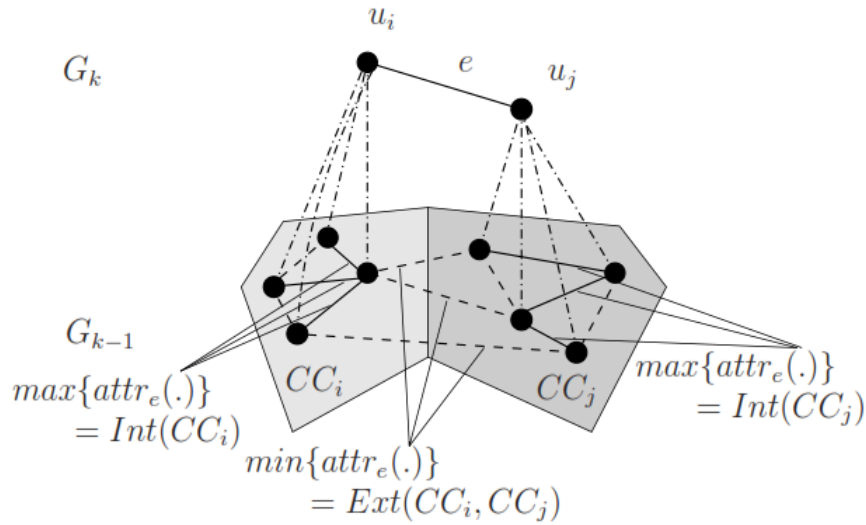


FIGURA 4.5: Contraste interno y externo

Con los contrastes internos y externos definidos, ahora es posible definir formalmente la función booleana de comparación entre componentes:

$$B(CC(u_i), CC(u_j)) = \begin{cases} V & \text{Si } Ext(CC(u_i), CC(u_j)) > PInt(CC(u_i), CC(u_j)) \\ F & \text{De otro modo.} \end{cases} \quad (4.5)$$

Donde $PInt(CC(u_i), CC(u_j))$ es la diferencia mínima del contraste interno entre dos componentes:

$$PInt(CC(u_i), CC(u_j)) = \min(Int(CC_i) + \tau(CC_i), Int(CC_j) + \tau(CC_j)) \quad (4.6)$$

Para simplificar la notación, $CC_i = CC(u_i)$ y $CC_j = CC(u_j)$. Gráficamente, se pueden representar las funciones de contraste externo e interno como se tiene en la figura 4.5.

Para que el borde exista, la diferencia del contraste externo debe ser mayor que las diferencias del contraste interno. La razón para usar una función de umbralización

$\tau(CC)$ en la ecuación 4.6 es que, para componentes pequeños CC , $Int(CC)$ no es un buen estimado de las características locales de los datos. En un caso extremo, cuando $|CC| = 1$, $Int(CC) = 0$. Cualquier función no negativa de un componente individual CC , puede ser usado para $\tau(CC)$. Uno puede definir τ para que dependa del tamaño de CC :

$$\tau(CC) = \frac{\alpha}{|CC|} \quad (4.7)$$

Donde $|CC|$ denota el tamaño del componente CC y α es una constante.

Algoritmo 4.4 Jerarquía de particiones

Entrada: Grafo con atributos $G_0(V, E)$

- 1: $k \leftarrow 0$.
 - 2: **Repetir**
 - 3: **Para todo** vértice $u \in G_k$ **Hacer**
 - 4: $E_{min}(u) \leftarrow argmin\{attr_e(e) | e = (u, v) \in E_k \text{ o } e = (v, u) \in E_k\}$.
 - 5: **Fin Para**
 - 6: **Para todo** $e = (u_i, u_j) \in E_{min}$ con
 $Ext(CC(u_i), CC(u_j)) \leq PInt(CC(u_i), CC(u_j))$ **Hacer**
 - 7: Incluir e en las aristas de contracción $N_{k,k+1}$.
 - 8: **Fin Para**
 - 9: Contraer grafo G_k con los núcleos de contracción $N_{k,k+1} : G_{k+1} = C[G_k, N_{k,k+1}]$.
 - 10: **Para todo** $e_{k+1} \in G_{k+1}$ **Hacer**
 - 11: $attr_e(e_{k+1}) \leftarrow min\{attr_e(e_k) | e_{k+1} = C[G_k, N_{k,k+1}]\}$.
 - 12: **Fin Para**
 - 13: $k \leftarrow k + 1$
 - 14: **Hasta que** $G_k = G_{k-1}$
- Salida:** Un grafo pirámide con regiones de adyacencia (RAG Region Adjacency Graph).
-

Finalmente, para construir la jerarquía de particiones se utiliza el algoritmo 4.4. En la línea 4 de 4.4, se tiene la función *argmin* aplicada sobre un conjunto de atributos de aristas, que devuelve la arista de menor peso de un vértice u y se define formalmente como $argmin\{attr_e(e) | e = (u, v) \in E_k \text{ o } e = (v, u) \in E_k\} = e^* \in E_k$ donde $attr_e(e^*) \leq attr_e(e), \forall e \in E_k$ tal que $e = (u, v)$ o $e = (v, u)$. Cada vértice $u_i \in G_k$ define a una *región conexa* $CC(u_i)$ en el nivel base de la pirámide y dado que este algoritmo está basado en el algoritmo de Borůvka (4.3), se construye un MST de cada región. Formalmente, $N_{0,k}(u_i) = MST(u_i)$. La idea es recolectar las aristas e con los atributos más pequeños que podrían ser parte del MST y después

verificar si el peso del atributo $attr_e(e)$ es menor que el contraste interno de ambos componentes (MST de los vértices extremos de e). Si se cumplen estas condiciones, entonces estos dos componentes se mezclan. Todas las aristas a contraerse forman el núcleo de contracción $N_{k,k+1}$ usado para construir el grafo $G_{k+1} = C[G_k, N_{k,k+1}]$. En general, $N_{k,k+1}$ es un bosque.

Los atributos de las aristas $e_{k+1} \in G_{k+1}$ se actualizan con el mínimo atributo de las aristas $e_k \in G_k$ que se contrajeron en e_{k+1} . La salida del algoritmo es una pirámide donde cada nivel representa un grafo de regiones de adyacencia (o RAG de Region Adjacency Graph, es decir, particiones). Cada vértice de estos RAG es el representante de un MST de una región en la imagen. Se dice que el algoritmo es voraz [17, 25] ya que junta un vértice sólo con el vecino más cercano cuyo peso de la arista incidente es mínimo y los mezcla si la comparación por pares es falsa.

En [25] se demuestra que el algoritmo 4.4 construye un árbol recubridor mínimo (MST), es decir, está basado en el algoritmo de Borůvka (4.3) si se omite la comparación por pares como criterio de contracción.

En la figura 4.6(a) se puede observar, de forma muy simplificada, el proceso de construcción de la pirámide, que comienza con la creación del grafo imagen primal y posteriormente la contracción primal y dual. Aunque la contracción dual no se muestra explícitamente, se muestran los resultados de ésta en el grafo primal. En la figura 4.6(b), se muestra el procedimiento para obtener la representación en imagen de alguno de los niveles de la pirámide que corresponde a una segmentación.

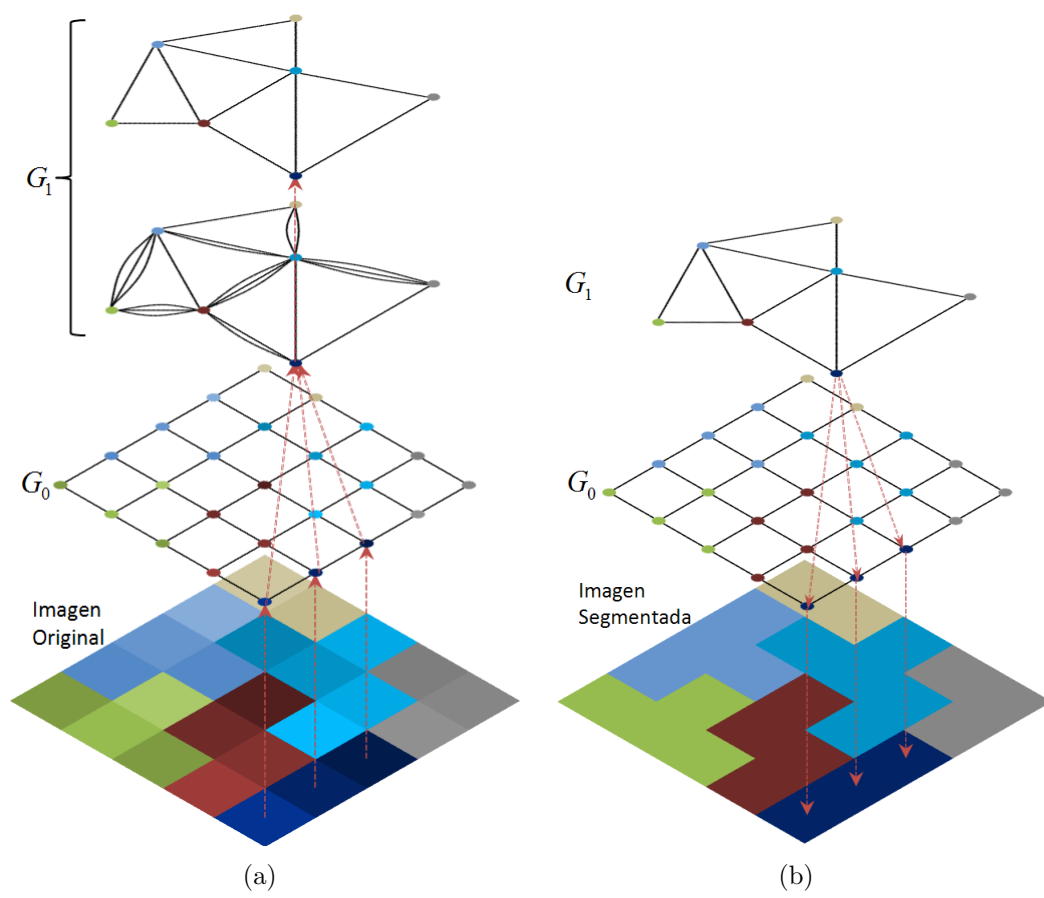


FIGURA 4.6: Construcción de pirámide y obtención de segmentación.

Capítulo 5

Implementación y resultados

5.1 Implementación

La implementación del algoritmo para la construcción de jerarquías, que comprende la contracción grafo dual y la construcción de un MST por cada componente conexas, mencionado en el capítulo 4 se hizo en el lenguaje C# bajo el sistema operativo Windows 7 y también funciona en Windows XP y Vista.

Las pruebas de la segmentación fueron realizadas en imágenes a color y en escala de grises. Inicialmente, cada píxel corresponde a una región homogénea en la base de la pirámide, es decir, cada vértice en la base de la pirámide representa una región homogénea cuyo atributo es el valor de su píxel correspondiente. Los atributos de las aristas difieren con base en el espacio de color utilizado. Si se elige el espacio RGB, el atributo de cada arista se define como la norma euclidiana de la diferencia del atributo de sus vértices extremos. Formalmente es:

$$attr_e(u_i, u_j) = \|F(u_i) - F(u_j)\| \quad (5.1)$$

Donde $F(u_i) = (r_i, g_i, b_i), \forall r, g, b \in \mathbb{N}[0, 255]$. La norma euclidiana para un vector de n dimensiones se define como [10]:

$$\|x\| = \sqrt{\sum_{i=1}^n |x_i|^2} \quad (5.2)$$

Si se elige el espacio HSV, el atributo de cada arista se define como [19]:

$$e \frac{\|F(u_i) - F(u_j)\|}{\sigma_1} \quad (5.3)$$

Donde $F(u_i) = (v_i, v_i \cdot s_i \cdot \sin(h_i), v_i \cdot s_i \cdot \cos(h_i))$ [12, 9] con $0 \leq h \leq 2\pi$, $0 \leq s \leq 1$, $0 \leq v \leq 1$, $\forall h, s, v \in \mathbb{R}$. Los valores h, s, v se obtienen de la conversión de los valores r, g, b correspondientes usando los algoritmos propuestos en [16, 11].

La función $\tau(CC)$ (ecuación 4.7), dependiente del tamaño del componente, posee un parámetro α suministrado por el usuario que funge como constante durante todo el algoritmo de segmentación y sirve para cambiar la influencia que tiene un componente en proporción a su tamaño. Es decir, mientras más grande sea α , el resultado de la función es mayor para las componentes de menor tamaño y provoca que - al comparar componentes diferentes - el de mayor tamaño absorba al más pequeño. Es importante recalcar que, debido al cambio de rangos en los atributos de las aristas si se utiliza uno u otro espacio de colores, la influencia de la función $\tau(CC)$ en la comparación de componentes cambia drásticamente. Con un parámetro α bien ajustado, la función $\tau(CC)$ permite trabajar aún con ruido o zonas de tamaño reducido que son muy contrastantes con su contorno pero que, visiblemente, no son lo suficientemente importantes para ser considerados como regiones. El contraste máximo interior de un componente, en niveles iniciales de la pirámide (donde el orden de sus componentes es relativamente pequeño), es directamente proporcional al resultado de la función $\tau(CC)$. Por lo tanto, se incrementan las probabilidades de que un componente pequeño sea contraído con uno de mayor tamaño, aunque visiblemente se contrasten.

Tanto la decisión de una función $\tau(CC)$ como la de un espacio de colores son problemas, por sí mismos, difíciles. Elegir una función τ que influya de manera deseable a los valores numéricos que suministra el espacio de colores requiere de un gran cantidad de experimentación.

Para mejorar la calidad de la segmentación y evitar la influencia de vértices con colores muy diferentes en un componente, se aplicó una técnica de eliminación de ruido simple en el que se eliminan los colores de menor y mayor intensidad de un grupo ordenado de nueve píxeles (en una máscara de 3×3 alrededor de un píxel central).

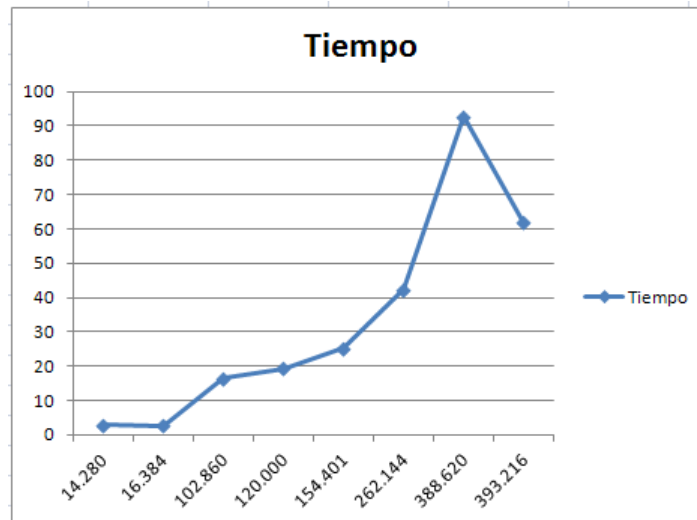
5.2 Resultados

5.2.1 Mediciones

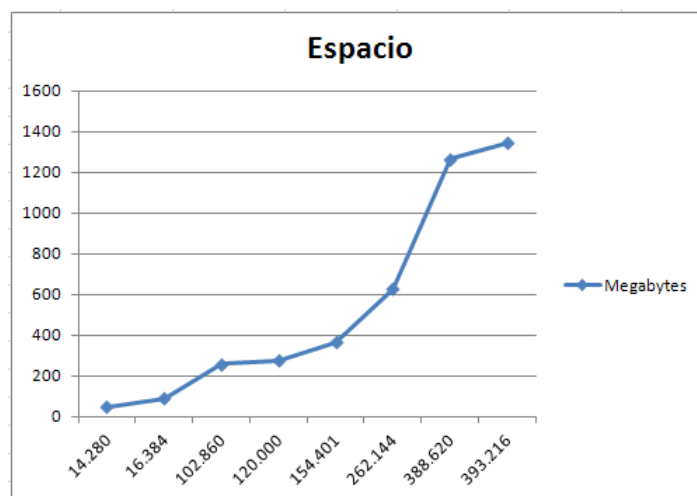
Se realizó un número considerable de pruebas con imágenes de diferentes dimensiones. El menor número de píxeles con el que se trabajó fue de 14,280 y corresponde a una imagen con dimensiones en píxeles de 170×84 . Por otro lado, la imagen segmentada más grande posee dimensiones en píxeles de 768×512 , es decir, 393,216 píxeles. El tiempo y espacio de procesamiento que se requirió para segmentar las imágenes que se muestran en esta sección, se detalla en la tabla 5.1 y la conducta de estos valores se muestra en las gráficas de la figura 5.1.

Dimensión	Píxeles	Tiempo	Memoria
170×84	14,280	2.9 s.	47.1 MB
128×128	16,384	2.6 s.	90.4 MB
370×278	102,860	16.5 s.	259 MB
400×300	120,000	19.4 s.	276.5 MB
481×321	154,401	25.3 s.	364.7 MB
512×512	262,144	42.3 s.	626.1 MB
765×508	388,620	92.6 s.	1,265.5 MB
768×512	393,216	61.8 s.	1,344.5 MB

TABLA 5.1: Mediciones de segmentación.



(a)



(b)

FIGURA 5.1: Curvas que describen la conducta de la memoria requerida y el tiempo de cómputo de la técnica propuesta.

5.2.2 Segmentación de propósito general

Las segmentaciones, basadas en pirámide irregulares, que se presentan a continuación fueron aplicadas en imágenes que se encuentran en bases de datos especializadas para el procesamiento digital y en la comparación de resultados (*benchmarking*) de algoritmos de segmentación y detectores de bordes [15, 8, 5]. Esta base de datos contiene gran cantidad de imágenes que han sido segmentadas por humanos y por técnicas propuestas por otros autores. Los resultados de este

trabajo de tesis son comparados específicamente con la técnica de segmentación inducida por invarianza de escala, debido a que comparten la característica con la técnica de este trabajo, de entregar regiones cerradas como resultado. A pesar de que la segmentación mediante grafos piramidales es de propósito general, es decir, puede aplicarse a cualquier tipo de imagen, hay que hacer notar que no es efectiva en todos los casos. En cada uno de los resultados incluidos en esta sección, se presenta la imagen original, la segmentación en con norma en RGB, la segmentación con norma HSV y un resaltado de bordes correspondiente para cada segmentación. El parámetro de la función τ para ambos espacios de colores es de 300 y el parámetro σ de escalamiento en el espacio de colores HSV es de 0.01. La segmentación corresponde a un solo nivel de la pirámide, el cual se encuentra indicado entre paréntesis en su figura correspondiente.

Puede observarse que la segmentación RGB resultante de las imágenes en la figura 5.2 delimita completamente la región correspondiente del objeto completo, mientras que la segmentación HSV identificó regiones diferentes que componen al mismo. En este caso se diferenció la superficie externa de la interna del tazón. Sin embargo, en ambos casos, la semejanza numérica entre el color del tazón (café) y el color del fondo (negro) provocó que el orificio correspondiente al asa del tazón se haya perdido. En el caso de la figura 5.3, la región correspondiente a la taza contiene la región de la figura impresa en ambas segmentaciones. Se debe recalcar que en la segmentación HSV, la diferencia de tonos en la manzana provocó que parte de ésta se mezclara con la taza, no obstante el área correspondiente a toda la taza fue unificada en una sola región, mientras que en la segmentación RGB no fue posible.

La figura 5.4 muestra la capacidad de la técnica para encontrar regiones en imágenes donde los colores que la componen tienen un cambio gradual, aunque una misma región tenga colores cuya diferencia sea demasiada. Este caso puede verse en la región izquierda de la figura 5.4(e) o 5.4(f): Una sola región está compuesta de colores extremos (blanco y negro). También se muestra el proceso de construcción de la pirámide (5.4(a), 5.4(b), 5.4(d), 5.4(e)), en donde los niveles anteriores a la segmentación final agrupan los colores más parecidos.

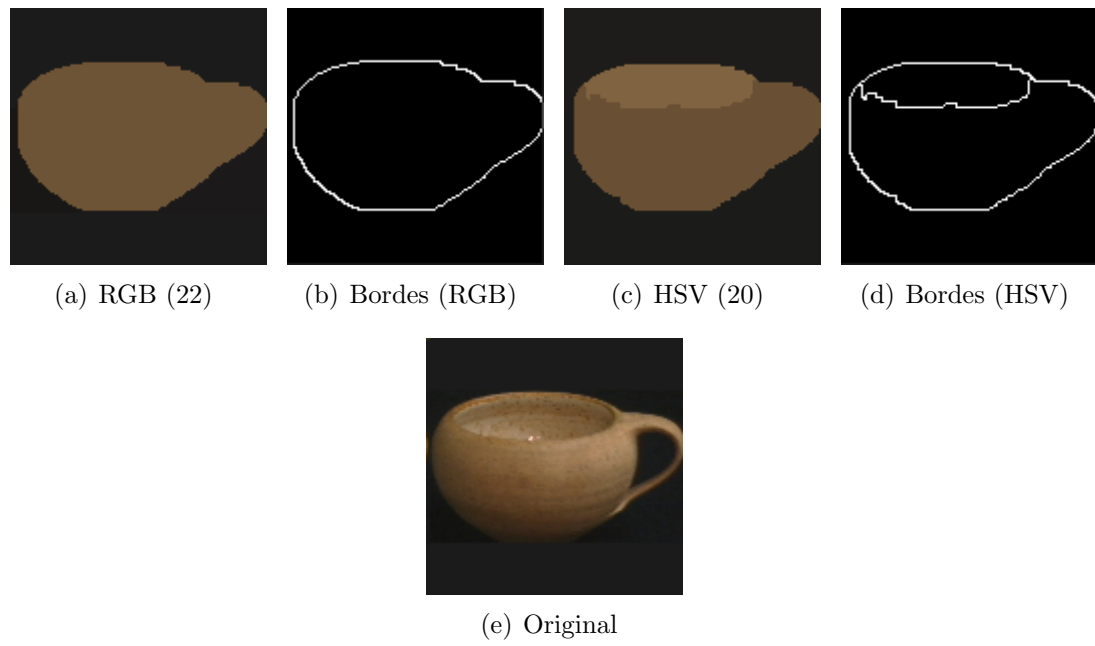


FIGURA 5.2: Segmentación en “Object11”

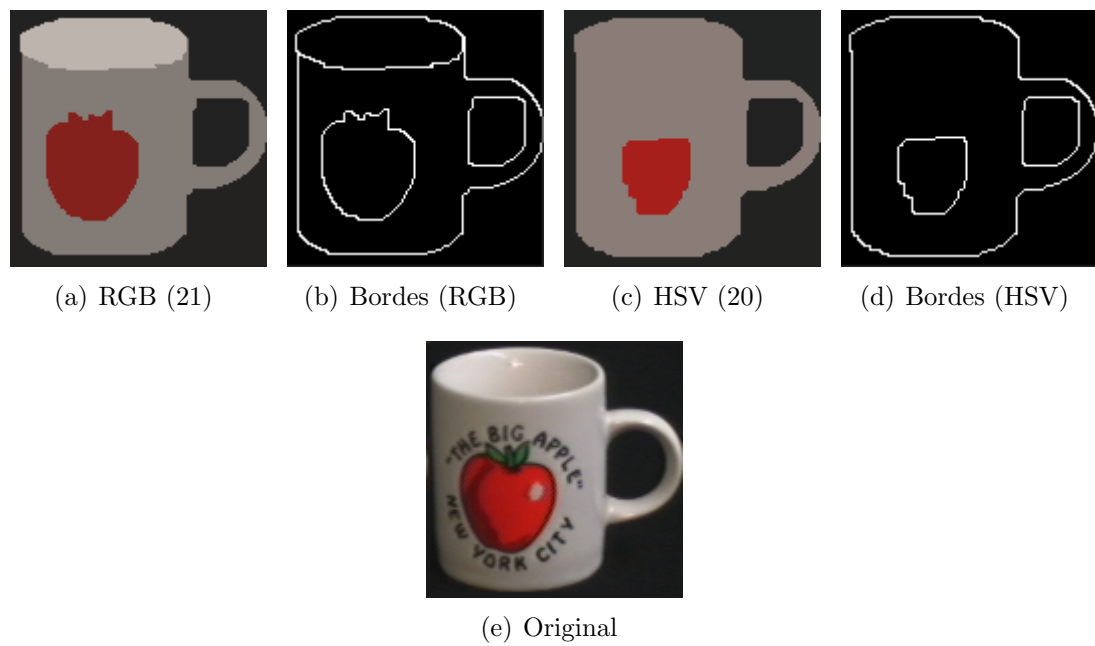
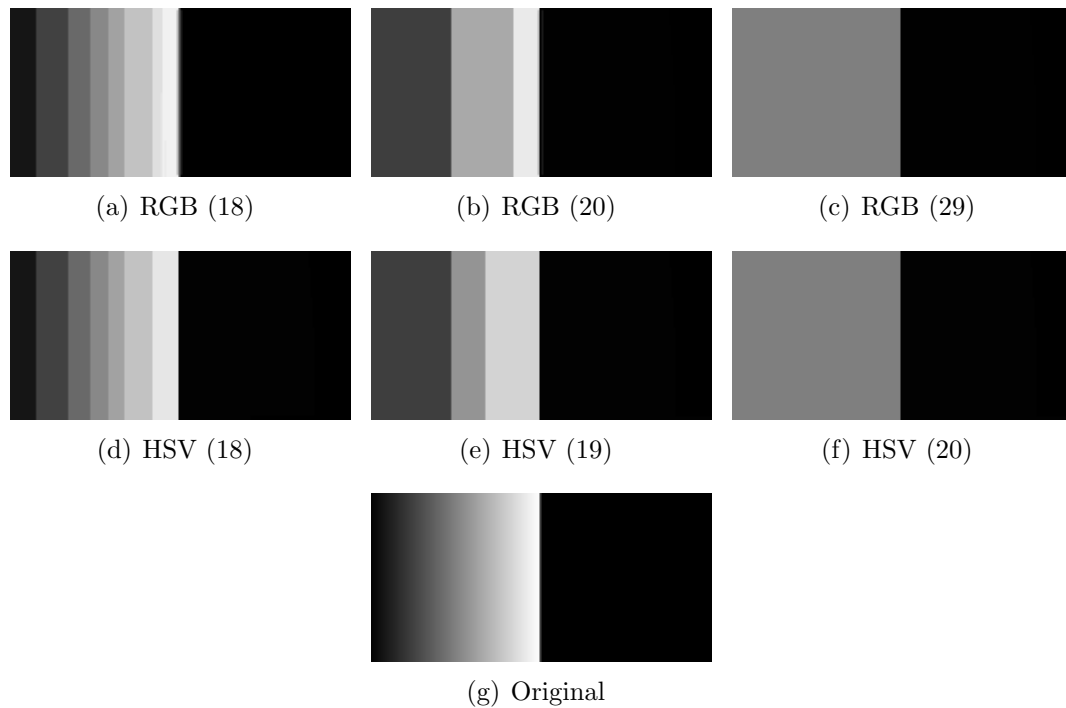


FIGURA 5.3: Segmentación en “Object45”

FIGURA 5.4: Segmentación en “*Rampa*”

Las imágenes mostradas en las figuras 5.5 y 5.6 son de una complejidad mucho mayor por la cantidad de regiones presentes en la imagen.

5.2.3 DGC vs Benchmarks

Aquí se muestran los resultados obtenidos al aplicar la técnica de segmentación elaborada en este proyecto y sus respectivas comparaciones con las segmentaciones alojadas en [5]. Estas corresponden a segmentaciones realizadas por sujetos humanos y por el algoritmo de segmentación mencionado anteriormente que sirve de referencia. Debe recalarse que es el objetivo de la segmentación de propósito general la de alcanzar la segmentación humana.

Por inspección, puede notarse en la figura 5.7 que los resultados obtenidos por la técnica propuesta se asemejan en gran medida a la segmentación humana y distinguen mejor la separación entre el fondo de la imagen y la estructura de las ramas que el algoritmo de referencia. Sin embargo, se producen artefactos en el área correspondiente al cielo e incluso una parte de la rama con este.

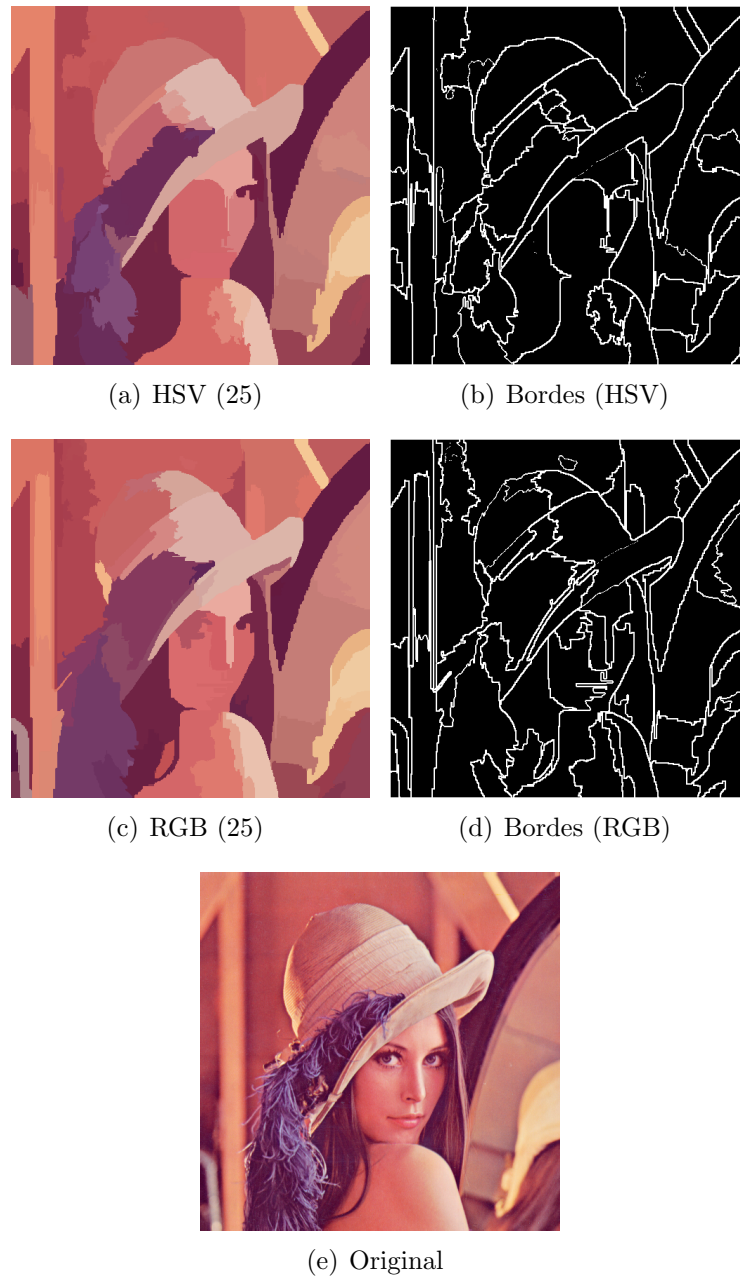
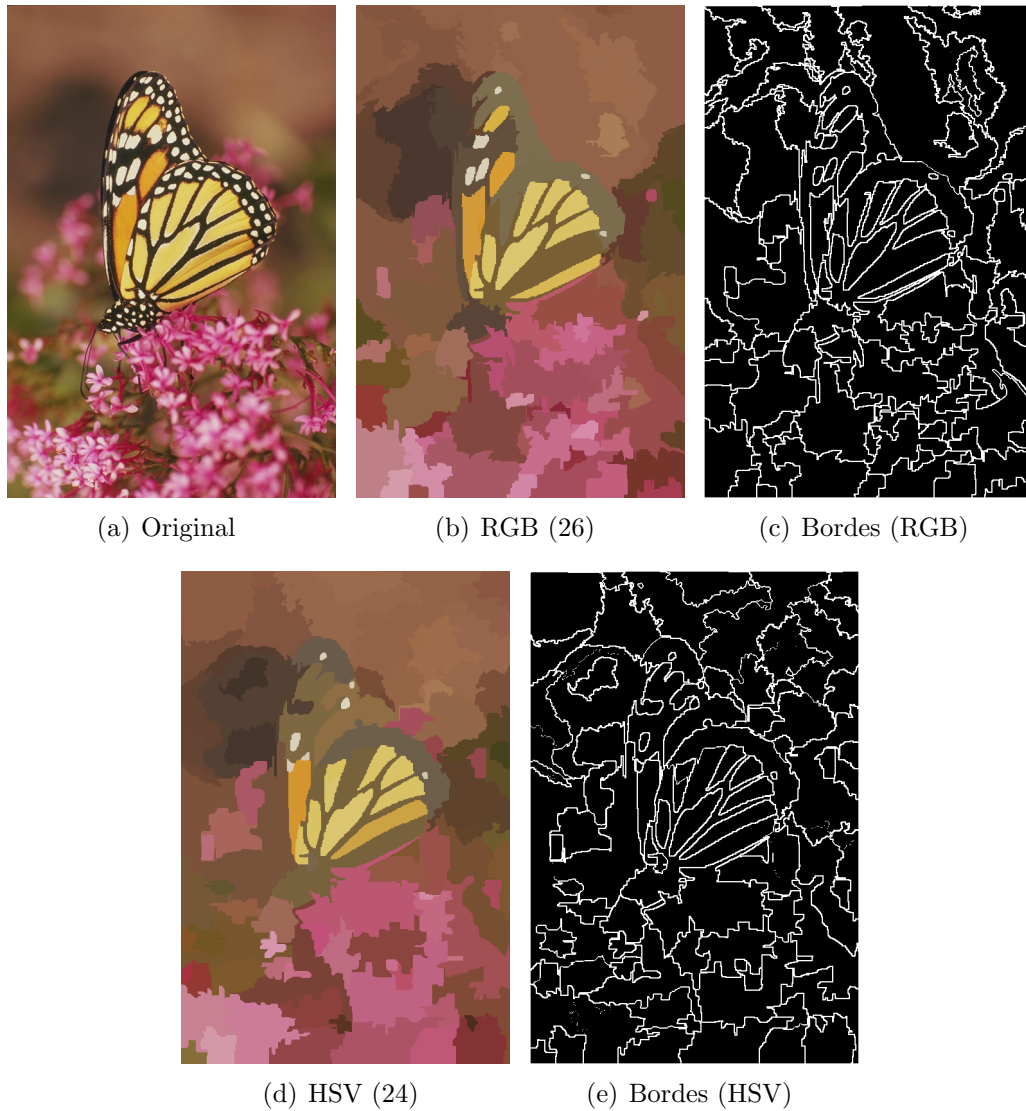


FIGURA 5.5: Segmentación en “Lena3”

FIGURA 5.6: Segmentación en “*Monarch*”

La imagen de la figura 5.8 es notablemente compleja y un problema difícil para técnicas de segmentación de propósito general. Mientras que un humano es capaz de distinguir las partes más importantes de un rostro (ojos, nariz, boca, etc.) por su forma o posición (características también usadas por algoritmos de reconocimiento facial), las técnicas de segmentación de propósito general no cuentan con ese tipo de información tan específica y en los resultados puede verse cómo diversos factores, tales como la iluminación de la escena o la heterogeneidad de las características faciales, pueden causar problemas en la abstracción de los elementos del rostro.

En la imagen de la figura 5.9, se puede observar nuevamente que los resultados obtenidos por la técnica elaborada en este trabajo se acercan más a la segmentación humana que el algoritmo de referencia en [5]. No obstante, se pueden identificar más artefactos en las áreas correspondientes al piso y a la pared del fondo.

5.2.4 Segmentación en fotografía ocular

La fotografía ocular, también llamada retinografía, fotografía de fondo o fotografía retinal permiten capturar imágenes de la estructura del ojo (retina, mácula, fovea y disco óptico) y observar el sistema circulatorio del mismo [23]. La fotografía ocular es útil en el diagnóstico de una gran variedad de condiciones anormales, tales como lesiones físicas (raspaduras y quemaduras), inflamación, glaucoma, leucoma, etc.

Debido a las características de la técnica, dependiendo de la condición, la información de interés podría quedar localizada en alguno de los niveles de la pirámide. Cuando el diagnóstico se centra en el examen del sistema circulatorio (ver figuras 5.10, 5.11, 5.12), es decir, detectar anomalías en alguno de los vasos sanguíneos o encontrar lesiones relativamente pequeñas, la información de interés se conserva en los niveles intermedios de la pirámide y se pierde en los niveles finales debido a que las arterias indican las fronteras entre regiones y aquellas desaparecen en las contracciones. En el caso de lesiones físicas grandes o anomalías que se extienden a lo largo de la estructura ocular (figura 5.13), como el desprendimiento de retina, la información de interés se encuentra en los niveles finales de la segmentación. Debido a una serie de factores, como la estructura irregular de los órganos, la cercanía de los valores numéricos que los instrumentos de captura de imágenes suministran para describir algún fenómeno físico (distancia, densidad, etc.), y la diferencia natural que existe en la formación de los órganos y tejidos de una persona a otra, hacen que la segmentación de imágenes médicas sea un problema sumamente difícil.

Los resultados obtenidos al aplicar la técnica propuesta se muestran en las figuras 5.10, 5.11, 5.12 y 5.13.



(a) Original



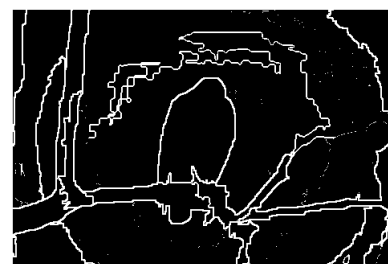
(b) Segmentación humana



(c) Benchmark



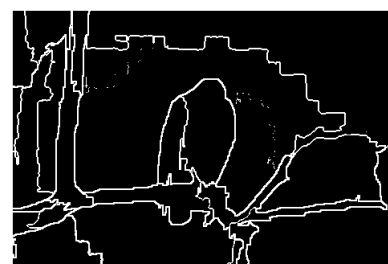
(d) Regiones RGB (25)



(e) Bordes (RGB)



(f) Regiones HSV (25)

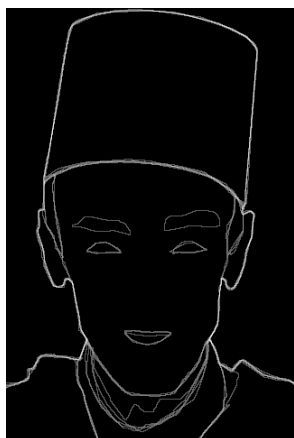


(g) Bordes (HSV)

FIGURA 5.7: Comparación de resultados del objeto 42049 en [5].



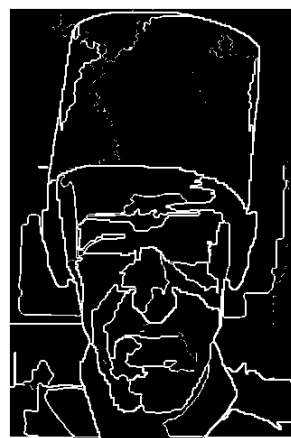
(a) Original



(b) Seg. humana



(c) Regiones RGB (21)



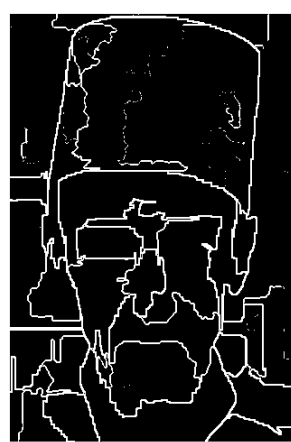
(d) Bordes (RGB)



(e) Benchmark



(f) Regiones HSV (23)

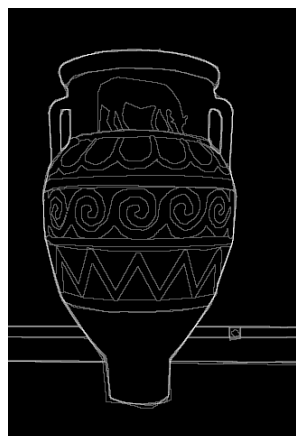


(g) Bordes (HSV)

FIGURA 5.8: Comparación de resultados del objeto 189080 en [5].



(a) Original



(b) Seg. humana



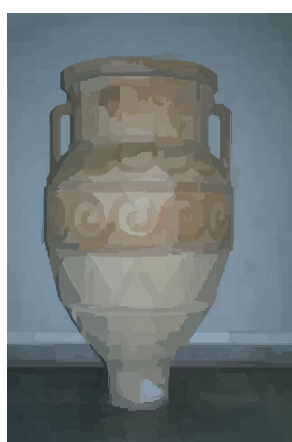
(c) Regiones RGB (12)



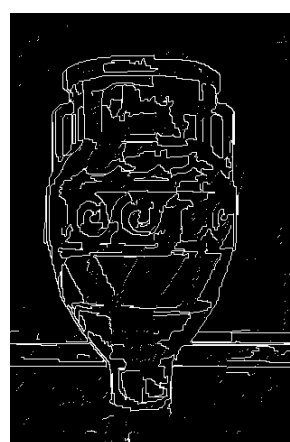
(d) Bordes (RGB)



(e) Benchmark



(f) Regiones HSV (13)



(g) Bordes (HSV)

FIGURA 5.9: Comparación de resultados del objeto 227092 en [5].

Además de obtener las regiones que componen la estructura de la retina, es posible notar diferentes detalles de los vasos sanguíneos como decoloración repentina que podría significar alguna condición anormal. En la figura 5.10(a), se puede apreciar fácilmente una discontinuidad de la arterias localizada en la parte superior derecha de la zona central mientras que, en la imagen original, este mismo detalle no se puede observar. Dicho resultado corresponde a la acumulación de líquido debajo de la retina y su posterior fuga en la capa de los vasos sanguíneos. Cuando existe dicha fuga, en los niveles inferiores/medios de la pirámide ocurre la contracción de los componentes conexos correspondientes a la fuga y a la arteria, causando una decoloración visible en la segmentación. Este fenómeno en la retina se le conoce como coroidopatía serosa central y provoca pérdida parcial de la visión.

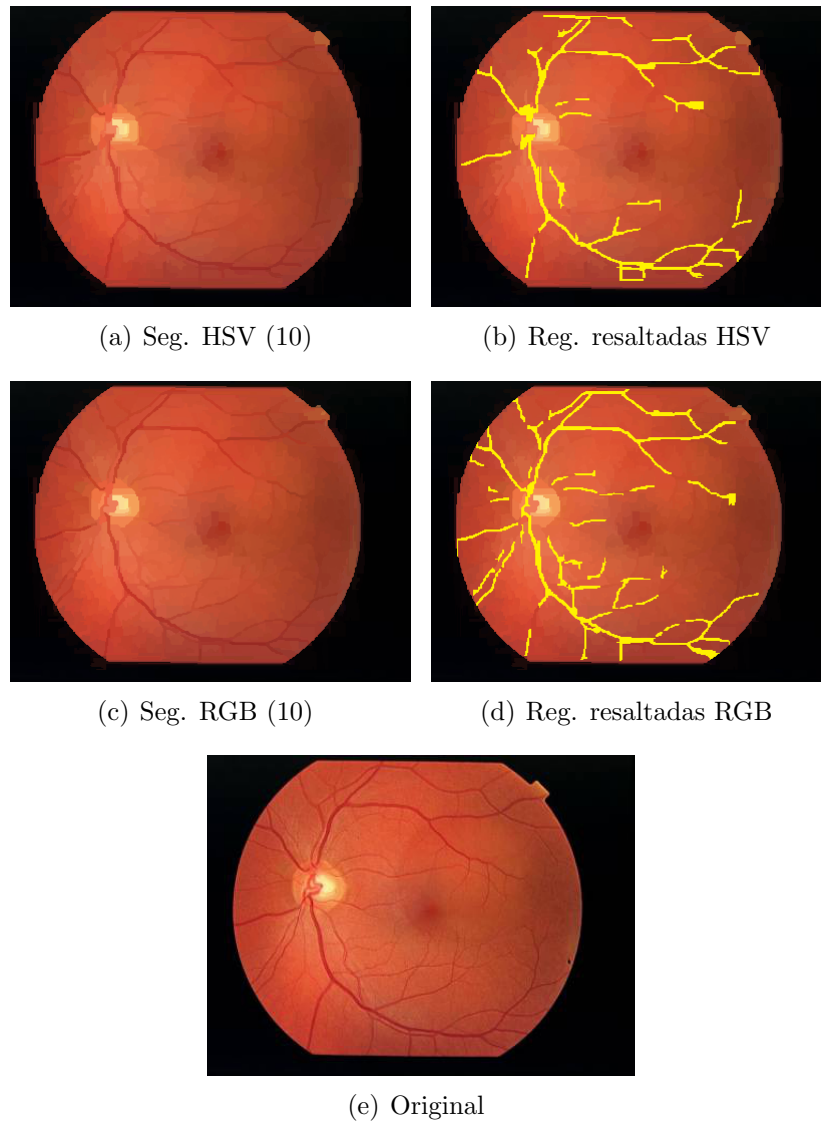


FIGURA 5.10: Segmentación en fotografía ocular 1 (con coroidopatía serosa central)

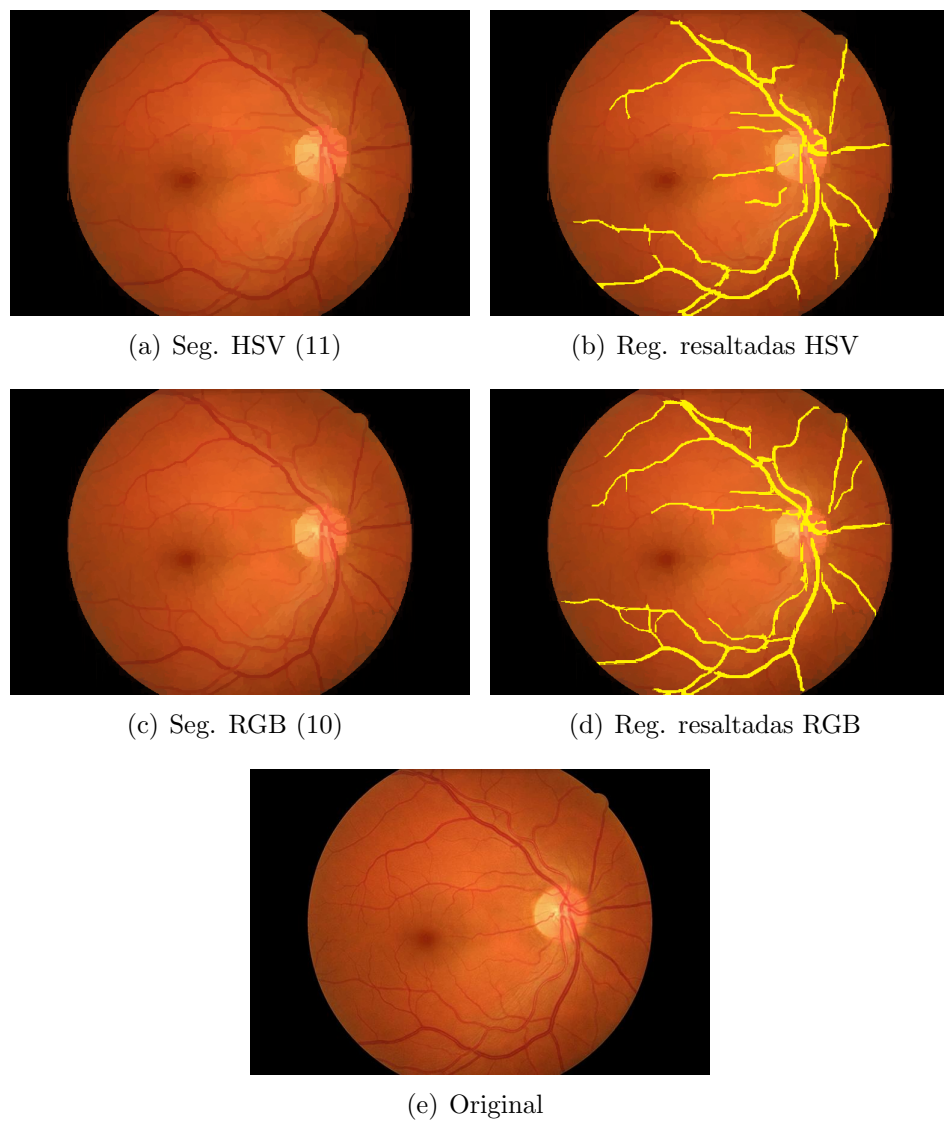


FIGURA 5.11: Segmentación en fotografía ocular 2 (retina normal)

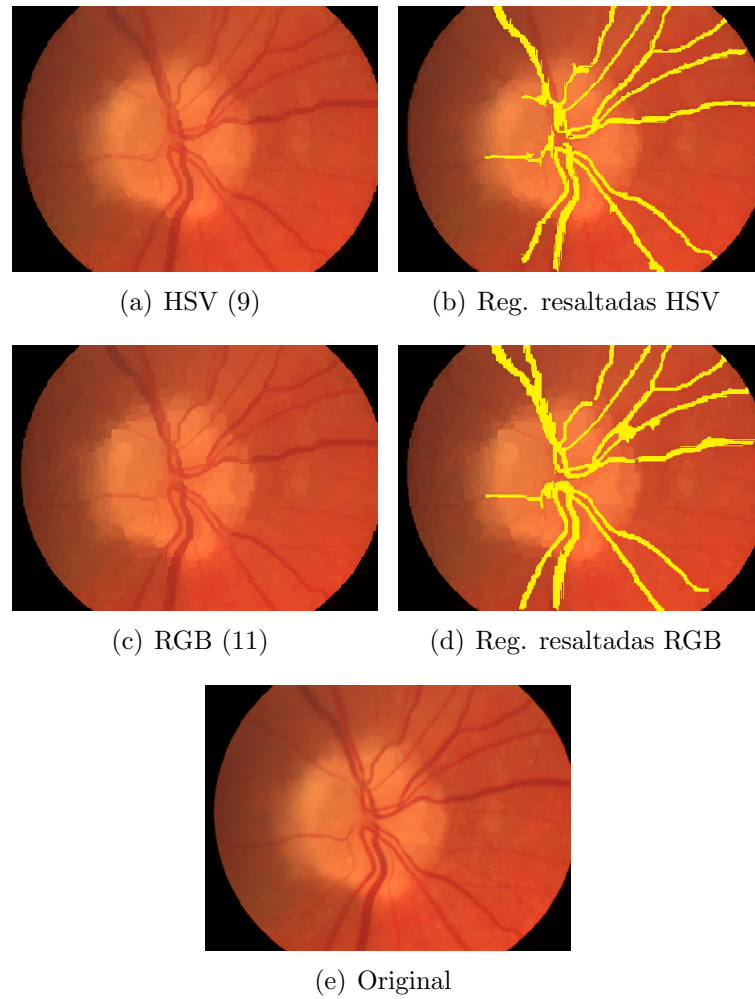


FIGURA 5.12: Segmentación en fotografía ocular 3 (retina normal)

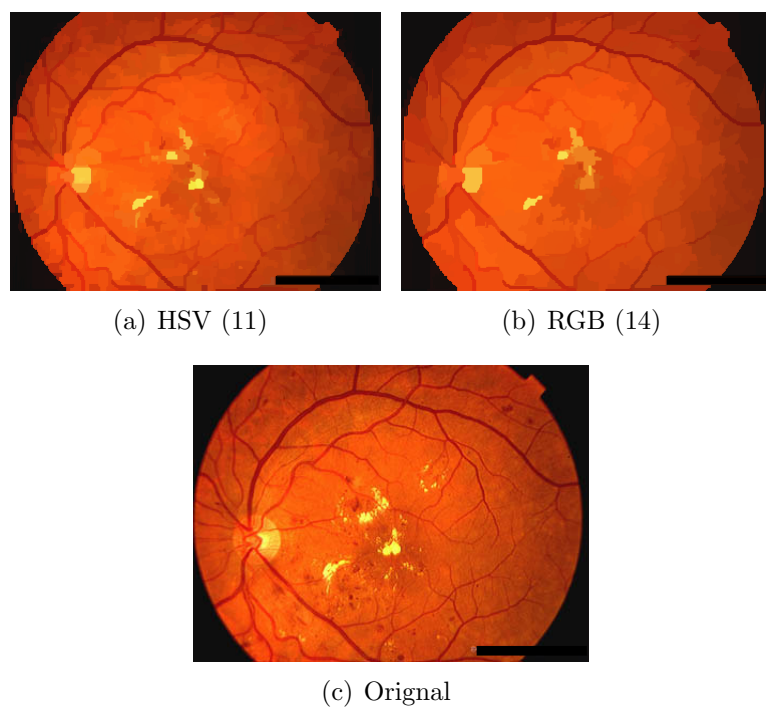


FIGURA 5.13: Segmentación en fotografía ocular 4 (retinopatía diabética edematosa)

Capítulo 6

Conclusiones y trabajo a futuro

6.1 Conclusiones

Los resultados obtenidos en las distintas imágenes, mostrados en el capítulo 5 y comparados con otras técnicas permiten visualizar las ventajas y desventajas del algoritmo de contracción en diferentes situaciones. Sin embargo, cuando se habla de imágenes médicas, no hay ninguna base de datos que permita la comparación y medición objetiva de los resultados, por lo que el análisis final se hace mediante inspección regular. La contracción en imágenes médicas tiene un mejor resultado cuando se utiliza el espacio de colores RGB, mientras que en imágenes generales, donde hay más variedad de colores, el espacio de colores HSV identifica las regiones de una forma más natural. Una cantidad de ruido sustancial (pero imperceptible para el ojo humano) empeora la calidad final de la segmentación, a pesar de la influencia de la función τ y un preprocesamiento de la imagen a tratar resulta ser muy beneficioso. Cuando hay una gran cantidad de regiones de colores parecidos pero visiblemente distinguibles unas de otras, la segmentación en la pirámide se torna en un problema muy difícil y dos fenómenos pueden llegar a presentarse en este caso:

- Construcción de la pirámide de forma parcial, es decir, a pesar de haber una gran cantidad de vértices en el último nivel, ningún par de componentes conexos cumplen la condición de mezcla.
- La norma entre puntos del espacio de colores elegido marca una diferencia notable pero que visualmente es prácticamente imperceptible.

El primer fenómeno puede enfrentarse verificando el número de contracciones realizadas por nivel e ignorar por ese nivel la condición de contraste. De esa manera, forzosamente se realizan contracciones de aristas mínimas, es decir, de colores semejantes. No obstante, mientras más alta sea la pirámide, los vértices sobrevivientes poseen colores cada vez más diferentes y estas contracciones forzadas dan lugar a unión de regiones que perceptualmente no se consideran naturales. Ambos fenómenos dan lugar a artefactos en las segmentaciones.

6.2 Trabajo a futuro

Para la comparación analítica y objetiva de los resultados de la técnica sobre otros algoritmos, es necesario en el futuro incluir los valores numéricos que describen la precisión y semejanza con los resultados humanos, como se menciona en [5]. Es posible profundizar la experimentación de la técnica de segmentación sobre diagnósticos específicos en imágenes médicas, así como la variación de los parámetros α y σ para mejorar los resultados obtenidos. Además de la mayor experimentación en distintos tipos de imágenes, también es posible trabajar en las siguientes mejoras sobre el algoritmo utilizado:

- Funciones τ más flexibles, que sean capaces de describir de una manera más adecuada la influencia de ciertos tipos de componentes conexos sobre otros, ya sea por su tamaño, posición, forma, etc.
- La elección de otros espacios de colores que reflejen mejor las similitudes entre colores, como *CIELAB*, entre otros.

- Agregar técnicas de entrenamiento al algoritmo, tales como suministrar la posición, tamaño o colores dominantes de la clase de objetos de mayor interés en la imagen.

Apéndice A. Código de Programa

Contracción Grafo Dual mediante el algoritmo de Borůvka

```
//Contracción de nivel n a n+1 (Jerarquía de Particiones)
public int contraccionN_Np1(int Sel){
    int maxComponente = 0;
    bool boruvka = Boruvka, contraste = Contraste;
    //Se crea nuevo nivel de la pirámide idéntico al anterior
    //y se realizan las operaciones sobre aquel
    GrafosPrimales.Add(GrafosPrimales[nivelAct].Duplicar());
    GrafosDuales.Add(GrafosDuales[nivelAct].Duplicar());
    nivelAct++; //13.- k <- k+1
    if(nivelAct > 1)
    {
        int numVerticesAnt = GrafosPrimales[nivelAct-2].numeroVertices;
        int numVerticesAct = GrafosPrimales[nivelAct-1].numeroVertices;
        int dif = numVerticesAnt - numVerticesAct;
        if (dif < 6)
            contraste = false;
    }
    //Se realiza la contracción con los núcleos de
    //contracción que se obtienen con el algoritmo de
```

```
// Jerarquía de Particiones (Algoritmo 4)
// 3.- Para todos los vértices en Gk
//Aquí pueden aplicarse métodos estocásticos
//de selección de Vértices
DateTime ini = DateTime.Now;
List<VerticeP> verticesGk;
//Selección de método estocástico de vértices
if (Sel == 0 && !Completa)
{
    verticesGk = GrafosPrimales[nivelAct].MIS();
    mensajes.Items.Add("Selección de Vertices: MIS");
}
else
{
    verticesGk = GrafosPrimales[nivelAct].vertices.Values.ToList();
    mensajes.Items.Add("Selección de Vertices: Todos");
}

HashSet<int> aristasMinimas = new HashSet<int>();
//Paso de Boruvka (búsqueda de aristas mínimas)
if (!boruvka)
{
    foreach (VerticeP v in verticesGk)
    {
        foreach(AristaP e in v.aristasIncidentes())
            if(e.Vertice1 != e.Vertice2)
                if(!aristasMinimas.Contains(e.Identificador))
                    aristasMinimas.Add(e.Identificador);
    }
}
else
```

```

{
    foreach (VerticeP v in verticesGk)
    {
        //Suponiendo que si Emin(u) ya está en Emin, entonces
        //se agrega el siguiente Emin(u) a Emin.
        Emin(v, aristasMinimas);
    } //5.- Fin Para
}

//6.- Para todo arista en Emin cuyo contraste externo
// sea menor a su contraste interno.

//Componentes conexos de nivel K a K+1
Dictionary<int, GrafoP> ComponentesConexosK_Kp1
    = new Dictionary<int, GrafoP>();
GrafoP CC1, CC2, CCk1, CCk2;

foreach(int idAr in aristasMinimas)
{
    AristaP e = GrafosPrimales[nivelAct].arista(idAr);
    int v1 = e.Vertice1;
    int v2 = e.Vertice2;

    //Obtención de los Componentes Conexos C0,k
    //en los que se encuentran v1 y v2
    CC1 = CC(v1);
    CC2 = CC(v2);

    //Obtención de los Componentes Conexos Ck-1,k
    //en los que se encuentran v1 y v2
    CCk1 = CC(v1, ComponentesConexosK_Kp1);
    CCk2 = CC(v2, ComponentesConexosK_Kp1);
    if (!(CCk1.vertices.ContainsKey(v2) ||
        CCk2.vertices.ContainsKey(v1)))

```

```

    {
        float contrasteInterno = pInt(CC1, CC2);
        //Condición de contraste entre CC
        if (!contraste)
        {
            mezclaCC(CC1, CC2, e);
            if (maxComponente < CC1.numeroVertices)
                maxComponente = CC1.numeroVertices;
            mezclaCC(CCk1, CCk2, e, ComponentesConexosK_Kp1);
        }
        else
        {
            float contrasteExterno = Ext(CC1, CC2);
            if (contrasteExterno <= contrasteInterno)
            {
                //7.- Incluir e en el núcleo de contracción  $N_{k,k+1}$ 
                mezclaCC(CC1, CC2, e);
                if (maxComponente < CC1.numeroVertices)
                    maxComponente = CC1.numeroVertices;
                mezclaCC(CCk1, CCk2, e, ComponentesConexosK_Kp1);
            }
        }
    }
}

//8.- Fin Para

int contracciones = 0;
//9.- Contraer Grafo  $G_k$  con núcleo de contracción  $N_{k,k+1}$ 

```

```
//10.- Para toda arista en G_k+1
foreach (GrafoP Ck in ComponentesConexosK_Kp1.Values)
{
    foreach (AristaP eC in Ck.aristas.Values)
    {
        contraerAristaPrimal(eC);
        contracciones++;
    }

    VerticeP vS = null;
    //Vértice Sobreviviente
    foreach (VerticeP v in Ck.vertices.Values)
        if (GrafosPrimales[nivelAct].vertices
            .ContainsKey(v.Identificador))
        {
            vS = GrafosPrimales[nivelAct].vertice(v.Identificador);
            break;
        }

    if (vS != null)
    {
        float r = 0, g = 0, b = 0;
        int numTotal = 0;
        foreach (VerticeP vAnt in Ck.vertices.Values)
        {
            VerticeP vH =
            GrafosPrimales[nivelAct - 1].vertice(vAnt.Identificador);
            //Identifica el vértice representante
            //del Componente Conexo
            vH.Padre = vS.Identificador;
            //Calcula el número de píxeles que representa el CC

```

```
        numTotal += vH.tamanoComponente;
    }
    //Calcula promedio ponderado de los colores de los píxeles
    //del CC
    foreach (VerticeP vAnt in Ck.vertices.Values)
    {
        VerticeP vH =
        GrafosPrimales[nivelAct - 1].vertice(vAnt.Identificador);
        float coef = vH.tamanoComponente / (float)numTotal;
        r += vH.Red * coef;
        g += vH.Green * coef;
        b += vH.Blue * coef;
    }
    //Asigna el promedio al vértice representante
    vS.tamanoComponente = numTotal;
    vS.Red = (byte)Math.Round(r);
    vS.Green = (byte)Math.Round(g);
    vS.Blue = (byte)Math.Round(b);
}
}

HashSet<int> nucleoContraccionD;
do
{
    nucleoContraccionD = new HashSet<int>();
    Dictionary<int, VerticeD> vDuales =
        GrafosDuales[nivelAct].vertices;
    foreach (VerticeD v in vDuales.Values)
    {
        int grad = v.grado();
        if (grad < 3)
```

```

{
    //Contrae la arista que NO corresponde
    //a las de menor valor
    List<AristaD> aristas = v.aristasIncidentes();
    List<AristaP> arParalelas = new List<AristaP>();
    foreach (AristaD e in aristas)
    {
        AristaP ep =
        GrafosPrimales[nivelAct].arista(e.Identificador);
        arParalelas.Add(ep);
    }

    //11.- Heredar a las aristas de  $G_{k+1}$ , el mínimo peso
    //de las aristas que se contraen en  $G_k$ 
    //attr_e( $e_{k+1}$ ) <= min{attr_e( $e_k$ )
    //      |  $e_{k+1} = C[e_k, N_k, k+1]$ }
    while(arParalelas.Count > 1)
    {
        float max = arParalelas[0].Peso;
        int emax = 0;
        for (int i = 1; i < arParalelas.Count; i++)
        {
            if (arParalelas[i].Peso > max)
            {
                max = arParalelas[i].Peso;
                emax = i;
            }
        }
        if (!nucleoContraccionD.Contains
            (arParalelas[emax].Identificador))
        {

```

```

        nucleoContraccionD.Add
            (arParalelas[emax].Identificador);
        break;
    }
    else
    {
        arParalelas.RemoveAt(emax);
    }
}
}
}

foreach (int Arista in nucleoContraccionD)
{
    contraerAristaDual(Arista);
    contracciones++;
}
} while (nucleoContraccionD.Count > 0);

//14.- Hasta que  $G_k = G_{k+1}$ 
//Si no hay contracciones en el grafo primal o dual
//entonces  $G_k = G_{k+1}$ 
if (contracciones == 0)
{
    GrafosPrimales[nivelAct - 1].limpiaPadre();
    GrafosPrimales.RemoveAt(nivelAct);
    GrafosDuales.RemoveAt(nivelAct);
    nivelAct--;

    //Si en la selección de vértices no se aplica MIS
    //y no hubo contracciones, se termina la construcción
    //de la pirámide, pues  $G_k = G_{k-1}$ 

```

```
    if (Completa)
    {
        return -1;
    }
    //Si sí se aplicó MIS y no hubo contracciones, se indica
    //que la próxima contracción se aplique en todos los
    //vértices
    Completa = true;
}
else
{
    Completa = false;
}

return nivelAct;
}
```

Bibliografía

- [1] H. Bunke y M. Last A. Kandel. *Applied Graph Theory in Computer Vision and Pattern Recognition*. Studies in Computational Intelligence. Springer, 2007.
- [2] J. Uriarte Adrian. *La Funcional de Mumford y Shah para Imágenes con Objetos Fractales*. Tesis de Maestría BUAP, 2011.
- [3] Á. Suárez Bravo. Análisis de métodos de procesamiento de imágenes estereoscópicas forestales. Technical report, Universidad Complutense de Madrid, 2009.
- [4] C. Xu y Jerry L. Prince D. L. Pham. A survey of current methods in medical image segmentation. Technical report, Department of Electrical and Computer Engineering, 1998.
- [5] D. Tal y J. Malik D. Martin, C. Fowlkes. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.
- [6] R. Diestel. *Graph Theory*. Springer-Verlag Heidelberg, 2005.
- [7] Chenyang Xu y Jerry L. Prince Dzung L. Pham. Current methods in medical image segmentation. Technical report, Department of Electrical and Computer Engineering, 2000.
- [8] H. Kunze y S.K. Alexander E.R. Vrscay, F. Mendivil. The university of waterloo image repository, 2009. <http://links.uwaterloo.ca/Repository.html>.

-
- [9] R. B. Fisher. Change detection in color images. Technical report, Division of Informatics, Edinburgh University, 1999.
- [10] A. Jeffrey y D. Zwillinger I. S. Gradshteyn, I. M. Ryzhik. *Table of Integrals, Series and Products*. Academic Press, 2000.
- [11] S. K. Feiner y J. F. Hughes J. D. Foley, A. van Dam. *Computer Graphics: Principles and Practice*. Addison-Wesley Professional, 1995.
- [12] W. G. Kropatsch. Building irregular pyramids by dual graph contraction. *IEE-Proc. Vision, Image and Signal Processing*, pages 366–374, 1994.
- [13] W. G. Kropatsch. Segmentation graph hierarchies. Technical report, Institute for Computer Aided Automation, Vienna University of Technology, 2004.
- [14] B. B. Mandelbrot. *The Fractal Geometry of Nature*. W.H. Freedman y Co., 1983.
- [15] S. K. Nayar y H. Murase S. A. Nene. Columbia object image library (coil-100), 1996. <http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php>.
- [16] E. Vishnevsky. Color conversion algorithms, 2009. <http://www.cs.rit.edu/~ncs/color>.
- [17] Y. Haxhimusa y A. Ion W. G. Kropatsch. Multiresolution image segmentations in graph pyramids. *Applied Graph Theory in Computer Vision and Pattern Recognition*, pages 3–41, 2007.
- [18] Z. P. y G. Langs W. G. Kropatsch, Y. Haxhimusa. Vision pyramids that do not grow too high. *Elsevier Science*, page 26, 2004.
- [19] S. Jianbo y J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, aug 2000.
- [20] D. Mumford y J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, pages 577–685, 1989.

-
- [21] G. Aubert y P. Kornprobst. *Mathematical Problems in Image Processing*. Springer, 2006.
- [22] R. C. González y R. E. Woods. *Digital Image Processing*. Prentice Hall, 2002.
- [23] B. Cassin y S. Solomon. *Dictionary of Eye Terminology*. Triad Publishing Company, 1990.
- [24] S. Horowitz y T. Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of the Association for Computer and Machinery*, pages 368–388, 1976.
- [25] Y. Haxhimusa y W. G. Kropatsch. Hierarchical image partitioning with dual graph contraction. Technical report, Institute for Computer Aided Automation, Vienna University of Technology, 2003.